### Scilab Manual for ANALOG AND DIGITAL COMMUNICATION LABORATORY

by Prof Shaik Aqeel
Electronics and Telecommunication
Engineering
Sreyas Institute Of Engineering And
Technology<sup>1</sup>

Solutions provided by
Prof Shaik Aqeel
Electronics and Telecommunication Engineering
Sreyas Institute Of Engineering And Technology

February 24, 2021

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



## Contents

Li	st of Scilab Solutions	4
1	AMPLITUDE MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS	7
2	DOUBLE SIDE BAND SUPPRESSED CARRIER MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS	12
3	SINGLE SIDE BAND MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS	17
4	FREQUENCY MODULATION AND ITS SPECTRUM AND YSIS	L- 23
5	PULSE AMPLITUDE MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS	28
6	TIME DIVISION MULTIPLEXING AND DEMULTIPLEXING	32
7	FREQUENCY DIVISION MULTIPLEXING AND DEMULTIPLEXING	38
8	BINARY AMPLITUDE SHIFT KEYING GENERATION AND DETECTION	43
9	BINARY PHASE SHIFT KEYING GENERATION AND DETECTION	48

10	FREQUENCY SHIFT KEYING GENERATION AND DETECTION	53
11	PULSE CODE MODULATION GENERATION AND DETECTION	58
<b>12</b>	DELTA MODULATION GENERATION	63

## List of Experiments

Solution 1.1	Exp01													7
Solution 2.0	Exp02													12
Solution 3.0	Exp03													17
Solution 4.0	Exp04													23
Solution 5.0	Exp05													28
Solution 6.0	Exp06													32
Solution 7.0	Exp07													38
Solution 8.0	Exp08													43
Solution 9.0	Exp09													48
Solution 10.0	Exp10													53
Solution 11.0	Exp11													58
Solution 12.0	Exp12													63

## List of Figures

1.1 1.2	Exp01 Exp01																10 11	
2.1 2.2	Exp02 Exp02																15 16	
3.1 3.2 3.3	Exp03 Exp03 Exp03																21 22 22	)
4.1 4.2	Exp04 Exp04																26 27	
5.1 5.2	Exp05 Exp05																31 31	
6.1 6.2 6.3	Exp06 Exp06 Exp06																36 36 37	,
7.1 7.2	Exp07 Exp07																41 42	
8.1 8.2	Exp08 Exp08																46 47	
9.1 9.2	Exp09 Exp09																51 52	
	Exp10 Exp10																57 57	

11.1	Exp11																62	
12.1	Exp12																66	

## AMPLITUDE MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS

#### Scilab code Solution 1.1 Exp01

1 //Experiment Number:1

11 // OS : Windows 10.1

12 // Scilab 6.0.2

10

```
13
14 clc;
15 clear;
16 close;
17 fm=3;
            // Message freq
18 fc = 20;
           // Carrier freq
19 \, \text{fs} = 100
20 t=0:1/fs:3;
21 p=length(t);
22 am=input('Enter the message signal amplitude=');
23 ac=input('Enter the carrier signal amplitude (ac>am)
     = ');
24
25
  // Message Signal Generation
26
27 msg=am*cos(2*\%pi*fm*t);
28
29 figure(1);
30 subplot(3,1,1);
31 plot(t,msg);
32 xlabel('TIME');
33 ylabel('AMPLITUDE')
34 title('Message Signal');
35
36 // Carrier Signal generation
37 carrier=ac*cos(2*%pi*fc*t);
38 subplot(3,1,2);
39 plot(t,carrier);
40 \text{ xlabel}('TIME');
41 ylabel ('AMPLITUDE')
42 title('Carrier Signal');
43
44 ka=1/ac; //Amplitude sensitivity of the modulator
45 \text{ u=ka*am};
46 disp(u, 'The Modulation Index is')
47
48 //Amplitude Modulation Generation
49
```

```
50 am_mod=(1+ka.*msg).*carrier;
51 subplot (3,1,3);
52 plot(t,am_mod);
53 xlabel('TIME');
54 ylabel('AMPLITUDE')
55 title ('Amplitude Modulated Signal')
56
57 // Frqeuncy Spectrum
d=(-p/2:1:p/2-1)*1/3; // Indexing
59 figure (2)
60 subplot(3,1,1);
61 plot(d,abs(fftshift(fft(am_mod))));// FOURIER
     TRANSFORM OF MODULATED SIGNAL
62 xlabel('FREQUENCY');
63 ylabel('AMPLITUDE')
64 title ('AM Signal Spectrum')
65
66
67 // Demodulation of AM Signal
68 demod=am_mod.*carrier;
69 k=abs(fft(demod));
70 filt = [ones(1,4*fm), zeros(1,p-4*fm)];
71 out=k.*filt;
72 subplot(3,1,3);
73 plot(t,ifft(out));
74 xlabel('TIME');
75 ylabel('AMPLITUDE');
76 title('Demodulated Message')
77
78 //Sample Inputs for the Program
79
80 //Enter the message signal amplitude=1
81
82 //Enter the carrier signal amplitude (ac>am)=2
83
84
85 // The Modulation Index is
86
```

