



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

ANALOG AND DIGITAL COMMUNICATIONS LABORATORY								
IV Semester: ECE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
AECC14	Core	L	T	P	C	CIA	SEE	Total
		-	-	3	1.5	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 45			Total Classes: 45			
Prerequisite: There are no prerequisites to take this course								

I. COURSE OVERVIEW:

Communication engineering is the field of study concerned with the transmission of information either in analog or digital form. The objective of this lab course provides a platform to the students to understand the basics of analog and digital communication systems, modulation techniques, data transmission, multiplexing, etc. There are a wide variety of applications for communications like outdoor broadcasting transmissions and long distance telephone calls

II. COURSE OBJECTIVES:

The students will try to learn

- I. The basic theory of communication system in practice.
- II. The concept of analog to digital conversion for pulse modulation techniques
- III. The analog and digital modulation techniques using ELVIS & MATLAB tool.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

- CO 1: Discriminate the generation and detection of amplitude modulated and frequency modulated signals to calculate the modulation index and frequency deviation.
- CO 2: Analyze the analog pulse modulation and demodulation methods for transmitting the information by pulses.
- CO 3: Apply the concept of pulse code modulation and demodulation for encoded data in analog to digital conversion.
- CO 4: Select the time division or frequency division multiplexing techniques for transmitting multiple signals at a time in the communication system.
- CO 5: Examine the digital modulation techniques for convey more information, high quality and security
- CO 6: Choose appropriate techniques for signal processing and filtering in communication systems.

IV. COURSE CONTENT:

EXERCISES FOR ANALOG AND DIGITAL COMMUNICATIONS LABORATORY

1. Getting Started Exercises

To be proficient in Matlab programming, you need to be able to Communication Systems:

1. Introduction to EMONA Communications board.
2. Generation of various Signals and Sequence

2. Exercises on Analog Modulation Techniques

To be skillful in Matlab programming, you need to be able to realize Modulation Techniques

1. Verify amplitude modulation (AM) system
2. Investigating depth of modulation
3. AM-Double Side Band Suppressed Carrier (DSB-SC) signal using Balanced Modulator
4. Design of Single Side band Suppressed Carrier (SSBSC).

3. Exercises on Angle Modulation Techniques

To be proficient in programming, you need to be able to realize Angle Modulation Techniques:

1. Verifying FM signal using speech signal
2. Determination of Modulation Index in Frequency Modulation (FM).
3. Create a Phase Modulation (PM) waveform using a programming language (like MATLAB) with a carrier frequency of and modulating signal frequency.
4. Phase Deviation Calculation in phase modulation.
5. Generate an FM waveform using MATLAB programming language with the given carrier frequency (e.g., 100 MHz), modulating signal frequency (e.g., 10 kHz), and modulation index (e.g., 2). Plot and visualize the FM waveform.

4. Exercises on Analog Pulse Modulation Techniques

To be proficient in programming, you need implement the following Analog Pulse Modulation Techniques

1. Reconstruction of the original signal from the sampled PAM signal.
2. Calculation of Duty Cycle in Pulse Width Modulation (PWM)
3. Design a reconstruction circuit to convert the PWM signal back to its original analog form.
4. Calculate the pulse width and separation between pulses for a PPM signal.
5. Encode a simple analog signal into a PPM waveform

5. Exercises on Digital Pulse Modulation Techniques

To be proficient in programming, you need implement the following Digital Pulse Modulation Techniques

1. Verify PCM system with a specific number of quantization levels
2. Determine the minimum sampling rate required for PCM encoding of an analog signal

- 3 Verify delta modulation system with a given step size
- 4 Analyze the performance of step size affects the in delta modulation system
- 5 Design an adaptive DPCM system

6. Exercises on Digital Modulation Techniques

To be proficient in programming, you need implement the following Digital Modulation

- 1 Implement: Generating an ASK signal
- 2 Implement: Generating an FSK signal
- 3 Implement: Generating an BPSK signal
- 4 Implement: Generating a QPSK signal

7. Exercises on Signal Processing in Communication Systems

To be proficient in programming, you need implement the signal processing

- 1 Implement: Spectrum of the impulse train
- 2 Implement: Spectrum of the filtered impulse train
- 3 Implement: Spectrum of pseudorandom sequence
- 4 Implement: Analog noise generation (AWGN)

8. Exercises on Signal filtering

To be proficient in programming, you need implement the filtering of signals

- 1 Basics of Filtering
- 2 Implement: Frequency domain analysis in signal filtering
- 3 Design of filter optimization in signal filtering

9. Exercises on Wireless communication systems

To be proficient in programming, you need implement the Wireless Communication Systems

- 1 Implement: Okumura model
- 2 Multipath fading

10. Exercises on Data transmission

To be proficient in programming, you need implement the Data transmission

- 1 Scrambler
- 2 Implement: Spread spectrum modulation

11. Exercises on Multiplexing

To be proficient in programming, you need implement the Multiplexing

- 1 Implement: Time division multiplexing (TDM)
- 2 Implement: Frequency division multiplexing (FDM)

12. Exercises on Adaptive modulation

The LTE is targeting to become the first global mobile phone standard regardless of the different LTE

frequencies and bands use in other countries barrier. Adaptive Modulation and Coding (AMC) is used to increase the network capacity or downlink data rates.

13. Exercises on Generation of Noises

This exercise aims to provide hands-on experience on generation of noises. The term is used, with this or similar meanings, in many scientific and technical disciplines, including physics, acoustic engineering, telecommunications, statistical forecasting, and many more.

1. Generation of White noise
2. Generation of Brown noise
3. Generation of Pink noise

14. Exercises on case study: communication system project

Create a simple voice communication system using microcontrollers and wireless modules.

IV. COURSE CONTENT

EXERCISES FOR ANALOG AND DIGITAL COMMUNICATIONS LABORATORY

Note: Students are encouraged to bring their own laptops for laboratory practice sessions.

1. Getting Started Exercises

To be proficient in programming, you need to be able to:

Overview of communication system

Communication is the process of establishing connection or link between two points for information exchange. The electronics equipments which are used for communication purpose are called communication equipments. Different communication equipments when assembled together form a communication system.

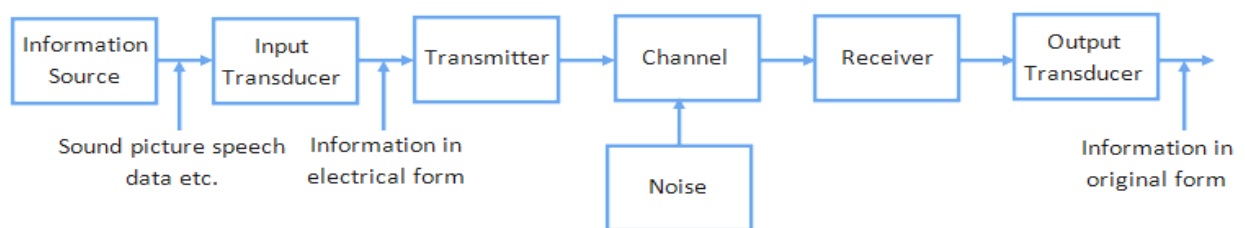


Figure 1.0 shows the block diagram of a general communication system, in which the different functional elements are represented by blocks.

1.1 Introduction to EMONA Communications board

Getting to know the NI ELVIS III platform

Required Tools and Technology

Platform: NI ELVIS III

Hardware: Emona Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Getting to know the EMONA Communications Board

The EMONA Communications Board(ECB) for the NI ELVIS III is used to help learn about communications and telecommunications principles. It lets you bring to life the block diagrams that fill communications textbooks. A "block diagram" is a simplified representation of a more complex circuit. An example is shown in Figure 1.1

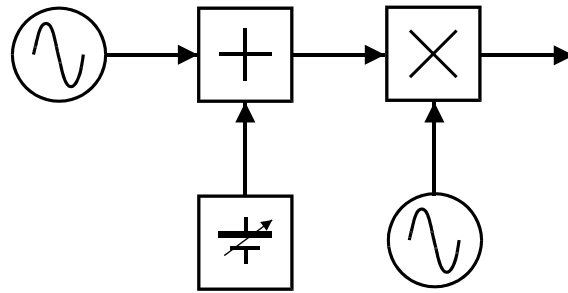
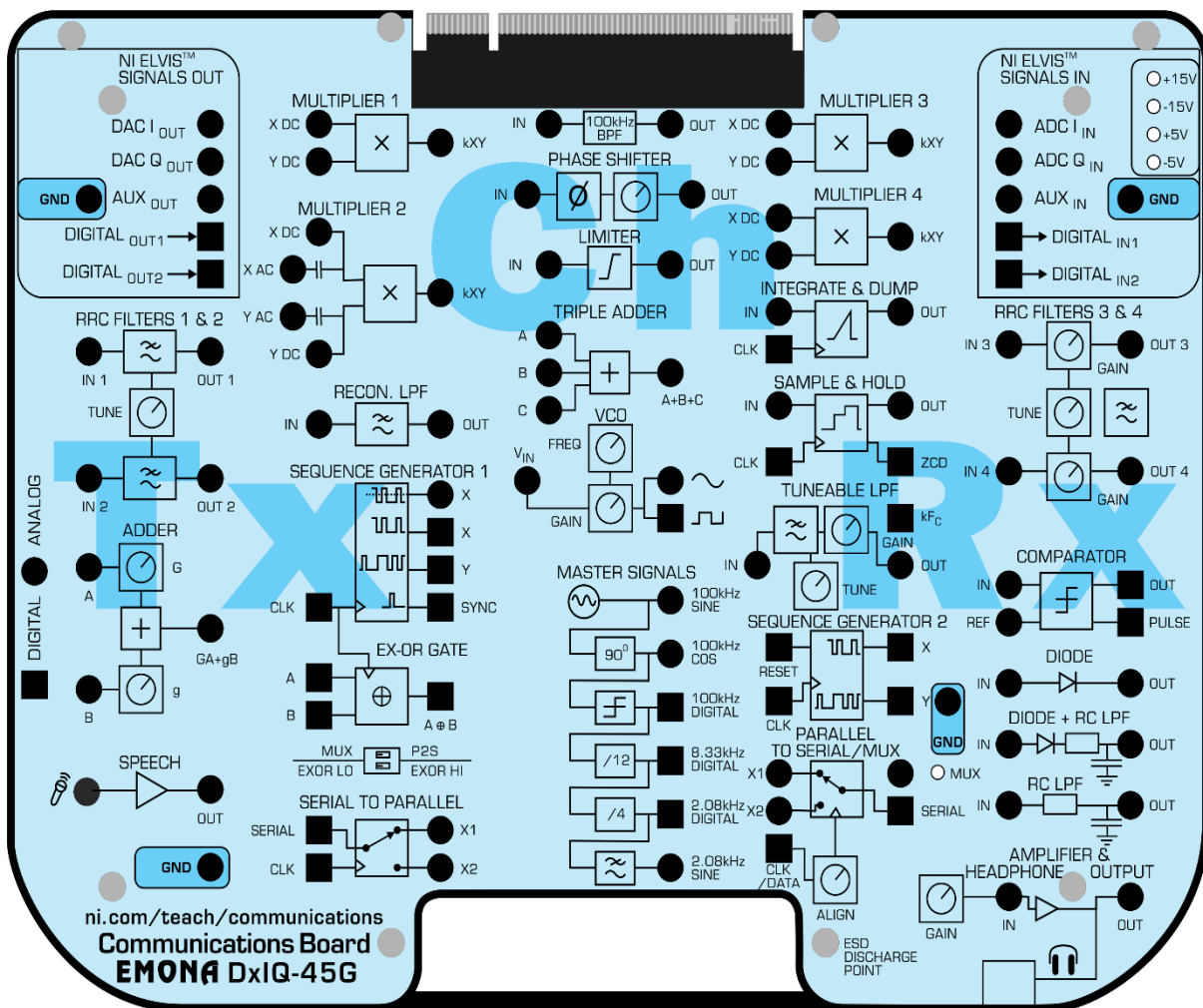


Figure 1.1: A sample communications block diagram



EMONA Communications Board overlay

Introduction to MATLAB

MATLAB is a high-level programming language and environment designed primarily for numerical computing, data analysis, and visualization. It is widely used in academia, research, and industry for tasks ranging from mathematical modeling to algorithm development and simulation.