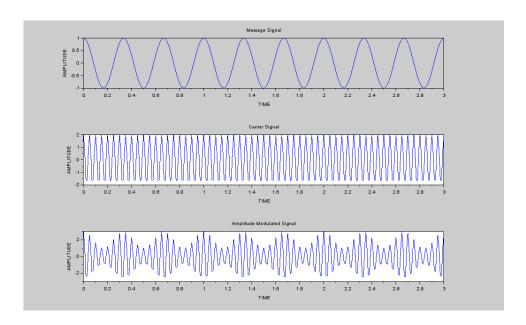


Figure 1.1: Exp01

87 // 0.5

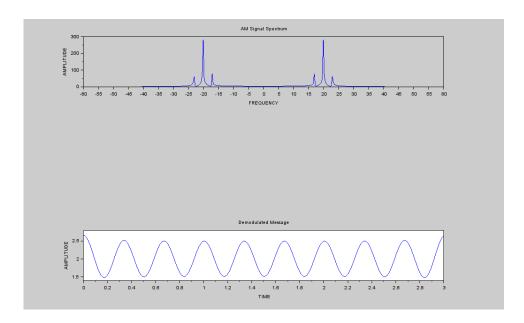


Figure 1.2: Exp01

# DOUBLE SIDE BAND SUPPRESSED CARRIER MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS

#### Scilab code Solution 2.0 Exp02

```
1 // Experiment Number:2
2 // Write a program to perform DSBSC modulation and demodulation and study its spectral characteristics
3 // Analog and Digital Communication Laboratory
4 // B. Tech II Year II Sem
5 // Student Name: Enrolment Number:
6 // Course Instructor: Aqeel Shaik
7 // Sreyas Institute Of Engineering & Technology, Hyderabad.
8 //
```

```
9
10
11 // OS : Windows 10.1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 \, \text{fs} = 100
19 t=0:1/fs:3;
20 p=length(t);
21 fm=input('Enter the message signal frequency =');
22 fc=input('Enter the carrier signal frequency (fc>>>
     fm) = ');
  am=input('Enter the message signal amplitude =');
23
24 ac=input('Enter the carrier signal amplitude =');
25
26 // Message Signal Generation
27
28 msg=am*cos(2*\%pi*fm*t);
29
30 figure (1);
31 subplot(3,1,1);
32 plot(t, msg);
33 xlabel('TIME');
34 ylabel('AMPLITUDE')
35 title('Message Signal');
36
37 // Carrier Signal generation
38 carrier=ac*cos(2*%pi*fc*t);
39 subplot (3,1,2);
40 plot(t,carrier);
41 xlabel('TIME');
42 ylabel('AMPLITUDE')
43 title('Carrier Signal');
44
45 //DSBSC Modulation Generation
```

```
46
47 dsbsc_mod=msg.*carrier;
48 subplot (3,1,3);
49 plot(t,dsbsc_mod);
50 xlabel('TIME');
51 ylabel('AMPLITUDE')
52 title ('Amplitude Modulated Signal')
53
54 // Frquency Spectrum
55 d=(-p/2:1:p/2-1)*1/3;
56 figure (2)
57 subplot (3,1,1);
58 plot(d,abs(fftshift(fft(dsbsc_mod))));// FOURIER
     TRANSFORM OF MODULATED SIGNAL
59 xlabel('FREQUENCY');
60 ylabel('AMPLITUDE')
61 title('DSBSC Signal Spectrum')
62
63
64 //Demodulation of DSBSC Signal
65 demod=dsbsc_mod.*carrier;
66 k=abs(fft(demod));
67 filt = [ones(1,4*fm), zeros(1,p-4*fm)];
68 out=k.*filt;
69 subplot (3,1,3);
70 plot(t,ifft(out));
71 xlabel('TIME');
72 ylabel('AMPLITUDE');
73 title('Demodulated Message')
74
75 // Sample Inputs for the Program
76
77 //Enter the message signal frequency =2
78
79 //Enter the carrier signal frequency (fc>>>fm) =20
80
81 //Enter the message signal amplitude =1
82
```

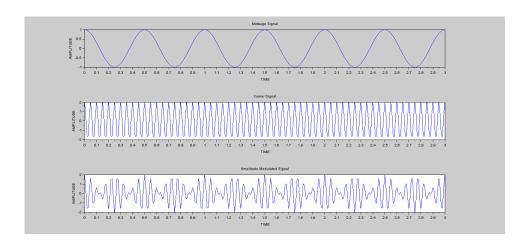


Figure 2.1: Exp02

83 //Enter the carrier signal amplitude =2

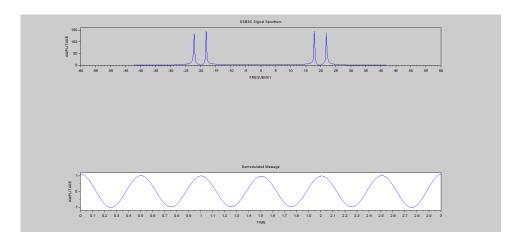


Figure 2.2: Exp02

## SINGLE SIDE BAND MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS

#### Scilab code Solution 3.0 Exp03

```
// Experiment Number:3
// Write a program to perform SSB modulation and demodulation and study its spectral characteristics
// Analog and Digital Communication Laboratory
//B. Tech II Year II Sem
// Student Name: Enrolment Number:
// Course Instructor: Aqeel Shaik
// Sreyas Institute Of Engineering & Technology, Hyderabad.
// 8
//
```

9

10

```
11 // OS : Windows 10.1
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 \text{ fs} = 200
19 t=0:1/fs:2;
20 p=length(t);
21
22 fm=input('Enter the message signal frequency =');
23 fc=input('Enter the carrier signal frequency (fc>>>
      fm) = ');
24 am=input('Enter the message signal amplitude =');
25 ac=input('Enter the carrier signal amplitude =');
26
27 // Message Signal Generation
28
29 msg=am*cos(2*\%pi*fm*t);
30
31 figure(1);
32 subplot (4,1,1);
33 plot(t,msg);
34 xlabel('TIME');
35 ylabel('AMPLITUDE')
36 title('Message Signal');
37
38 // Carrier Signal generation
39
40 carrier=ac*cos(2*%pi*fc*t);
41
42 subplot (4,1,2);
43 plot(t, carrier);
44 xlabel('TIME');
45 ylabel('AMPLITUDE')
46 title('Carrier Signal');
47
```

```
48
49 // Hilbert Transform of Message Signal
50
51 h_msg=imag(hilbert(msg));
52
53 subplot (4,1,3);
54 plot(t,h_msg);
55 xlabel('TIME');
56 ylabel('AMPLITUDE')
57 title('Message Signal');
58
59 // Hilbert Transform of Carrier Signal
60
61 h_carrier=imag(hilbert(carrier));
62
63 subplot (4,1,4);
64 plot(t,h_carrier);
65 xlabel('TIME');
66 ylabel('AMPLITUDE')
67 title('Message Signal');
68
69
70
71 //SINGLE SIDE BAND MODULATION GENERATION
72 ssbmod_lsb=(msg.*carrier)+(h_msg.*h_carrier) //Lower
       Side Band
73
74 figure (2)
75 subplot (4,1,1);
76 plot(t,ssbmod_lsb);
77 xlabel('time');
78 ylabel('amplitude')
79 title('SSB Modulated Signal (LSB)');
80
81 ssbmod_usb=(msg.*carrier)-(h_msg.*h_carrier) //Upper
       Side Band
82 subplot (4,1,2);
83 plot(t,ssbmod_usb);
```

```
84 xlabel('time');
85 ylabel('amplitude')
86 title('SSB Modulated Signal (USB)');
87
88 //Frqeuncy Spectrum of SSB (LSB) Signal
89 d=(-p/2:1:p/2-1)*1/2;
90 subplot (4,1,3);
91 plot(d,abs(fftshift(fft(ssbmod_lsb))));// Normalized
        Frequency spectrum
92 xlabel('frequency');
93 ylabel('amplitude');
94 title('SSB Signal Spectrum (LSB)')
95
96 //Frquency Spectrum of SSB (USB) Signal
97
98 subplot (4,1,4);
99 plot(d,abs(fftshift(fft(ssbmod_usb))));// Normalized
        Frequency spectrum
100 xlabel('frequency');
101 ylabel('amplitude');
102 title ('SSB Signal Spectrum (USB)')
103
104
105 // Demodulation of SSB Signal
106 demod=ssbmod_lsb.*carrier;
107 k=abs(fft(demod));
108 filt = [ones(1,3*fm), zeros(1,p-3*fm)];
109 out=k.*filt;
110 figure (3)
111 subplot (3,1,1);
112 plot(t,ifft(out));
113 xlabel('TIME');
114 ylabel('AMPLITUDE');
115 title ('Demodulated Message')
116
117 // Sample Inputs for the Program
118
\frac{119}{\text{Enter}} the message signal frequency =2
```

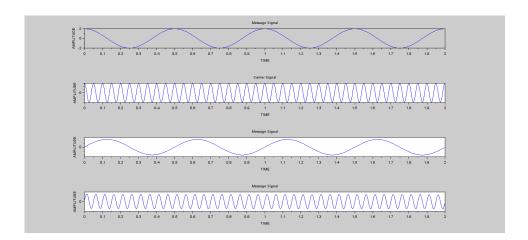


Figure 3.1: Exp03

```
120
121 //Enter the carrier signal frequency (fc>>>fm) =20
122
123 //Enter the message signal amplitude =2
124
125 //Enter the carrier signal amplitude =3
```

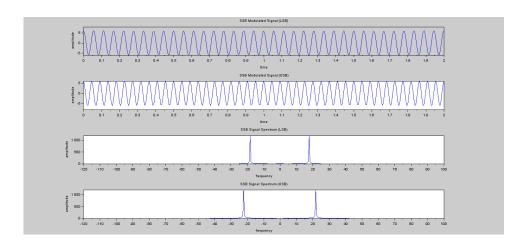


Figure 3.2: Exp03

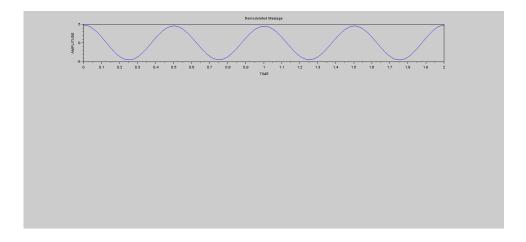


Figure 3.3: Exp03

## FREQUENCY MODULATION AND ITS SPECTRUM ANALYSIS

#### Scilab code Solution 4.0 Exp04

```
14 clc;
15 clear;
16 close;
17
18 \, \text{fs} = 300
19 t=0:1/fs:2;
20 p=length(t);
21
22 fm=input('Enter the message signal frequency =');
23 fc=input('Enter the carrier signal frequency (fc>>>
      fm) = ');
24 am=input('Enter the message signal amplitude =');
25 ac=input('Enter the carrier signal amplitude =');
26
27 // Message Signal Generation
28
29 msg=am*cos(2*\%pi*fm*t);
30
31 figure(1);
32 subplot(3,1,1);
33 plot(t,msg);
34 \text{ xlabel}('TIME');
35 ylabel('AMPLITUDE')
36 title('Message Signal');
37
38 // Carrier Signal generation
39
40 carrier=ac*cos(2*%pi*fc*t);
41
42 subplot (3,1,2);
43 plot(t,carrier);
44 xlabel('TIME');
45 ylabel('AMPLITUDE')
46 title('Carrier Signal');
47
48 // Frequency Modulation Generation
49 \text{ kf} = 4;
50 mod_index=(kf*am)/fm;
```

```
51 disp(mod_index, 'The Modulation Index is');
52
53 \text{ fm_mod=ac*cos}((2*\%pi*fc*t)+(mod_index.*sin(2*\%pi*fm*)
     t)));
54 subplot (3,1,3);
55 plot(t,fm_mod);
56 xlabel('Time');
57 ylabel('Amplitude')
58 title ('Frequency Modulated Signal');
59
60 // Frquency Spectrum
61
62 d=(-p/2:1:p/2-1)*1/3;
63 figure (2)
64 subplot (3,1,1);
65 plot(d,abs(fftshift(fft(fm_mod))));// FOURIER
     TRANSFORM OF MODULATED SIGNAL
66 xlabel('Frequency');
67 ylabel('Amplitude');
68 title ('FM Signal Spectrum')
69
70 //Sample Inputs for Program
71
72 //Enter the message signal frequency =2
73
74 //Enter the carrier signal frequency (fc>>>fm) =23
75
76 //Enter the message signal amplitude =4
77
78 //Enter the carrier signal amplitude =3
79
80
81
    //The Modulation Index is
82
83
    // 8.
```