Basic Syntax:

```
% MATLAB script example
A = [1 2; 3 4];      % Define a matrix
b = [5; 6];          % Define a vector
x = A \ b;            % Solve the system of linear equations
disp(x);              % Display the result
```

1.2 Generation of various Signals and Sequence

Generation of sinusoidal signal

1.2a: Implement: Generating Sinusoidal Signal

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear buds

Powering up the ELVIS III + EMONA Communications Board

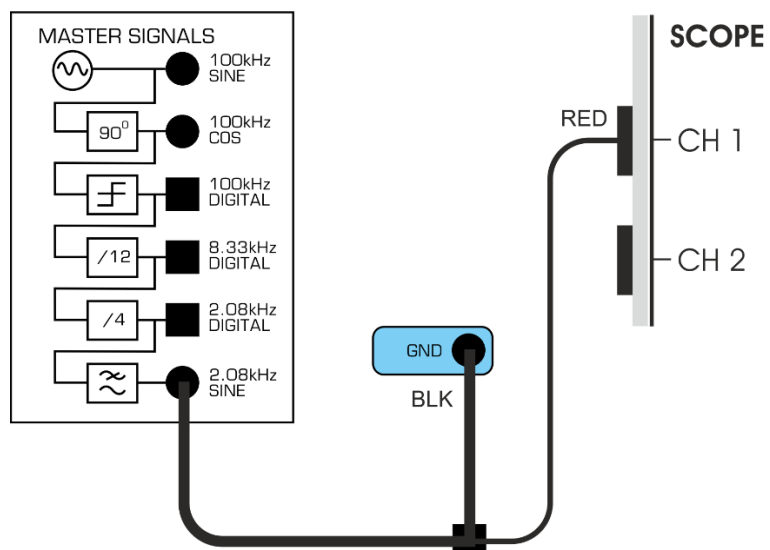


Figure 1.2: Viewing the 2.08 kHz SINE signal
Master Signals



Figure1.2a: 2.08 kHz sine wave block diagram

Generating a sinusoidal signal in MATLAB involves defining the time vector and specifying the amplitude, frequency, and phase of the sine wave.

Hints:

```
% Parameters
clc ;
clear all ;
close all ;

t = 0 : 0.01 : 1 ;
A = 1 ; % Amplitude
f = 2 ; % Frequency (Hz)
phase = 0 ; % Phase ( radians )
t = . . . . . ; % Time vector from 0 to 1 second with a step size of 0. 01
A = . ; % Amplitude
f = . ; % Frequency (Hz)
phase = 0 ; % Phase ( radians )
y = A * . . . . . ; % Sinusoidal signal
plot ( t , y ) ; % Plot the signal
xlabel ( 'Time ( s ) ' ) ;
ylabel ( 'Amplitude ' ) ;
title ( 'Sinusoidal Signal' ) ;
grid on;
```

Try

1. Generate a sinusoidal signal with the parameters amplitude 5 volts and frequency 1KHz . Plot the Sinusoidal signal.
2. Generate a sinusoidal waveform with a frequency of 1 kHz and an amplitude of 1. Plot the waveform for one period.

1.2b -Generating Square wave

Generating a square wave in MATLAB involves defining the time vector and specifying the amplitude, frequency, and phase of the square wave.

Hint

```
% Parameters
t = . . . . . ;
A = 1 ;
% Amplitude ( typically 1 for a square wave )
f = . . ;
% Frequency (Hz ) , e . g . , 2 Hz
phase = 0 ;
% Phase ( radians ) , not as relevant for a square wave
y = . . . . . ;
```



```

t = . . . . .;
% Time vector from 0 to 1 second with a step size of 0.01 seconds
A = . .;
% Amplitude (typically 1 for a square wave)
f = . .;
% Frequency (Hz), e.g., 2 Hz
phase = 0;
% Phase (radians), not as relevant for a square wave
y = . . . . . % Square wave signal
plot ( t , y ) ; % Plot the signal
grid on;

```

Try

1. Use the square function to generate a square waveform with a frequency of 1 kHz and a duty cycle of 50%. Plot the waveform for one period.
2. Generate a square signal with the parameters amplitude 5 volts, frequency 500Hz and duty cycle is 0.5. Plot the Sinusoidal signal.

1.2c -Generating Unit step signal

Generating a square wave in MATLAB involves defining the time vector and specifying the amplitude, frequency, and phase of the square wave.

Hint

```

% Parameters
% Time vector
% Define the time axis
t = . . . . .;
% Generate the unit step signal
u = . . . . . % Initialize the signal with zeros
u(t >= 0) = 1; % Set values to 1 where t is greater than or equal to 0
% Plot the unit step signal
% Set y-axis limits to show the step
grid on;

```

Try

1. Generate and plot a unit step function with a delay of 2 units.
2. Generate a unit step function with the parameter t=-8 to 8 seconds with a step of 0.05 seconds. Plot the unit step function.

1.2d-Generating Exponential signal

Generating a square wave in MATLAB involves defining the time vector and specifying the amplitude, frequency, and phase of the square wave.

Hints:

```
% Parameters
% Generate y values using the exponential function
y = exp(x);
% Plot the exponential function
plot(x, y);
xlabel('x');
ylabel('y');
grid on;
```

Try

1. Generate and plot an exponential growth function $y=2^t$ over the range $t=0$ to $t=5$.
2. Generate an exponential function with the x values range from -3 to +3. Plot the exponential function.

2. Exercises on Analog Modulation Techniques

To be proficient in programming, you need to be able to:

In analog modulation, the modulation is applied continuously in response to the analog information signal. Common analog modulation techniques include: AM, DSB-SC and SSBSC

2.1 Verify amplitude modulation (AM) system

Realize the amplitude modulation (AM) system

Required Tools and Technology

Platform: NI ELVIS III

Hardware: Emona Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Implement: Verifying Amplitude modulation (AM)

Powering up the ELVIS III + EMONA Communications Board

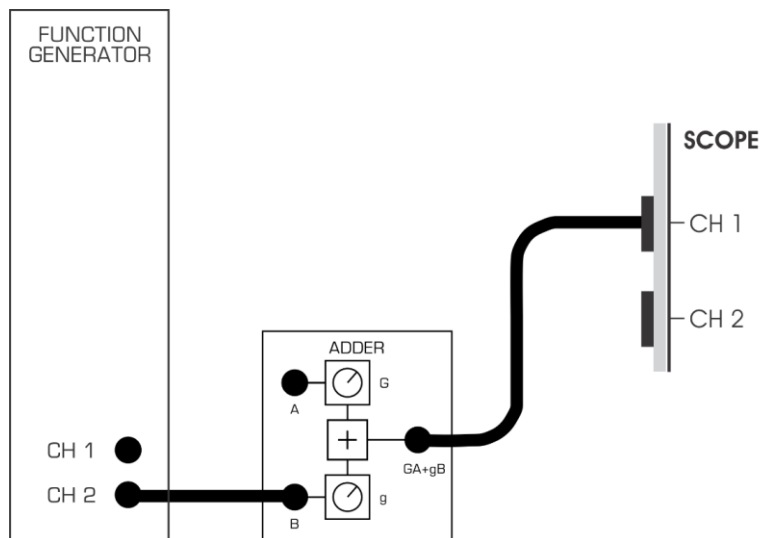


Figure 2.3: Create DC voltage

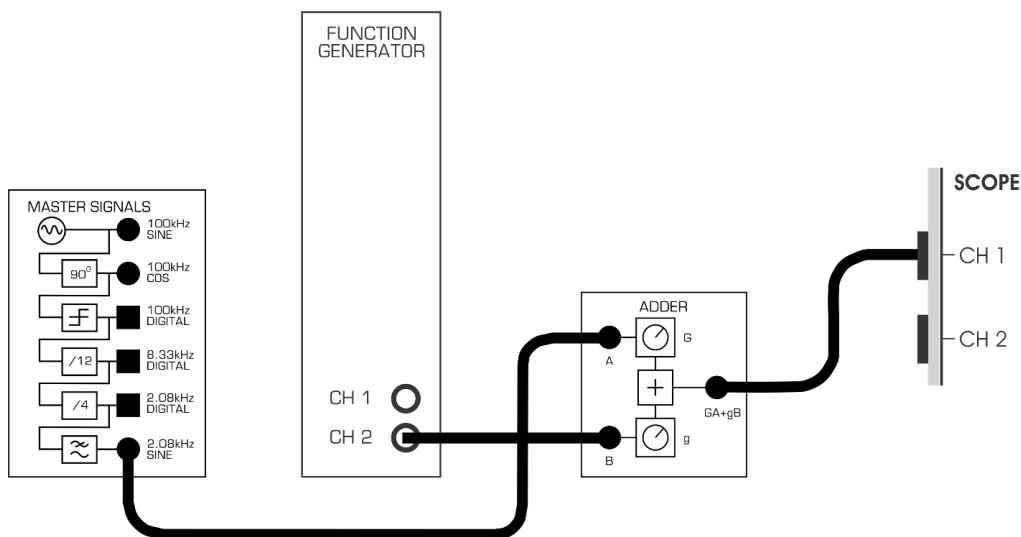


Figure 2.4: Add DC to the message

Modify the set-up as shown in Figure 2.6.

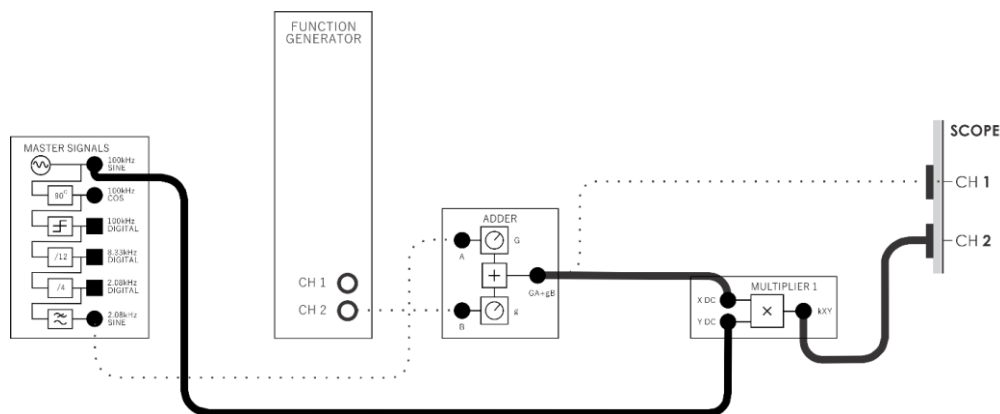


Figure 2.6: Multiply the baseband message by the carrier

$$AM = (DC + \text{message}) \times \text{the carrier.}$$

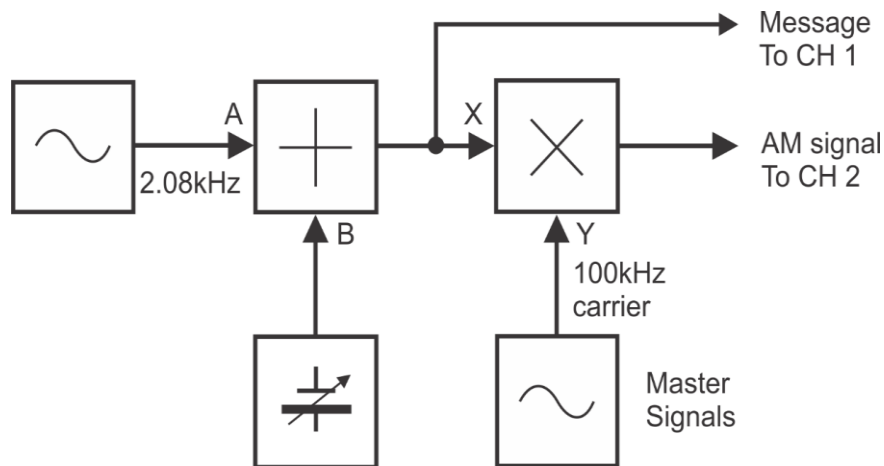


Figure 2.7: Block diagram for AM

With values, the equation on the previous page becomes:

$$AM = (1VDC + 1Vp-p \text{ 2kHz sine}) \times 4Vp-p \text{ 100kHz sine.}$$

Hint

```
% Parameters

t = . . . . .;
f2 =50;
title ( 'under modulation' ) ;
. . . . .
title ( '100% modulation' ) ;
. . . . .
title ( 'over modulation' ) ;
. . . . .
title ( 'demodulation' ) ;
```

Try

1. An AM signal, depth of modulation 100%, from a single tone message, has peak-to-peak amplitude of 4 volts. What would an RMS voltmeter read if connected to this signal?
2. Given an AM signal with carrier amplitude of 10 volts and a modulated signal with a peak amplitude of 14 volts and a trough amplitude of 6 volts: Calculate the modulation depth and determine the percentage modulation.

2.2 Investigating depth of modulation

Hints:

```
% Define parameters
. . . . .
% Time axis
```

```

% Carrier signal
% Modulating signal
% Modulated signal
% Plotting
title('Carrier Signal');
title('Modulating Signal');
. . . . .
title('Modulated Signal');
xlabel('Time (s)');
    % Calculate Peak-to-Peak Amplitudes
peak_to_peak = . . . . .
% Calculate Depth of Modulation (%)
depth_of_modulation = . . . . .
disp(['Depth of modulation: ' num2str(depth_of_modulation) '%']);

```

Try

1. For an AM signal with a carrier signal of 50 volts and a modulating signal with a peak amplitude of 20 volts: Calculate the modulation index and determine if this modulation index indicates over-modulation, under-modulation, or 100% modulation.
2. Given a carrier signal with an RMS voltage of 20 volts and a modulation index of 0.5: Calculate the total power in the modulated signal. Compare it with the total power of the carrier signal alone.

2.3 Verify AM-Double Side Band Suppressed Carrier (DSB-SC) signal using Balanced Modulator

Realize AM-Double Side Band Suppressed Carrier (DSB-SC) signal using Balanced Modulator

Hints:

```

% Define parameters
clc ;
clear all ;
close all ;
. . . . .
title ('message') ;
. . . . .
title ('carrier') ;
s=m.*c;
subplot ( 4,2, [ 5, 6] ) ;
title ('DSB=SC ) ;
. . . . .
title ( 'demodulation' ) ;

```

Try

1. Generate a DSB-SC signal using MATLAB. Assume a message signal of a 3 kHz sine wave and a carrier frequency of 50 kHz. Plot the DSB-SC modulated signal.
2. Design a circuit using a balanced modulator to generate a DSB-SC signal with a message signal of 2 kHz and a carrier frequency of 10 kHz. Assume an ideal balanced modulator and an input message signal amplitude of 1V.

2.4 Design of Single Side band Suppressed Carrier (SSBSC) using Matlab

To design of Single Side band Suppressed Carrier (SSBSC).

Hints:

```
clc;
clear all;
close all;
% Define parameters
. . . . .
% Generate modulating signal
. . . . .
% Generate upper and lower sidebands
. . . . .
% SSBSC signal is the sum of upper and lower sidebands
SSBSC_signal = upper_sideband - lower_sideband;
% Plotting
. . . . .
. . . . .
```

Try:

1. Generate an SSB-SC signal using MATLAB. Assume a message signal of a 4 kHz sine wave and a carrier frequency of 50 kHz. Plot the SSB-SC modulated signal.
2. Design a circuit to generate an SSB-SC signal with a message signal of 4 kHz and a carrier frequency of 20 kHz using the phasing method. Assume an ideal multiplier and phase shifter and an input message signal amplitude of 2V.

3. Exercises on Angle Modulation Techniques

To be proficient in programming, you need implement the following Angle Modulation Techniques: Angle modulation is a class of carrier modulation that is used in telecommunications transmission systems. The class comprises frequency modulation (FM) and phase modulation (PM), and is based on altering the frequency or the phase, respectively, of a carrier signal to encode the message signal.

3.1 Verifying FM signal using speech

Required Tools and Technology

Platform: NI ELVIS III

Instruments used in this lab:

- Oscilloscope-Time
- Oscilloscope-FFT
- Function Generator

Hardware: Emona Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds.

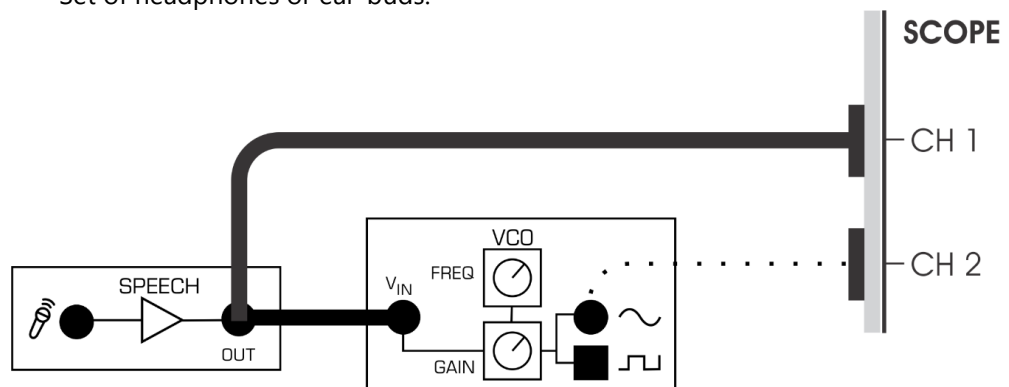


Figure 3.1.1: Modulating the VCO with speech

3.2 Calculation of Frequency Deviation in Frequency Modulation (FM)

In Frequency Modulation (FM), frequency deviation refers to the maximum change in frequency from the carrier frequency due to modulation by the information-bearing signal. This deviation is related to the amplitude of the modulating signal and the modulation index.

Hints:

```
// Calculation of Frequency Deviation in Frequency Modulation (FM)

% Define parameters
. . . . .
% Calculate frequency deviation
. . . . .
% Display the result
. . . . .
```

Try

1. Consider a carrier signal with a frequency of 1 MHz and a modulating signal with a frequency of 100 kHz. The peak amplitude of the modulating signal is 0.5 Volts. Assume the maximum frequency deviation per unit amplitude (k_f) is 10 kHz/V. Calculate the modulation index and the frequency deviation.
2. Design an FM system with a modulation index of 2.5 and a maximum modulating frequency of 10 kHz. Determine the frequency deviation and the bandwidth occupied by the FM signal.

3.3 Determination of Modulation Index in Frequency Modulation (FM)

The modulation index in Frequency Modulation (FM) measures the extent of frequency deviation caused by the modulating signal relative to the carrier frequency. It's a crucial parameter that helps define the bandwidth and characteristics of an FM signal.

Hints

```
//Write the program for Determination of Modulation Index in Frequency Modulation (FM)

% Given parameters
. . . . .
% Calculate modulation index
k = f_dev / f_mod;
% Display result
. . . . .
```

Try

1. Given a carrier signal with a frequency of 100 MHz and a modulating signal with a frequency of 10 kHz, calculate the modulation index for the frequency modulation.
 2. In an FM radio transmission, the maximum frequency deviation is 75 kHz when modulated by an audio signal with a frequency range of 20 Hz to 15 kHz. Determine the modulation index for this FM system.
-

3.4 Create a Phase Modulation (PM) waveform using a programming language (like MATLAB) with a carrier frequency of and modulating signal frequency.

Hints:

```
/**      Implementation of Phase Modulation      **/

% Parameters
fs = 1000; % Sampling frequency
fc = 100; % Carrier frequency
fm = 10; % Modulating frequency
Ac = 1; % Carrier amplitude
Am = 0.5; % Modulating amplitude
deviation = pi/2; % Phase deviation

% Time axis
t = 0:1/fs:1-1/fs;

% Modulating signal
modulating = Am * cos(2*pi*fm*t);

% Phase modulation
. . . . .

% Plotting
. . . . .
```

Try

1. Generate a Phase Modulation (PM) signal let's consider a carrier frequency of 1000 Hz and a modulating signal frequency of 100 Hz.
2. For a phase modulation system, the phase deviation is 45° when modulated by a sinusoidal signal with a frequency of 5 kHz. Calculate the modulation index for this PM system.

3.5 Phase Deviation Calculation in phase modulation

Hints:

```
/**
 * Calculation of Phase Deviation.
 */

% Define carrier and modulating signal parameters
carrier_freq = 1000; % Carrier frequency in Hz
. . . . .

% Calculate the modulation index (assuming peak amplitudes of 1 for simplicity)
. . . . .

% Calculate maximum phase deviation
max_phase_deviation = m * pi; % Maximum phase deviation in radians
```

Try

1. In a phase modulation system, a sinusoidal modulating signal with a frequency of 10 kHz is used to modulate a carrier. If the modulation index is 0.4, calculate the phase deviation produced by this modulation.
2. For a phase modulation system, a modulating signal with a frequency of 2 kHz causes a phase deviation of 30°. Calculate the modulation index for this PM system.

3.6 Generate an FM waveform using MATLAB programming language with the given carrier frequency (e.g., 100 MHz), modulating signal frequency (e.g., 10 kHz), and modulation index (e.g., 2). Plot and visualize the FM waveform.

```
% Define parameters
carrier_freq = . . . . . ; % Carrier frequency in Hz (e.g., 100 MHz)
modulating_freq = . . . . . ; % Modulating signal frequency in Hz (e.g., 10 kHz)
modulation_index = 2; % Modulation index

% Time parameters
. . . . .

% Generate modulating signal
modulating_signal = sin(. . . . .);

% Generate FM waveform
fm_waveform = cos(. . . . .);

% Plot FM waveform
```

Try

- 1 Assume you have a carrier signal with a frequency of 1 kHz and a modulating signal with a frequency of 200 Hz. The peak amplitude of both signals is 1 Volt. Calculate the modulation index and the maximum phase deviation.
- 2 For an FM system, the modulation index is 22 and the modulating signal has a frequency of 5 kHz. Determine the maximum frequency deviation and the bandwidth of the FM signal.

4. Exercises on Analog Pulse Modulation Techniques

To be proficient in programming, you need implement the following Analog Pulse Modulation Techniques.

In Analog pulse modulation also known as continuous wave modulation, the sine wave is used as a carrier wave. In pulse communication system a train of pulse is used as a carrier wave. On superimposing the sinusoidal signal, one of the parameters of the pulse can be varied with respect to signal.