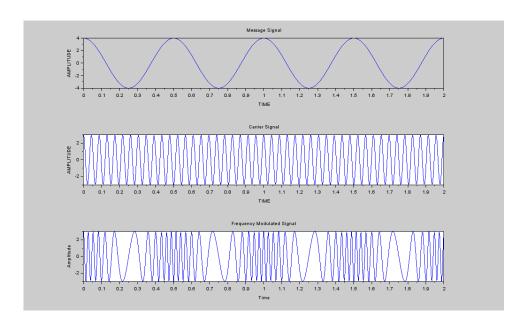


Figure 4.1: Exp04

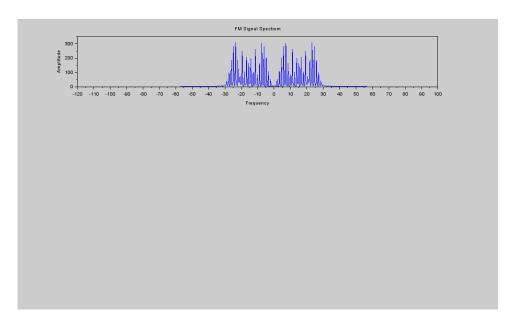


Figure 4.2: Exp04

## PULSE AMPLITUDE MODULATION AND DEMODULATION AND ITS SPECTRUM ANALYSIS

#### Scilab code Solution 5.0 Exp05

9 10 11 // OS : Windows 10.1

```
12 // Scilab 6.0.2
13
14 clc;
15 clear;
16 close;
17
18 \text{ fs} = 300
19 t=0:1/fs:2;
20 p=length(t);
21 fm=input('Enter the message signal frequency =');
22 fc=input('Enter the carrier signal frequency (fc>>>
     fm) = ');
23 am=input('Enter the message signal amplitude =');
24 ac=input('Enter the carrier signal amplitude =');
25
26
27 // Message Signal Generation
28 msg=am+am*sin(2*%pi*fm*t);
29 figure(1);
30 subplot(3,1,1);
31 plot(t,msg);
32 xlabel('time');
33 ylabel('amplitude')
34 title('Message Signal');
35
36 // Carrier Signal generation
37 carrier=ac+ac*squarewave(2*%pi*fc*t);
38
39 subplot(3,1,2);
40 plot(t,carrier);
41 h=gca();
42 h.data_bounds = [0, -1; 2, 3*ac]
43 xlabel('time');
44 ylabel ('amplitude')
45 title('Carrier Signal');
46
47 // Generation of PAM Signal
48 pam_mod=msg.*carrier;
```

```
49 subplot(3,1,3);
50 plot(t,pam_mod);
51 xlabel('time');
52 ylabel('amplitude')
53 title('Pulse Amplitude Modulated Signal');
54
55 //Demodulation of PAM Signal
56 demod=pam_mod.*carrier;
57 k=abs(fft(demod));
58 filt = [ones(1,3*fm), zeros(1,p-3*fm)];
59 out=k.*filt;
60 figure (2)
61 subplot(3,1,1);
62 plot(t,ifft(out));
63 xlabel('TIME');
64 ylabel('AMPLITUDE');
65 title('Demodulated Message')
66
67 //Sample inputs for program
68 //Enter the message signal frequency =3
69
70 //Enter the carrier signal frequency (fc \gg fm) = 25
71
72 //Enter the message signal amplitude =3
73
74 //Enter the carrier signal amplitude =5
```

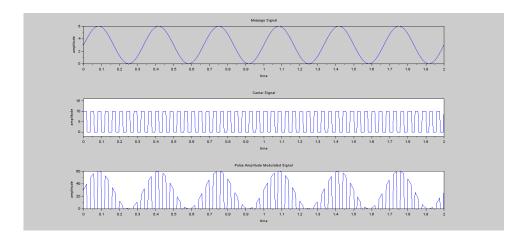


Figure 5.1: Exp05

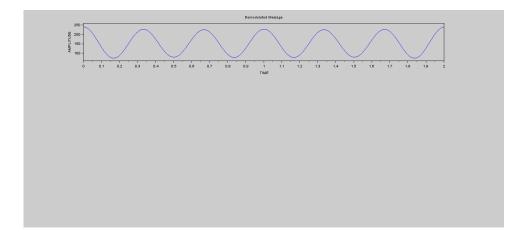


Figure 5.2: Exp05

## TIME DIVISION MULTIPLEXING AND DEMULTIPLEXING

#### Scilab code Solution 6.0 Exp06

```
14 clc;
15 close;
16 clear
17 \text{ fs} = 100
18 t=0:1/fs:1;
19
20 //GENERATION OF 3 MESSAGE SIGNALS FOR MULTIPLEXING
21
22
        //Message Signal 1
23 message_1 = 2*sin(2*\%pi*3*t); //Sine signal of
      frequency 3hz
24 figure(1)
25 subplot (3,1,1)
26 plot2d3(t,message_1)
27 xlabel('TIME');
28 ylabel('AMPLITUDE')
29 title('MESSAGE SIGNAL 1(SINE WAVE)');
30
        //Message Signal
31
32 message_2 = 1*squarewave(2*\%pi*3*t); //Squarewave
      signal of frequency 3hz
33 subplot(3,1,2)
34 plot2d3(t,message_2)
35 xlabel('TIME');
36 ylabel('AMPLITUDE')
37 title('MESSAGE SIGNAL 2(SQUAREWAVE)');
38
39
        //Message Signal 3
40 message_3 = 3*\cos(2*\%pi*3*t) // Cosine signal of
      frequency 3hz
41 subplot (3,1,3)
42 plot2d3(t,message_3)
43 xlabel('TIME');
44 ylabel('AMPLITUDE')
45 title('MESSAGE SIGNAL 3(COSINE WAVE)');
46
47
48 // GENERATIONN OF TIME DIVISION MULTIPLEXED SIGNAL
```

```
49
50 \text{ tdm} = 0;
51 j = 1
52
53 for i=1:3:3*length(t)
54
            tdm(i)=message_1(j);
55
56
            i=i+1;
57
            tdm(i)=message_2(j);
            i=i+1;
58
            tdm(i)=message_3(j);
59
60
            j = j + 1
61
62 end
63
64 figure (2)
65 subplot (2,1,1)
66 plot2d3(tdm)
67 xlabel('TIME');
68 ylabel('AMPLITUDE')
69 title('TIME DIVISION MULTIPLEXED SIGNAL');
70
     // DEMULTIPLEXING OF TDM SIGNAL
71
72
73 n=1
74
      for k=1:1:length(t)
75
            m3(k)=tdm(n)
76
            n=n+1;
77
78
            m4(k) = tdm(n)
79
            n=n+1;
            m5(k)=tdm(n)
80
81
            n=n+1;
82
83 end
84
85
86 figure (3)
```

```
87
88 subplot(3,1,1)
89 plot2d3(m3)
90 xlabel('TIME');
91 ylabel('AMPLITUDE')
92 title('DEMUX MESSAGE SIGNAL 1(SINE WAVE)');
93
94 subplot (3,1,2)
95 plot2d3(m4)
96 xlabel('TIME');
97 ylabel('AMPLITUDE')
98 title('DEMUX MESSAGE SIGNAL 2(SQUAREWAVE)');
99
100
101 subplot (3,1,3)
102 plot2d3(m5)
103 xlabel('TIME');
104 ylabel('AMPLITUDE')
105 title('DEMUX MESSAGE SIGNAL 3(COSINE WAVE)');
```

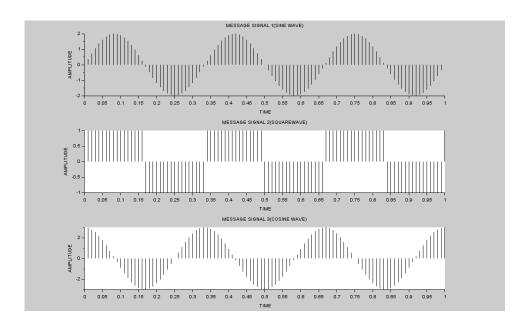


Figure 6.1: Exp06

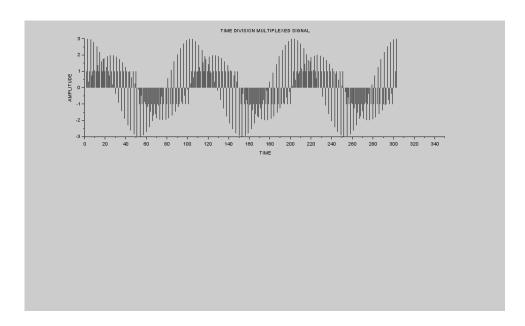


Figure 6.2: Exp06

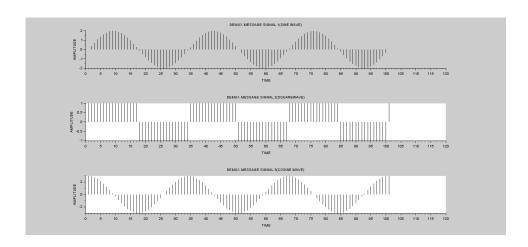


Figure 6.3: Exp06

# FREQUENCY DIVISION MULTIPLEXING AND DEMULTIPLEXING

### Scilab code Solution 7.0 Exp07

```
14 clc;
15 clear;
16 close;
17 \text{ fs} = 100
18 t = 0:1/fs:2;
19
20 //Message signal 1
21
22 msg_1 = 2*cos(2*\%pi*2*t); // Cosine signal of
      frequency 2hz
23 figure(1)
24 subplot (4,1,1);
25 plot(t,msg_1);
26 title("signal 1");
27 xlabel('TIME');
28 ylabel('AMPLITUDE')
29
30 // Message signal 2
31 msg_2 = cos(2*\%pi*9*t); //Cosine signal of frequency
       9 hz
32 subplot (4,1,2);
33 plot(t,msg_2);
34 title("signal 2");
35 xlabel('TIME');
36 ylabel('AMPLITUDE')
37
38 // Frequency Response of Signal -1
39 freqres_msg1 = abs(fft(msg_1));
40 subplot(4,1,3);
41 plot(freqres_msg1);
42 title('Spectrum of signal 1');
43 xlabel('FREQUENCY');
44 ylabel('MAGNITUDE');
45
46 // Frequency Response of Signal -2
47 freqres_msg2 = abs(fft(msg_2));
48 subplot (4,1,4);
49 plot(freqres_msg2);
```

```
50 title("Spectrum of signal 2");
51 xlabel('FREQUENCY');
52 ylabel('MAGNITUDE');
53
54
55 // Frequency Division Multiplexing
56
57 freqres = freqres_msg1+freqres_msg2;
58 figure (2)
59 subplot (3,1,1);
60 plot(fregres);
61 xlabel('FREQUENCY');
62 ylabel('MAGNITUDE');
63 title("FREQUENCY DIVSION MULTIPLEXED SIGNALS");
64
65
66 // Frequency Demultiplexing
67 //Applying filter for signal 1 (Filtering in
      Frequency domian)
68
69 filter_1 = [ones(1,10), zeros(1,180), ones(1,11)];
70 dz1 =freqres.*filter_1;
71 \text{ demod_msg1} = ifft(dz1);
72 subplot (3,1,2)
73 plot(demod_msg1);
74 title("Recovered signal 1");
75 xlabel('TIME');
76 ylabel('AMPLITUDE')
77
78
  // Applying filter for signal 2(Filtering in
      Frequency domian)
80
81 filter_2 = [zeros(1,10),ones(1,180),zeros(1,11)];
82 dz2 =freqres.*filter_2;
83 \text{ demod_msg2} = ifft(dz2);
84 subplot (3,1,3)
85 plot(demod_msg2);
```

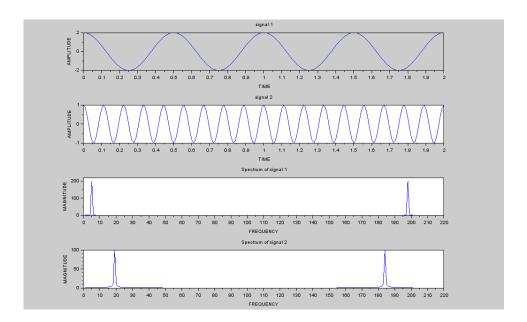


Figure 7.1: Exp07

```
86 title("Recovered signal 2");
87 xlabel('TIME');
88 ylabel('AMPLITUDE')
```