4.1 Implement: Sampling a simple message

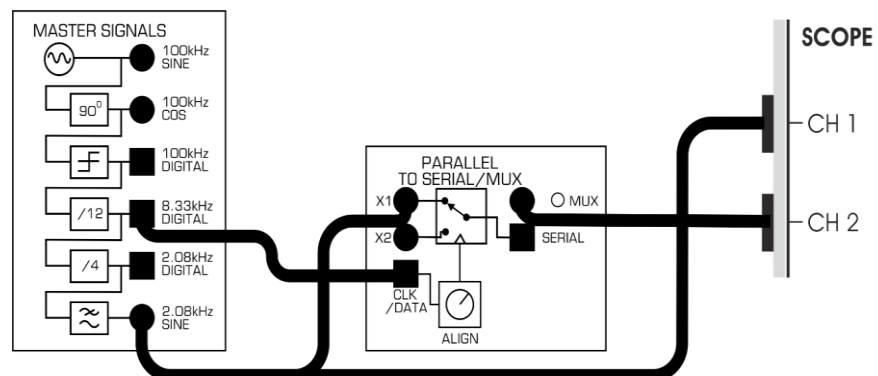


Figure 4.1.2: Patching for sampling with the MUX module

Reconstructing a sampled message

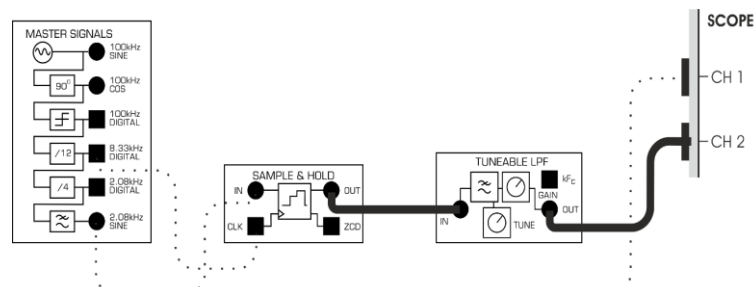


Figure 4.1.3: Patching for sampling reconstruction

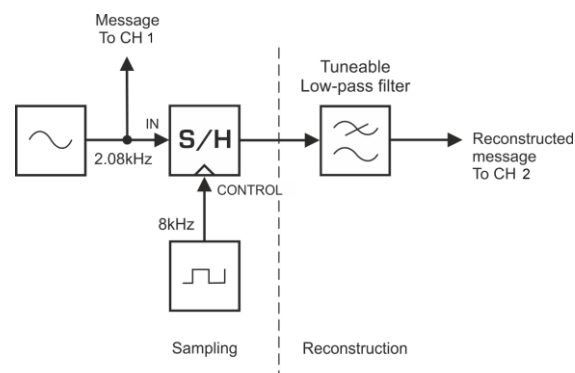


Figure 4.1.4: Block diagram for reconstruction

Hints:

```
/**
 * Implementation of Sampling a simple message.
 */

% Define the parameters for the message signal
message_freq = 1000; % Frequency of the message signal in Hz
amplitude = 1; % Amplitude of the message signal

% Time parameters
sampling_freq = 10 * . . . ; % Sampling frequency (higher than Nyquist)
                        % Time duration for sampling (1 second)
                        % Time vector

% Generate the message signal (sine wave)
message_signal = amplitude * sin(. * . . * . . . * . . .);

% Plot the message signal
figure;
plot(time, message_signal);
xlabel('. . . .');
ylabel('. . . .');
title('Message Signal (Sine Wave)');
```

Try

1. Given the message is a 2.08 kHz sine wave, what's the theoretical minimum frequency for the sampling signal?
2. What happens to the sampled signal's lowest frequency alias when the sampling rate is 4kHz?

4.2 Calculation of Duty cycle in Pulse width modulation (PWM)

. Hints:

```
/** Calculation of Duty cycle */

% High time and total time values (in microseconds)
high_time = 25; % Example high time (in microseconds)
total_time = 100; % Example total time (in microseconds)

% Calculate duty cycle
duty_cycle = (. . . . . / . . . . .) * 100;
fprintf('The duty cycle is: %.2f%%\n', duty_cycle);
```

Try

1. Given a PWM signal with a period of 50 milliseconds and a high time (on time) of 10 milliseconds, calculate the duty cycle.
2. A PWM signal has a period of 20 microseconds and a duty cycle of 40%. Calculate the high time and low time of the signal.

4.3 Design a reconstruction circuit to convert the PWM signal back to its original analog form

Hints:

```
/**
 Realize a reconstruction circuit to convert the PWM signal back to its original
 analog form */

% Define parameters
sampling_freq = 100e3; % Sampling frequency (Hz)
t = 0:1/sampling_freq:1; % Time vector (1 second)

% Generate a PWM signal (example)
pwm_signal = square(. . . . .); % 50 Hz PWM signal with 25% duty cycle

% Reconstruction using an RC low-pass filter
R = 1000; % Resistor value (ohms)
C = 1e-6; % Capacitor value (farads)

% Simulate the low-pass filter response
fc = 1 / (. . . . .); % Cutoff frequency
[num, den] = . . . . .(1, fc / (. . . . . / 2)); % Butterworth filter design
analog_signal = filter(num, den, pwm_signal);

% Plot the signals

subplot(2,1,2);
```

Try

1. Given a PWM signal with a frequency of 1 kHz and a duty cycle of 70%, determine the pulse width and the period of the PWM signal.
2. A PWM signal has a frequency of 1.5 kHz and a duty cycle of 40%. Calculate the pulse width and the period of this PWM signal.

4.4 Calculate the pulse width and separation between pulses for a PPM signal

Pulse Width (PW) = (Modulation Index) * (Time for One Symbol)

Separation Between Pulses (SP) = (Time for One Symbol) - Pulse Width

Hints

```
% Given parameters
modulation_index = 0.4; % Modulation index
time_one_symbol = 10e-3; % Time for one symbol (10 milliseconds)

% Calculate pulse width and separation between pulses
pulse_width = . . . . . ;
separation_between_pulses = . . . . . ;

% Display the results
fprintf(. . . . . );
fprintf(. . . . . );
```

Try

1. Given a PPM signal with a modulation index of 0.6 and a symbol time of 5 milliseconds, calculate the pulse width and separation between pulses.
2. For a PPM signal with a modulation index of 0.4 and a symbol time of 8 microseconds, determine the pulse width and separation between pulses.

4.5 Encode a simple analog signal into a PPM waveform

Hints:

```
// Encode a simple analog signal into a PPM waveform

% Parameters
fs = 1000; % Sampling frequency
f_signal = 10; % Signal frequency
t = 0:1/fs:1-1/fs; % Time axis

% Generate the analog signal (e.g., a sinusoid)
. . . . .
% Sample the analog signal
. . . . .

% Encode the samples into PPM waveform
. . . . .
% Plotting
. . . . .
title('Sampled Signal');
. . . . .
title('PPM Waveform');
xlabel('Time (s)');
ylabel('Amplitude (PPM)');
```

5. Exercises on Digital Pulse Modulation Techniques

To be proficient in programming, you need implement the Digital Pulse Modulation Techniques

Digital pulse modulation techniques are a subset of modulation methods for sending digital information over an analog channel. These techniques do not use digital signals with constant level for modulation. Instead, pulse modulation involves using quantized pulses to modulate a carrier signal or using a pulse train as the carrier signal for digital data.

5.1 Verify PCM system with a specific number of quantization levels

Hints:

```
// Design a PCM system

% Parameters
Fs = 1000; % Sampling frequency (Hz)
t = 0:1/Fs:1-1/Fs; % Time vector for 1 second
f = 5; % Frequency of the sine wave (Hz)
A = 5; % Amplitude of the sine wave
n_levels = 8; % Number of quantization levels
% Create a sine wave signal
x = A * sin(2 * pi * f * t);
% Calculate the quantization step size
. . . . .
% Quantize the signal
```

```
quantized_signal = round(x / delta) * delta;
% Plot the original and quantized signals
. . . . .
```

Try

1. Create a PCM system for an input signal that needs to be quantized with a variable number of quantization levels based on user input.
2. Calculate and display the bit rate required for different numbers of quantization levels.

5.2 Determine the minimum sampling rate required for PCM encoding of an analog signal

Formula for Nyquist Theorem:

Minimum Sampling Rate = $2 \times$ Maximum Frequency in the Signal

Hints

```
//Write Matlab code for minimum sampling rate required for PCM encoding of an analog signal

% Maximum frequency in the analog signal
max_frequency = 4000; % in Hz
% Calculate the minimum sampling rate
min_sampling_rate = 2 * max_frequency;
fprintf('The minimum sampling rate required is %d Hz.\n', min_sampling_rate);
```

Try

1. Determining Minimum Sampling Rate from Maximum Frequency
2. Demonstrates the Nyquist criterion by plotting the spectrum of a sine wave and showcasing the Nyquist frequency.

5.3 Verify delta modulation system with a given step size

Delta modulation is a form of analog-to-digital signal conversion where the difference (delta) between successive samples of the analog signal is encoded and transmitted as a single bit.

Hints:

```
//Write Matlab code for a delta modulation system

% Parameters
fs = 1000; % Sampling frequency
f_signal = 10; % Signal frequency
t = 0:1/fs:1-1/fs; % Time axis
% Generate the analog signal (e.g., a sinusoid)
. . . . .
% Delta modulation parameters
delta = 0.1; % Step size for quantization
initial_state = 0; % Initial state for the quantizer
n_bits = 1; % Number of bits for encoding
```

```
% Delta modulation
. . . . .
% Decode the delta modulated signal
. . . . .
% Plotting
. . . . .
```

Try

1. Implementing a delta modulation system with a specified step size and analyzing its performance.
2. Develop a method to optimize the selection of the step size in delta modulation for a given signal. Consider factors such as signal characteristics, quantization noise, and distortion to determine an optimal step size dynamically.

5.4 Analyze the performance of step size effects in the Delta modulation system

Hints

```
//Write Matlab code for analyze the performance of different step sizes in
a delta modulation system

% Parameters
fs = 1000; % Sampling frequency
f_signal = 10; % Signal frequency
t = 0:1/fs:1-1/fs; % Time axis
% Generate the analog signal (e.g., a sinusoid)
. . . . .
% Quantization step sizes to test
. . . . .
% Initialize arrays to store results
. . . . .
% Perform delta modulation for each step size
. . . . .
% Calculate quantization error
. . . . .
% Plot quantization error and SNR for different step sizes
. . . . .
```

Try

1. Analyze the impact of different step sizes on the performance of a delta modulation system.
2. Encode a sinusoidal signal using delta modulation with different step sizes. Plot the output signal for each step size and observe the impact on signal fidelity, especially in terms of distortion and quantization noise.
3. Calculate the Signal-to-Noise Ratio (SNR) for delta-modulated signals encoded with different step sizes. Analyze how the SNR changes as the step size vary and its effect on the quality of the reconstructed signal.

5.5 Design an adaptive DPCM system

Hints

```
//Write Matlab code for an adaptive DPCM system
% Parameters
fs = 1000; % Sampling frequency
f_signal = 10; % Signal frequency
t = 0:1/fs:1-1/fs; % Time axis
% Generate the analog signal (e.g., a sinusoid)
. . . . .
% Initialize parameters
. . . . .
% Adaptive DPCM encoding
. . . . .
% Decode the quantized signal
. . . . .
% Plotting
. . . . .
```

Try

1. Implement an Adaptive Differential Pulse Code Modulation (ADPCM) system in MATLAB and analyzing its performance.
2. In an ADPCM system, the quantizer step size is initially set to 4. If the received signal deviates from the predicted signal by 6 units and the step size adjustment factor α is 0.5, calculate the updated quantizer step size.

6. Exercises on Digital Modulation Techniques

To be proficient in programming, you need implement the following Digital Modulation Common digital modulation techniques include amplitude-shift keying (ASK), frequency-shift keying (FSK), and phase-shift keying (PSK). ASK is the simplest of these techniques, and is used for low-speed data transmission. FSK is more complex and is used for high-speed data transmission. PSK is the most complex of the three, and is used for very high-speed data transmission.