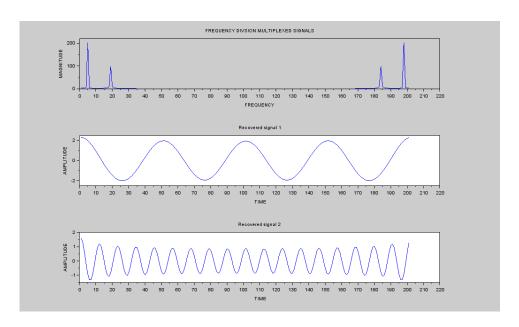


Figure 7.2: Exp07

# BINARY AMPLITUDE SHIFT KEYING GENERATION AND DETECTION

### Scilab code Solution 8.0 Exp08

13

```
14 clc
15 clear
16 close
17
18 n=[0 1 0 1 0 0]; // Random binary Input
19
20 // Binary to polar conversion of Bits
21
22 for m=1:length(n)
         if n(m) == 0
23
24
             nn(m) = -1;
25
         else
26
             nn(m)=1;
27
         end
28 end
29
30
  // Generating NRZ Waveform from bit sequence of bit
       duration 1 sec
32
33 i=1;
34 t=0:0.01:length(n);
35
36 \text{ for } j=1:length(t)
37
        if t(j) \le i
            data(j)=nn(i);
38
39
        else
40
            i=i+1;
41
42
            data(j)=nn(i);
43
44
        end
45
46 \text{ end}
47
48 figure(1)
49 subplot(3,1,1);
50 plot(t,data');
```

```
51 h=gca();
52 \text{ h.data\_bounds} = [0, -1.5; length(n), 1.5]
53 xlabel('TIME');
54 ylabel('AMPLITUDE')
55 title('BINARY INPUT');
56
57 // Carrier Generation
58 \text{ carrier} = \sin(2.*\%\text{pi}*4*\text{t});
59 subplot (3,1,2);
60 plot(t, carrier);
61 xlabel('TIME');
62 ylabel('AMPLITUDE')
63 title('CARRIER SIGNAL');
64
65
     //AMPLITUDE SHIFT KEYING SIGNAL GENERATION
66
67 \text{ for } j=1:length(t)
        if data(j)==1
68
            ask(j)=carrier(j);
69
70
        else
            ask(j)=0;
71
72
        end
73
74 end
75
76
77 subplot(3,1,3);
78 plot(t,ask');
79 xlabel('TIME');
80 ylabel('AMPLITUDE')
81 title('AMPLITUDE SHIFT KEYING SIGNAL');
82
83
84 // Demodualation of ASK Signal
85 for j=1:length(t)
        if ask(j) == carrier(j)
86
            demod(j)=1
87
88
        else
```

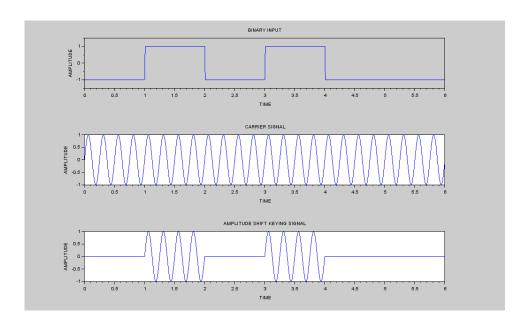


Figure 8.1: Exp08

```
demod(j) = -1
89
90
        end
91
92
   end
93
94 figure(2)
95 subplot(3,1,1)
96 plot(t,demod')
97 xlabel('TIME');
98 ylabel('AMPLITUDE')
99 title('DEMODULATED MESSAGE SIGNAL');
100 h=gca();
101 h.data_bounds=[0,-1.5;length(n),1.5]
```

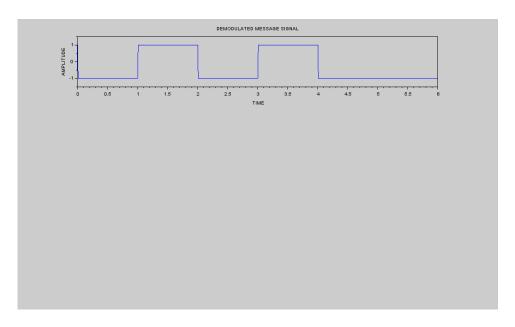


Figure 8.2: Exp08

# BINARY PHASE SHIFT KEYING GENERATION AND DETECTION

### Scilab code Solution 9.0 Exp09

```
// Experiment Number:9
// Write a program to perform Binary Phase Shift
Keying Generation and Detection
// Analog and Digital Communication Laboratory
// B. Tech II Year II Sem
// Student Name: Enrolment Number:
// Course Instructor: Aqeel Shaik
// Sreyas Institute Of Engineering & Technology,
Hyderabad.
//
// OS: Windows 10.1
// OS: Windows 10.1
// Scilab 6.0.2
```

```
14 clear
15 clc
16 close
17
18 n=[1 0 1 0 1 1]; //INPUT RANDOM BINARY SEQUENCE
19
20 // BINARY TO POLAR CONVERSION
21 for m=1:length(n)
22
         if n(m) == 0
23
             nn(m) = -1;
24
         else
25
             nn(m)=1;
26
         end
27 end
28
    // Generating NRZ Waveform from bit sequence of
29
       bit duration 1 sec
30
31 i=1;
32 t=0:0.01:length(n);
33
34 for j=1:length(t)
       if t(j) \le i
35
            data(j)=nn(i);
36
37
       else
38
39
            i=i+1;
            data(j)=nn(i);
40
41
42
       end
43 end
44
45 // Plotting of NRZ Data Waveform
46 figure(1)
47 subplot(3,1,1);
48 plot(t,data');
49 h=gca();
50 \text{ h.data\_bounds} = [0, -1.5; length(n), 1.5]
```

```
51 xlabel('TIME');
52 ylabel('AMPLITUDE')
53 title('BINARY INPUT');
54
55
    // Carrier Generation
56
57 carrier=sin(2.*%pi*2*t);
58 subplot(3,1,2);
59 plot(t,carrier);
60 xlabel('TIME');
61 ylabel('AMPLITUDE')
62 title('CARRIER SIGNAL');
63
64
65 // Generation of BPSK Signal
66 z=carrier';
67 bpsk=data.*z;
68 subplot(3,1,3);
69 plot(t,bpsk');
70 xlabel('TIME');
71 ylabel('AMPLITUDE')
72 title('BINARY PHASE SHIFT KEYING SIGNAL');
73
74
75 // Demodulation of BPSK Signal
76 for j=1:length(t)
77
       if carrier(j) == bpsk(j)
78
           demod(j)=1;
79
       else
           demod(j) = -1;
80
81
       end
82
83 end
84
85 figure (2)
86 subplot(3,1,1);
87 plot(t,demod');
88 xlabel('TIME');
```

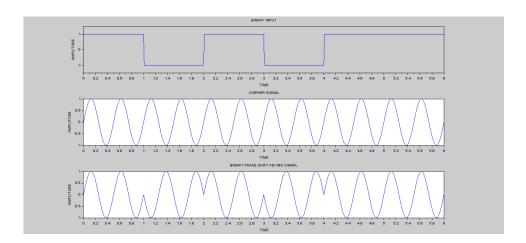


Figure 9.1: Exp09

```
89 ylabel('AMPLITUDE')
90 title('RECOVERED BINARY DATA');
91 h=gca();
92 h.data_bounds=[0,-1.5;6,1.5]
```

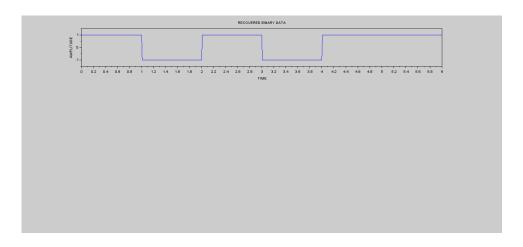


Figure 9.2: Exp09

# FREQUENCY SHIFT KEYING GENERATION AND DETECTION

### Scilab code Solution 10.0 Exp10

13

```
14 clc
15 clear
16 close
17
18 n=[1 0 1 0 0 1]; // Random binary Input
19
20 // Binary to polar conversion of Bits
21
22 for m=1:length(n)
         if n(m) == 0
23
24
             nn(m) = -1;
25
         else
26
             nn(m)=1;
27
         end
28 end
29
30
31 // Generating NRZ Waveform from bit sequence of bit
       duration 1 sec
32
33 i=1;
34 t=0:0.01:length(n);
35
36 \text{ for } j=1:length(t)
        if t(j) \le i
37
            data(j)=nn(i);
38
39
        else
40
            i=i+1;
41
            data(j)=nn(i);
42
43
44
        end
45
46 \, \text{end}
47
    //Plotting of NRZ Data
48
49
50 figure (1)
```

```
51 subplot(3,1,1);
52 plot(t,data');
53 h=gca();
54 \text{ h.data\_bounds} = [0, -1.5; length(n), 1.5]
55 xlabel('TIME');
56 ylabel('AMPLITUDE')
57 title('BINARY INPUT');
58
59 // Carrier Generation
60 carrier_1=sin(2.*%pi*8*t); // Higher Frequency
      Carrier
61 subplot(3,1,2);
62 plot(t,carrier_1);
63 xlabel('TIME');
64 ylabel('AMPLITUDE')
65 title('CARRIER SIGNAL 1');
66
67 carrier_2=sin(2.*%pi*3*t); // Lower Frequency
      Carrier
68 subplot(3,1,3);
69 plot(t,carrier_2);
70 xlabel('TIME');
71 ylabel('AMPLITUDE')
72 title('CARRIER SIGNAL 2');
73
74
     //FSK SIGNAL GENERATION
75 for j=1:length(t)
       if data(j) == 1
76
77
           fsk(j)=carrier_1(j);
78
       else
           fsk(j)=carrier_2(j);
79
80
       end
81
82 end
83
84 figure (2)
85 subplot(3,1,1);
86 plot(t,fsk');
```

```
87 xlabel('TIME');
88 ylabel('AMPLITUDE')
89 title('FREQUENCY SHIFT KEYING SIGNAL');
90
91
92 // Demodualation of FSK Signal
93 for j=1:length(t)
        if fsk(j) == carrier_1(j)
94
            demod(j)=1
95
96
        else
            demod(j) = -1
97
98
        end
99
100 \, \text{end}
101
102 figure(2)
103 subplot(3,1,2)
104 plot(t,demod')
105 xlabel('TIME');
106 ylabel('AMPLITUDE')
107 title('RECOVERED BINARY DATA');
108 h=gca();
109 h.data_bounds=[0,-1.5;length(n),1.5]
```