6.1 Implement: Generating an Amplitude shift keying (ASK) signal

Generating an ASK signal

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Implement: Setting up the data signal

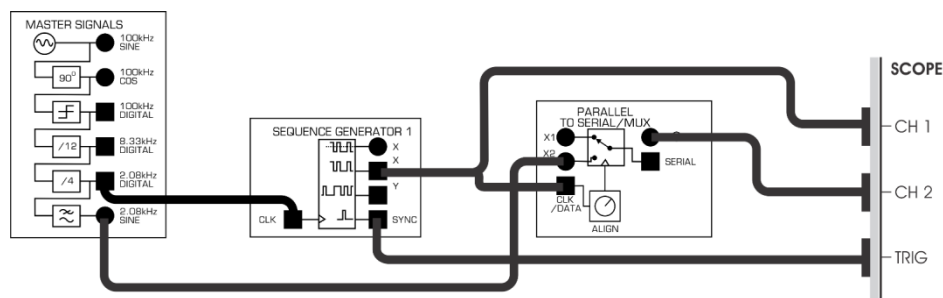


Figure 6.1: Patching for baseband ASK signal generation

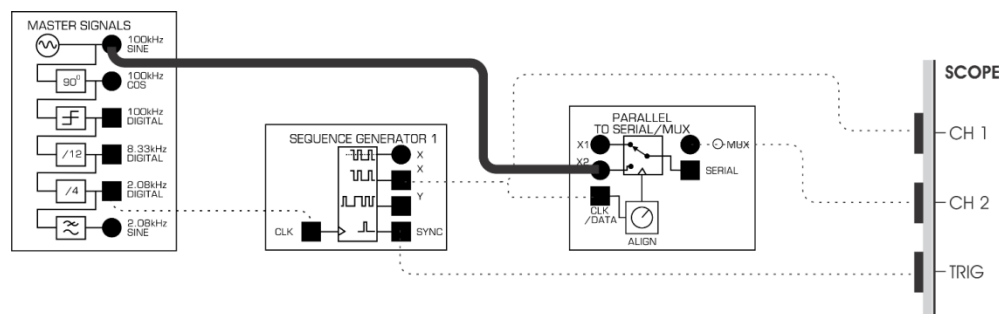


Figure 6.1a: Patching for ASK with 100 kHz carrier

Hints:

```
% Parameters
fs = 10000; % Sampling frequency
fc = 1000; % Carrier frequency
fd = 100; % Data frequency
bit_rate = 1000; % Bit rate
t = 0:1/fs:1-1/fs; % Time axis
% Generate the digital signal (e.g., random binary data)
data = randi([0, 1], 1, length(t) * bit_rate / fs);
% Modulation
. . . . .
% Demodulation
. . . . .
. . . . .
% Plotting
. . . . .
```

Try

1. Generate ASK signal with the sample signal frequency is 1200 and carrier frequency is with 100. Plot the ASK Signal.
2. Generate ASK signal with the sample signal frequency is 1500 and carrier frequency is with 125. Plot the ASK Signal.

6.2 Implement: Generating a Frequency shift keying (FSK) signal

Generating an FSK signal

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Implement: Setting up the data signal

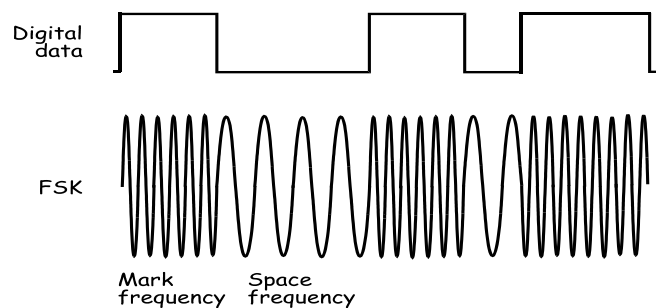


Figure 6.2: Typical FSK signal

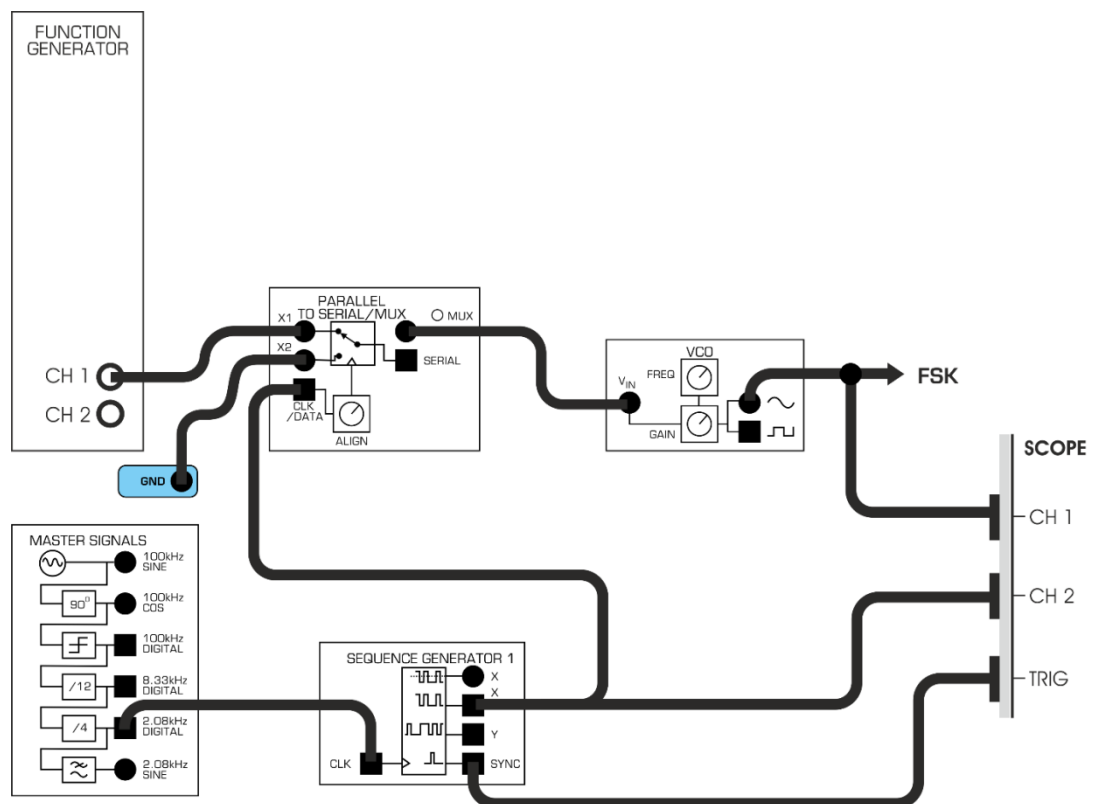


Figure 6.2a: Patching diagram for dual-tone FSK generation

Hints:

```
% Frequency Shift Keying (FSK) modulation and demodulation
% Parameters
fs = 10000; % Sampling frequency
fc1 = 1000; % Carrier frequency for '0'
fc2 = 2000; % Carrier frequency for '1'
fd = 100; % Data frequency
bit_rate = 1000; % Bit rate
t = 0:1/fs:1-1/fs; % Time axis
% Generate the digital signal (e.g., random binary data)
. . . . .
% Modulation
. . . . .
% Demodulation
. . . . .
% Plotting
. . . . .
```

Try

1. Generate FSK signal with the sample signal frequency is 1200 and carrier frequency is with 100. Plot the FSK Signal.
2. Generate FSK signal with the sample signal frequency is 1500 and carrier frequency is with 125. Plot the FSK Signal.

6.3 Implement: Generating an BPSK signal

Generating an BPSK signal

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Implement: Setting up the data signal

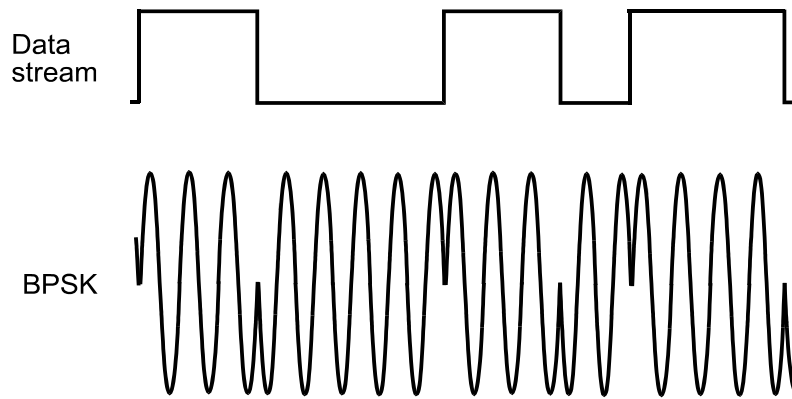


Figure 6.3: BPSK waveform

Figure 6.3a: Patching for BPSK generation

Hints

```
% Binary Phase-Shift Keying (BPSK) modulation and demodulation
% Parameters
fs = 10000; % Sampling frequency
fc = 1000; % Carrier frequency
fd = 100; % Data frequency
bit_rate = 1000; % Bit rate
t = 0:1/fs:1-1/fs; % Time axis
% Generate the digital signal (e.g., random binary data)
. . . . .
% Modulation
. . . . .
% Demodulation
. . . . .
% Plotting
. . . . .
```

Try

1. For a BPSK modulation scheme, if the bit rate is 10 kbps and the carrier frequency is 100 MHz, calculate the duration of each bit (bit period) and the phase shift used in this BPSK modulation.
2. In a BPSK system, if the carrier frequency is 1 MHz and the bit rate is 100 kbps, determine the phase shift used for modulation.

6.4 Implement: Generating an QPSK signal

Generating an QPSK signal

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads

- Assorted 2mm banana-plug patch leads
- Set of headphones or earbuds

Implement: Setting up the data signal

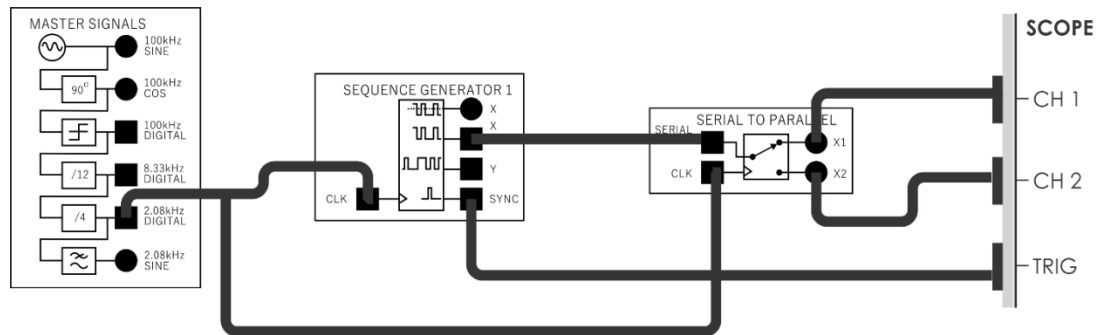


Figure 6.4: Patching for data bit splitting

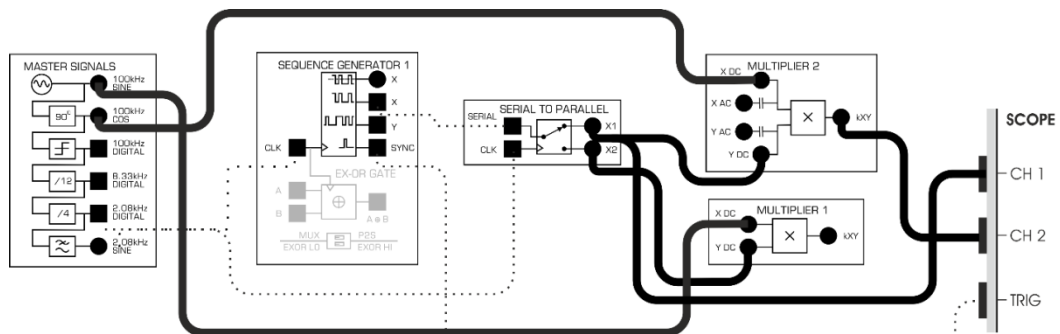


Figure 6.4a: Patching for dual BPSK generation while examining the PSK₁ signal

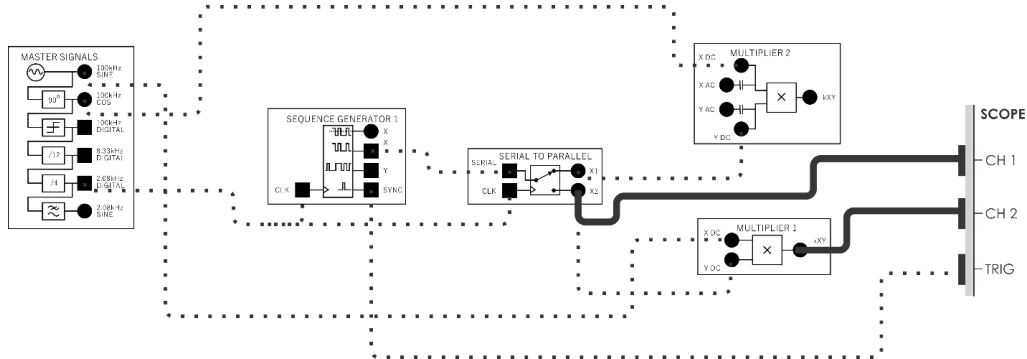


Figure 6.4b: Patching for dual BPSK generation while examining the PSK_Q signal

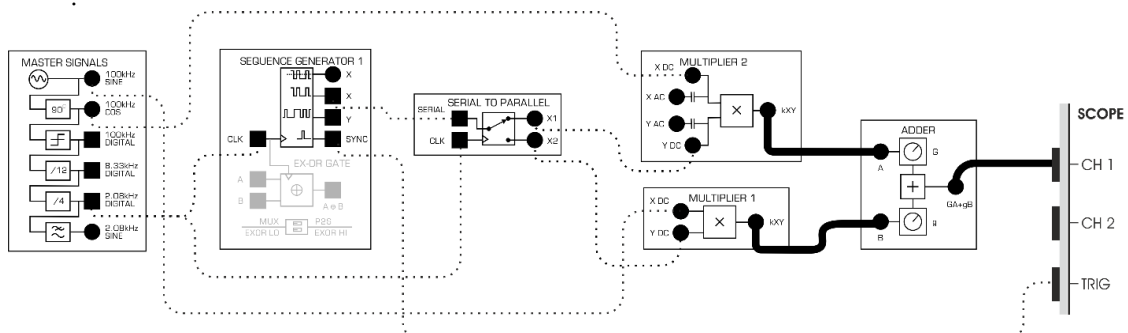


Figure 6.4c 1: Patching for addition of dual BPSK channels

Hints

```
% Quadrature Phase-Shift Keying (QPSK) modulation and demodulation
% Parameters
fs = 1000;           % Sampling frequency
fc = 50;             % Carrier frequency
T = 1;              % Duration of the signal in seconds
Bits Per Symbol = 2; % Number of bits per symbol
% Generate random digital data
. . . . .
% Modulation
. . . . .
% Create QPSK signal
. . . . .
% Demodulation
. . . . .
% Low-pass filter to remove high-frequency components
. . . . .
% Perform phase detection to demodulate the signal
. . . . .
% Plot original data, QPSK signal, and demodulated data figure;
. . . . .
```

Try

1. In a QPSK modulation system, if the carrier frequency is 2 MHz and the bit rate is 500 kbps, calculate the duration of each symbol (symbol period) and the phase shifts used in QPSK modulation.
2. In a QPSK modulation system, if the carrier frequency is 10 MHz and the symbol rate is 2 Mbps, calculate the duration of each symbol (symbol period) and the phase shifts used in QPSK modulation.

7. Exercises on Signal Processing in Communication Systems

Signal processing techniques are used in a wide range of applications, including telecommunications, audio and video processing, image processing, speech recognition, and control systems. Some common signal-processing tasks include filtering, noise reduction, compression, and feature extraction.

7.1 Implement: Spectrum of the impulse train

Generating a Spectrum of the impulse train

EMONA Communications board to generate a BPSK signal with the Multiplier module to implement its mathematical model. Digital data for the message is modeled by the Sequence Generator module

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or earbuds

Implement: Setting up the data signal

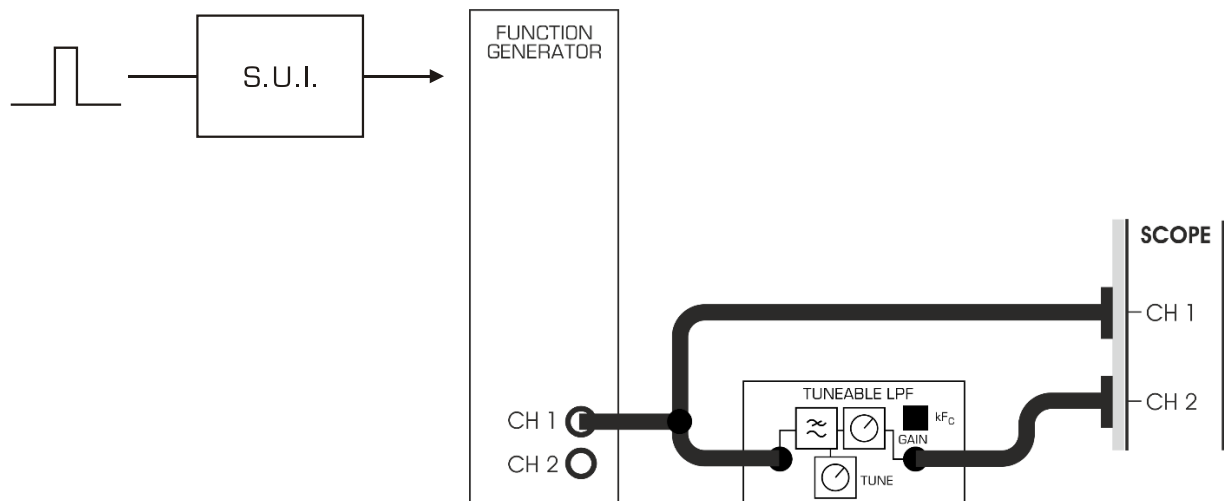


Figure 7.1: Pulse source to System Under Investigation.

Hints

```
% Parameters
% Parameters
fs = 1000;           % Sampling frequency (Hz)
T = 1/fs;           % Sampling period
f0 = 10;            % Fundamental frequency of impulse train (Hz)
duration = 1;       % Duration of the signal (seconds)
% Time vector
. . . . .
% Impulse train
. . . . .
% Compute the spectrum
. . . . .
% Plot the spectrum
figure;
stem(frequencies, abs(X));
title('Spectrum of Impulse Train');
xlabel('Frequency (Hz)');
```

```
ylabel('Magnitude');  
grid on;
```

Try

1. Generate an impulse train with an interval of $T=0.1$ seconds, starting at time $t=0$ and ending at $t=1$ second.
2. Generate an impulse train with an interval of $T=0.2$ seconds, starting at time $t=0$ and ending at $t=2$ second.

7.2 Implement: Spectrum of the filtered impulse train

Generating a Spectrum of the impulse train

EMONA Communications board to generate a BPSK signal with the Multiplier module to implement its mathematical model. Digital data for the message is modeled by the Sequence Generator module

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or earbuds

Implement: Setting up the data signal

Hints

```
% Parameters  
% Parameters  
fs = 1000;           % Sampling frequency (Hz)  
T = 1/fs;           % Sampling period  
f0 = 10;            % Fundamental frequency of impulse train (Hz)  
duration = 1;       % Duration of the signal (seconds)  
filter_freq = 50;   % Cut-off frequency of the filter (Hz)  
% Time vector  
. . . . .  
x = zeros(size(t));  
x(1:round(fs/f0):end) = 1;  
% Design a simple low-pass filter  
. . . . .  
% Apply the filter to the impulse train  
filtered_signal = filter(filter_coeff, 1, x);  
% Compute the spectrum of the filtered signal  
. . . . .
```



```

frequencies = (0:N-1) * fs / N;
% Plot the spectrum
. . . . .

```

7.3 Implement: Spectrum of pseudorandom sequence

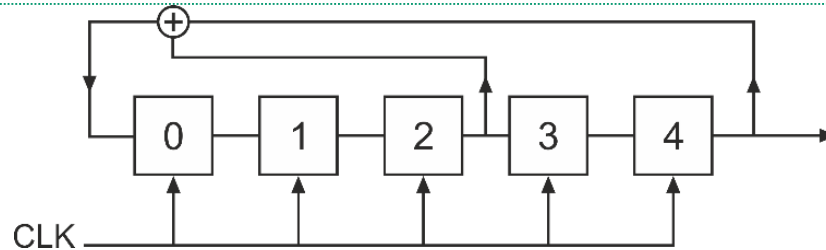


Figure 7.3a: Block diagram for the of DxlQ-45GSEQUENCE GENERATOR 1

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or ear-buds

Implement: Setting up the data signal

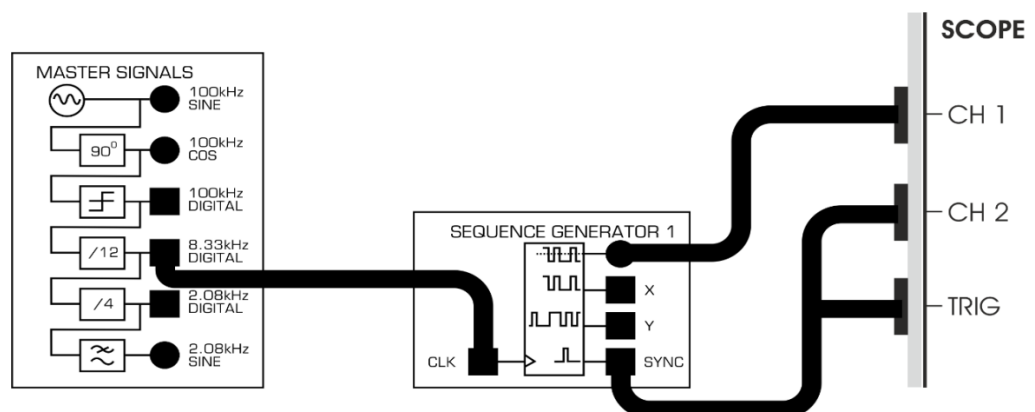


Figure 2.3b Patching diagram for PN sequence from SEQUENCE GENERATOR 1

Hints

```

% Generate a pseudorandom sequence
% Parameters
N = 1024; % Length of the sequence
seq = randn(1, N); % You can use rand() for a uniform distribution
% Compute the FFT
fft_result = fft(seq);
% Compute the single-sided spectrum

```

```

. . . . .
% Plot the spectrum
figure;
plot(f, P1);
title('Single-Sided Amplitude Spectrum of Pseudorandom Sequence');
xlabel('Frequency (Hz)');
ylabel('Amplitude');

```

7.4 Implement: Analog noise generation (AWGN)

Implementation of Analog noise generation (AWGN)

Required Tools and Technology

Platform: NI ELVIS III

Hardware: EMONA Communications Board

Components used in this lab:

- Four BNC to 2mm banana-plug leads
- Assorted 2mm banana-plug patch leads
- Set of headphones or earbuds

Implement: Setting up the data signal



Figure 7.4: Block diagram for PRBS generated analog noise

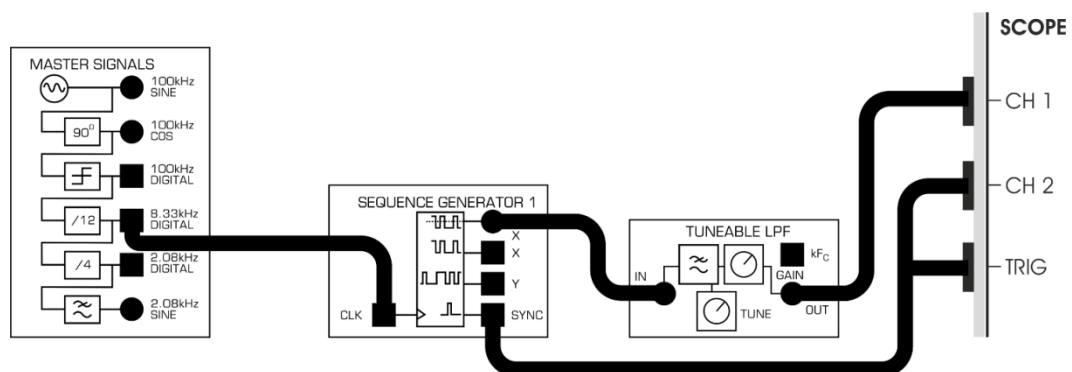


Figure 7.4b: noise Patching diagram for PRBS analog noise

8. Exercises on signal filtering

In the field of signal processing, a filter is a device or process that, completely or partially, suppresses unwanted components or features from a signal. This usually means removing some frequencies to suppress interfering signals and to reduce background noise.