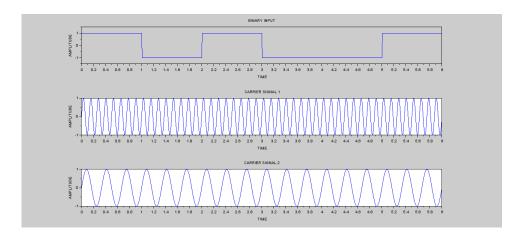


Figure 10.1: Exp10

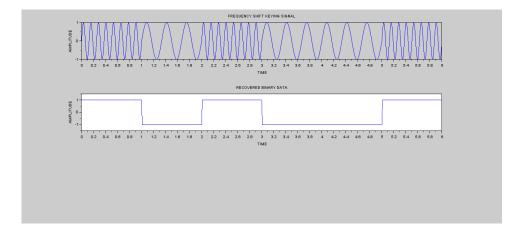


Figure 10.2: Exp10

# PULSE CODE MODULATION GENERATION AND DETECTION

### Scilab code Solution 11.0 Exp11

```
// Experiment Number:11
// Write a program to perform Pulse Code Modulation
Generation and Detection
// Analog and Digital Communication Laboratory
// B. Tech II Year II Sem
// Student Name: Enrolment Number:
// Course Instructor: Aqeel Shaik
// Sreyas Institute Of Engineering & Technology,
Hyderabad.
//
// OS: Windows 10.1
// OS: Windows 10.1
// Scilab 6.0.2
```

```
14 clc;
15 close;
16 clear;
17 f=2;
18 fs=20*f; //Sampling Frequency
19 t=0:1/fs:2;
20 a=2;
21
22 msg=a*sin(2.*%pi*f*t);
23 subplot (3,1,1);
24 plot(t,msg)
25 xlabel('TIME');
26 ylabel('AMPLITUDE')
27 title('Message Signal');
28
29
30 x1=msg+a; // Level Shifting to onesided signal
31 disp(x1, 'Discrete Sampled Values of Message Signal')
     // Displays sampled values
32
33 quant=round(x1);//Quantization
34 disp(quant, 'Quantized Sampled Values'); //Displays
      quantized values
35 enco=dec2bin(quant); //Encoding into binary data
36
37
38
39 deco=bin2dec(enco); // Recovering Analog Message
      signal
40 recover=deco-a;
41 subplot(3,1,2);
42 plot(t,recover)
43 xlabel('TIME');
44 ylabel('AMPLITUDE')
45 title('Recovered Signal');
46 h=gca();
47 h.data_bounds=[0,-3;2,3]
48
```

```
49
50 subplot(3,1,3);
51 plot(t, msg,t, recover, 'r');
52 xlabel('TIME');
53 ylabel('AMPLITUDE')
54 title ('Recovered VS Original Signal');
55 h = gca();
56 \text{ h.data\_bounds} = [0, -3; 2, 3]
57
58
  //Discrete Sampled Values of Message Signal
59
60
61
           // column 1 to 12
62
63
     // 2. 2.618034
                          3.1755705 \qquad 3.618034
64
                                                   3.902113
                3.902113
                             3.618034
                                         3.1755705
           4.
        2.618034 2. 1.381966
65
           // column 13 to 23
66
67
     // 0.8244295
                     0.381966
                                 0.097887 0. 0.097887
68
                                   1.381966
                                                2.
           0.381966 \qquad 0.8244295
        2.618034
                   3.1755705
69
70
             //column 24 to 34
71
                              4. \qquad 3.902113
                    3.902113
72
      //3.618034
                                                  3.618034
           3.1755705 \qquad 2.618034
                                    2. 1.381966
         0.8244295
                    0.381966
73
           // column 35 to 46
74
75
      //0.097887
                         0.097887 \qquad 0.381966 \qquad 0.8244295
76
                    0.
                        2.
                              2.618034
                                          3.1755705
            1.381966
                     3.902113 4.
         3.618034
77
           // column 47 to 57
78
```

```
79
     80
          1.381966 \qquad 0.8244295 \qquad 0.381966 \qquad 0.097887
         0.097887
81
82
           //column 58 to 68
83
     // 0.381966 0.8244295 1.381966 2. 2.618034
84
          3.1755705 \qquad 3.618034 \qquad 3.902113 \qquad 4.
       3.902113
               3.618034
85
         // column 69 to 79
86
87
     //3.1755705 2.618034 2. 1.381966
88
        0.8244295 \qquad 0.381966
                          0.097887
        0.097887 \qquad 0.381966
                          0.8244295
89
          //column 80 to 81
90
91
    // 1.381966 2.
92
93
    //Quantized Sampled Values
94
95
96
          // column 1 to 24
97
98
     //2. 3. 3. 4. 4. 4. 4. 4. 3.
99
           2. 1. 1. 0. 0. 0. 0. 1.
          1. 2. 3. 4.
100
          // column 25 to 48
101
102
     //4. 4. 4. 4. 3. 3. 2. 1. 1. 0.
103
           0. 0. 0. 1. 1.
                                     2. 3.
          4. 4. 4. 4. 4.
104
          //column 49 to 72
105
106
```

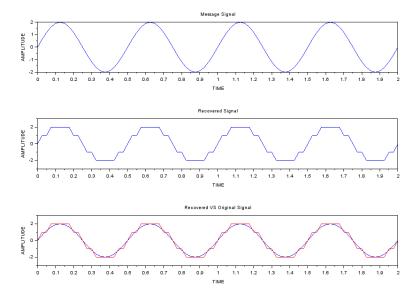


Figure 11.1: Exp11

# DELTA MODULATION GENERATION

### Scilab code Solution 12.0 Exp12

```
1 //Experiment Number:12
2 //Write a program to perform Delta Modulation
      Generation and Demodulation
3 // Analog and Digital Communication Laboratory
4 //B.Tech II Year II Sem
5 //Student Name:
                                  Enrolment Number:
6 // Course Instructor: Ageel Shaik
7 // Sreyas Institute Of Engineering & Technology,
     Hyderabad.
8 //
9
10
11 // OS : Windows 10.1
12 // Scilab 6.0.2
13
14 clc
15 clear
```

```
16 close
17
18 am=input('Enter the message signal amplitude =');
19 fm=input('Enter the message signal frequency =');
20 // Higher Samplig Frequency gives better recovery of
       message Signal
21 fs=input('Enter the sampling frequency(50-300) =');
22 t=0:1/fs:1;
23
24 msg=am*sin(2.*%pi*fm*t);
25 p=length(msg);
26
27 subplot (3,1,1)
28 plot(t, msg);
29 title('Message Signal');
30 xlabel('TIME');
31 ylabel('Amplitude');
32
33 delta=(2.*%pi*am*fm)/fs; //To prevent slope overload
       distortion and Granualar Noise
34 disp(delta, 'The Step Size is')
35
36
37 // Generation of Delta Modulation
38 \text{ delta_mod=0}
39 for i=1:p
40
       if msg(i)>delta_mod(i)
           d(i)=1;
41
           delta_mod(i+1) = delta_mod(i) + delta;
43
       else
           d(i) = 0;
44
45
           delta_mod(i+1) = delta_mod(i) - delta;
46
       end
47 end
48
49
50 subplot (3,1,2)
51 plot2d2(delta_mod)
```

```
52 title('Delta Modulated Output');
53 xlabel('TIME');
54 ylabel('AMPLITUDE');
55
56
57 // Recovery of Message signal (Demodulation)
58 \text{ demod=0}
59 \text{ for } i=1:p
       if d(i) == 1;
60
61
62
            demod(i+1) = delta_mod(i) + delta;
63
       else
64
            demod(i+1) = delta_mod(i) - delta;
65
66
       end
67 end
68
69 subplot (3,1,3)
70 plot(demod);
71 title('RECOVERED MESSAGE SIGNAL');
72 xlabel('TIME');
73 ylabel('AMPLITUDE');
74
75 //Sample Inputs for program
76 //Enter the message signal amplitude =2
77
78 //Enter the message signal frequency =4
79
80 //Enter the sampling frequency (50-300) = 150
81
82
    //The Step Size is
83
84
      //0.3351032
85
```

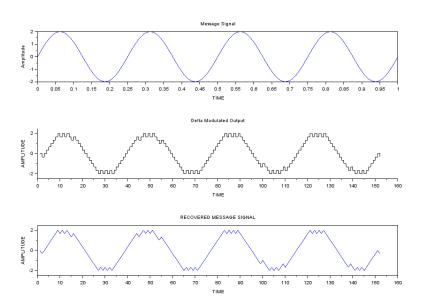


Figure 12.1: Exp12