8.1 BASICS OF FILTERING

Implement: FILTERING

Filtering in signal processing is a technique used to modify or extract information from a signal by allowing certain frequencies to pass while attenuating others. Here are the basics of filtering in signal processing:

Hints

```
% Parameters
Fs = 1000;           % Sampling frequency
t = 0:1/Fs:1;        % Time vector
f = 10;              % Frequency of the signal
% Generate a simple sine wave signal
. . . . .
% Add random noise
. . . . .
% Plot the noisy signal
. . . . .
% Design a simple moving average filter
. . . . .
% Plot the filtered signal
. . . . .
grid on;
```

Try

1. Design low pass filter for signal filtering in communication systems.
2. Design high pass filter for signal filtering in communication systems.

8.2 Implement: Frequency domain analysis in signal filtering.

Here's an example MATLAB code that demonstrates frequency domain analysis and filtering of a signal using a band-stop filter:

Hint

```
% Parameters
Fs = 1000;           % Sampling frequency
t = 0:1/Fs:1;        % Time vector
f1 = 50;             % Frequency of the signal

% Generate a simple sine wave signal
. . . . .
% Add noise at specific frequency
. . . . .
% Perform Fourier Transform to analyze frequency spectrum
. . . . .
% Plot the frequency spectrum of the noisy signal
```

```

. . . . .
ylabel('Amplitude');
grid on;
% Design a band-stop filter to remove noise at 'noise_freq'
Fstop1 = noise_freq - 10; % Lower cutoff frequency
Fstop2 = noise_freq + 10; % Upper cutoff frequency
Fpass1 = noise_freq - 20; % Lower passband edge
Fpass2 = noise_freq + 20; % Upper passband edge
Astop = 80; % Stopband attenuation in dB
Apass = 1; % Passband ripple in dB
Fs = 1000; % Sampling frequency
% Design the filter using FIR band-stop method
. . . . .
% Apply the filter to the noisy signal
filtered_signal = filtfilt(d, noisy_signal);
% Perform Fourier Transform of the filtered signal
. . . . .
% Plot the frequency spectrum of the filtered signal
. . . . .
grid on;

```

Try

1. Perform frequency domain analysis for signal filter filtering using band pass filter.
2. Perform frequency domain analysis for signal filtering using band stop filter.

8.3 Design of filter optimization in signal filtering.

Hints

```

% Define parameters
% Parameters
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1; % Time vector
f1 = 50; % Frequency of the signal

% Generate a simple sine wave signal
. . . . .
% Add noise at specific frequency bands
noise_freq1 = 150; % Frequency of noise 1
. . . . .
% Design the filter using FIR band-pass method
. . . . .

```

```

% Apply the filter to the noisy signal
. . . . .
% Perform Fourier Transform of the filtered signal
. . . . .
% Plot the frequency spectrum of the filtered signal
. . . . .
grid on;
% Plot the filtered signal in time domain
. . . . .
grid on;

```

Try

1. Design a Butterworth low-pass filter with the following specifications: Cutoff frequency: 1 kHz, Maximum pass band ripple: 0.5 dB, Stop band attenuation: 40 dB. Design the filter and plot its frequency response.
2. Design a Chebyshev Type I high-pass filter with the following specifications: Cutoff frequency: 500 Hz, Maximum passband ripple: 1 dB, Stopband attenuation: 30 dB Design the filter and visualize its frequency response.

9. Exercises on Wireless Communication Systems

Wireless communication (or just wireless, when the context allows) is the transfer of information (telecommunication) between two or more points without the use of an electrical conductor, optical fiber or other continuous guided medium for the transfer. The most common wireless technologies use radio waves.

9.1 Implement: Okumura model

Hints

```

// Okumura model
% System Parameters
frequency = 900e6;      % Frequency in Hz
transmitterHeight = 50; % Transmitter height in meters
receiverHeight = 10;    % Receiver height in meters
distance = 100:100:1000; % Distance between transmitter and receiver in meters %
% Two-ray Channel Model
Pt = 1;                 % Transmitted power in Watts
Gt = 1;                 % Transmitter antenna gain
Gr = 1;                 % Receiver antenna gain
L = 1;                  % System loss
% Calculate received power using Two-ray channel model
. . . . .
% Calculate path loss using Okumura-Hata model
. . . . .
grid on;

```

Try

1. Is the Hata model used for signal strength prediction

9.2 Multipath fading

Hints

```
/**
 % Simulation parameters
 numSamples = 1000; % Number of samples
 numPaths = 3;      % Number of multipath paths
 fadePower = 0.5;   % Fading power
 % Generate Rayleigh fading channel coefficients
 . . . . .
 % Generate transmitted signal
 . . . . .
 % Plot the transmitted and received signals
 . . . . .
```

Try

1. Perform Channel Modeling using MATLAB by writing program.
2. What are the effects of multipath fading?

10. Exercises on Data transmission

Data transmission is the transfer of data from one digital device to another. This transfer occurs via point-to-point data streams or channels.

10.1 Scrambler

Hints

```
% Parameters
% System Parameters
B = 4 ;
% bits in shift register
. . . . .
% shift register seed . 11 dec = 1011 % bin
. . . . .
% connection register . 9 dec = 1001 bin
. . . . .
% binary 1000
. . . . .
% binary 0001
. . . . .
% select rightmost bit of shift register
. . . . .
%main shift register shifted right and OR in MSB
. . . . .
```

Try

1. Perform the experiment on Signal Bandwidth Analysis in MATLAB

10.2 Implement: To design modulation: Spread Spectrum – DSSS Modulation & Demodulation

Hints

```
% Parameters
% Simulation parameters
data = [1 0 1 0 1 1 0 0];
% Original data signal
spreadingCode = [1 1 0 1];
% Spreading code
spreadingFactor = length(spreadingCode);
% DSSS Modulation
. . . . .
% DSSS Demodulation
. . . . .
% Display Results
disp('Original Data:');
disp(data);
disp('Demodulated Data:');
disp(demodulatedSignal);
```

Try

1. Input data 111000111

11. Exercises on Multiplexing and Demultiplexing techniques.

In telecommunications and computer networking, multiplexing (sometimes contracted to muxing) is a method by which multiple analog or digital signals are combined into one signal over a shared medium. The aim is to share a scarce resource – a physical transmission medium.

11.1 Implement: Time division multiplexing.

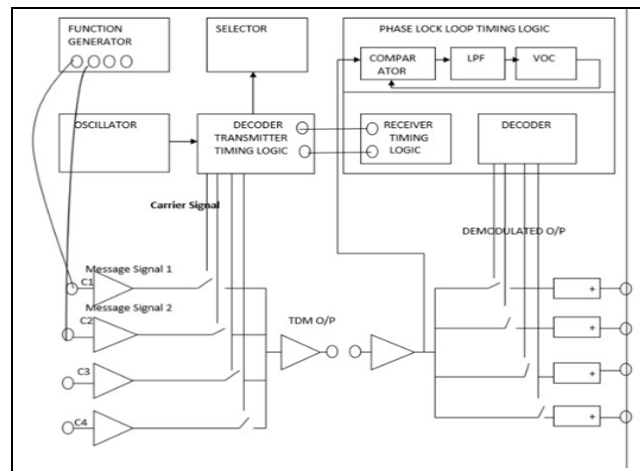


Figure 11.1: Time division multiplexing

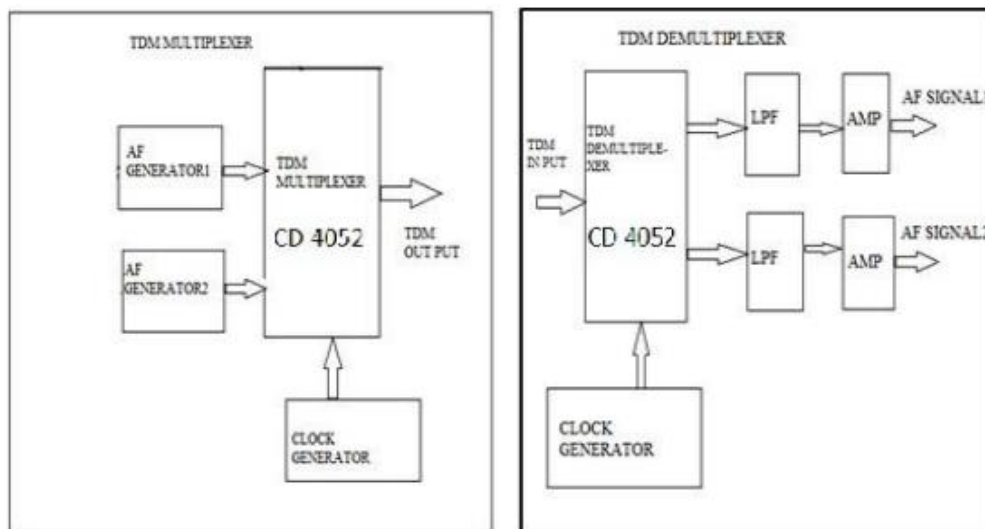


Figure 11.2: Block diagram of Time division multiplexer and demultiplexer

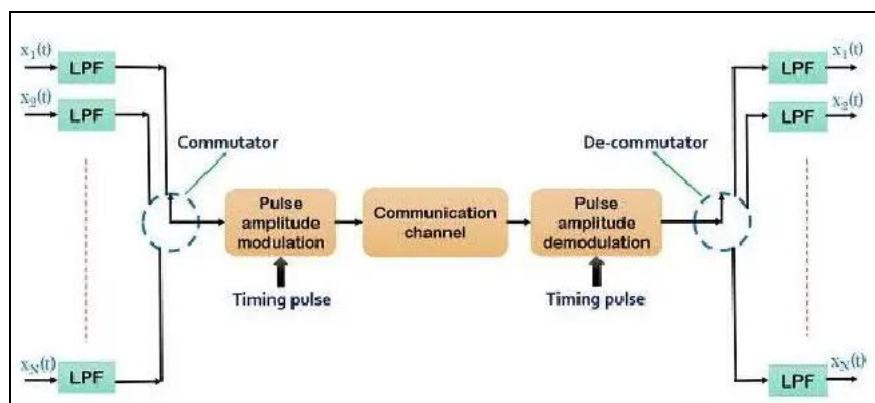


Figure 11.3: Block diagram of Time division multiplexing system.

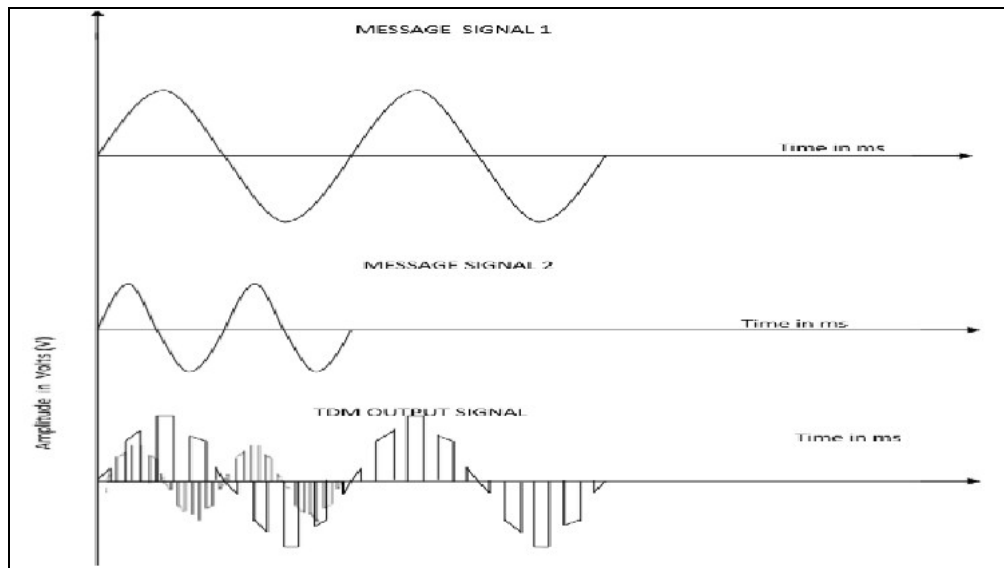


Figure 11.4: Input and output waveforms of Time division multiplexing.

Hints

% Time Division Multiplexing (TDM) MATLAB Code

```

close all;
clear all;
% Signal generation
x=0:.5:4*pi; % signal taken upto 4pi
sig1=8*sin(x); % generate 1st sinusoidal signal
l=length(sig1);
sig2=8*triang(1); % Generate 2nd triangular signal
% Display of Both Signal
. . . . .

% Display of Both Sampled Signal

. . . . .
% TDM of both quantize signal
tdmsig=reshape(sig,1,2*l);
% Display of TDM Signal
. . . . .
% Demultiplexing of TDM Signal
. . . . .
% display of demultiplexed signal
. . . . .

```

Try

1. What is the time allocated for each channel if the number of samples per frame is 4 and the frame rate is 100 frames/sec?
2. Design a TDMA system for three users with variable data rates: User 1 requires 20 milliseconds, User requires 15 milliseconds, and User 3 requires 25 milliseconds for transmission. Allocate time slots efficiently to accommodate these varying data rates within a frame of 60 milliseconds.
3. Given a TDMA system with six users sharing a channel, each allocated a time slot of 10 milliseconds in a frame of 60 milliseconds; calculate the maximum achievable throughput in bits per second assuming ideal conditions (no overhead, no gaps between time slots).

11.2 Implement: Frequency division multiplexing (FDM).

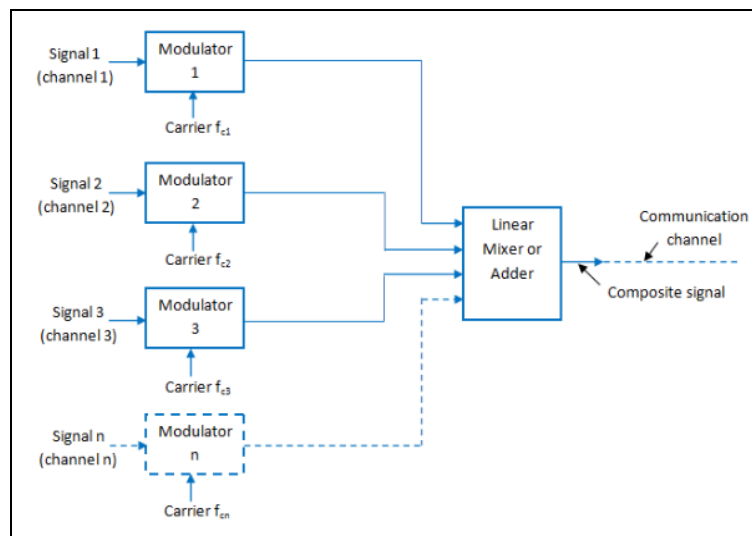


Figure 11.2.3: FDM Transmitter

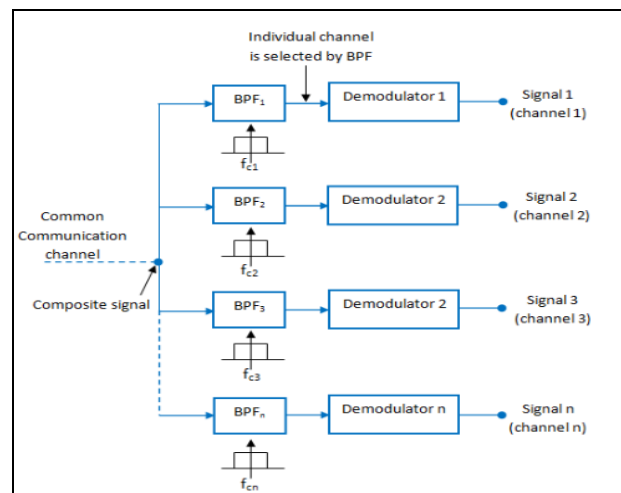


Figure 11.2.4: FDM Receiver

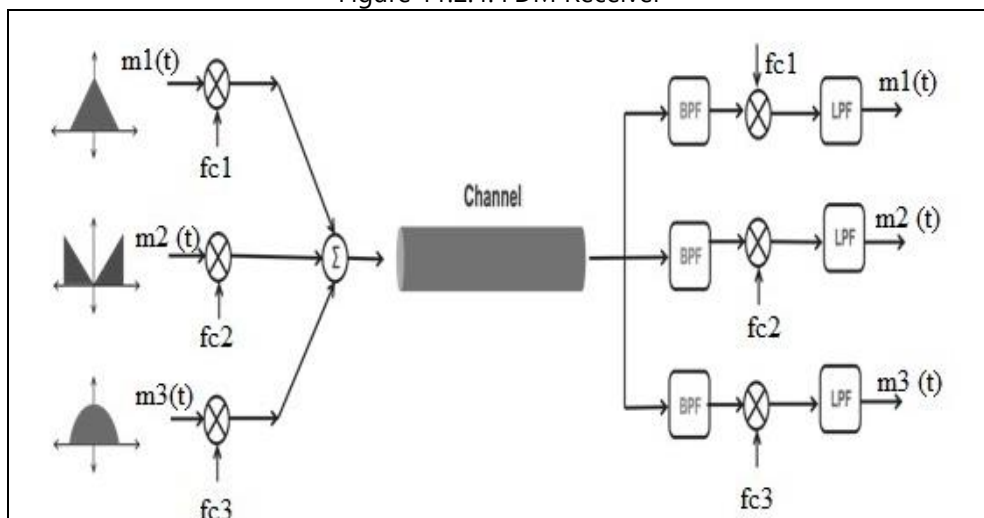


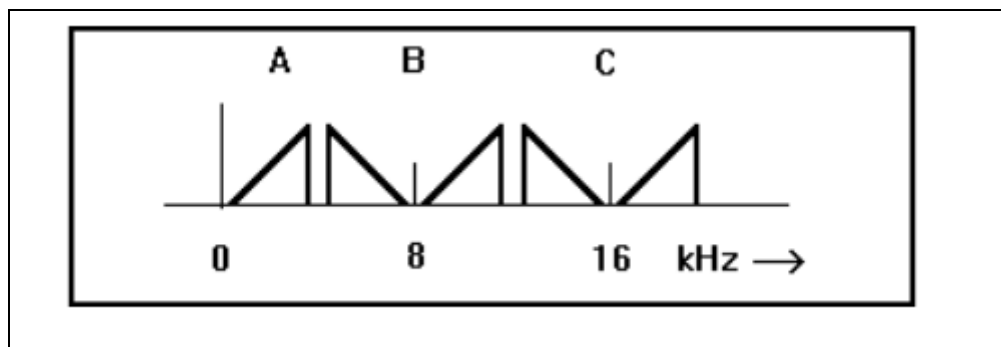
Figure 11.2.5: Block diagram of FDM

Hints

```
/** Frequency division multiplexing */  
% Frequency Division Multiplexing (FDM) Example in MATLAB  
  
clc;  
close all;  
clear all;  
  
Fs = 100; % sampling frequency  
t = [0 : 2 * Fs+1] / Fs ;  
  
. . . . .  
  
. . . . .  
  
title ( 'Spectrum of signal 1' ) ;  
xlabel ( 'frequency' ) ;  
ylabel ( 'magnitude' ) ;  
  
. . . . .  
  
% frequency multiplexing  
  
. . . . .  
  
% frequency demultiplexing  
  
. . . . .
```

Try

To avoid the need for SSB generators to make the FDM signals, an alternative scheme could generate the spectrum shown below. Could a DSBSC demodulator be used in principle to recover the three channels independently? Can you suggest a possible practical problem(s) with this scheme?



12. Exercises on ADAPTIVE MODULATION

Adaptive modulation is a technique used to modify the modulation of a signal based on the characteristics of the channel between the transmitter and receiver. It can also be used by an end user to manually adapt a modulated signal for their own modulation purposes.

Implementation:

Hints:

```
% ADAPTIVE MODULATION OFDM LTE in MATLAB

% Parameters
clc;
clear all;
close all;

%assigning the variables
N = 1536;
bw = 15*10^6;
T = 1/bw;
SNR = 10:1:30;
BER_tar = 10^(-3);
% finding the target snr
. . . . .
%Indoor delay
. . . . .
. . . . .
% Finding the multipath channel response
. . . . .
. . . . .

. . . . .
ylabel('Spectral efficiency in bits/Hz');
legend('Indoor','Outdoor');
```

Try

LTE supports various modulation schemes such as QPSK, 16QAM, and 64QAM. Calculate the maximum data rates (Mbps) supported by each modulation scheme assuming a bandwidth of 20 MHz and 2x2 MIMO.

Hints:

```
% ADAPTIVE DELTA MODULATION in MATLAB

% Parameters
clc;
clear all;
close all;
%% Input Signals,
% m(t).
t = 0 : 2*pi/100 : 2*pi;
mt = sin(t); % Sine wave.
mt = sin(t) + 2; % Sine wave with non-zero DC value.%% Step Size, S.
quantizationLevels = 16;
S = (max(mt) - min(mt)) / quantizationLevels; %% Modulate.
totalSamples = length(mt) - 1;
mqt = zeros(1, totalSamples);
dk = zeros(1, totalSamples); % Output Binary Sequence, d[k].
dt = 0; % Difference,d(t) = m(t) - mq(t).
Sk = zeros(1, totalSamples); % Step Size, S[k].
```

```

for n = 2 : totalSamples
    dt = mt(n) - mqt(n);
    if(dt >= 0)dk(n) = 1;
    else dk(n) = -1;
    end
    Sk(n) = abs(Sk(n-1))*dk(n) + S*dk(n-1);
    mqt(n+1) = mqt(n) + Sk(n);
end

dk = zeros(1, totalSamples);
% Output Binary Sequence, d[k].
dt = 0; % Difference, d(t) = m(t) - mq(t).
Sk = zeros(1, totalSamples); % Step Size, S[k].
for n = 2 : totalSamples
    dt = mt(n) - mqt(n);
    if(dt >= 0)
        dk(n) = 1;
    else dk(n) = -1;
    end
    Sk(n) = abs(Sk(n-1))*dk(n) + S*dk(n-1);
    mqt(n+1) = mqt(n) + Sk(n);
end
%% Display Modulation Result.
plot(t, mt, 'r', 'LineWidth', 2);
hold on;
stairs(t, mqt, 'k', 'LineWidth', 2);
axis([t(1) t(end) (min(min(mqt), min(mt)) - 0.5) (max(max(mqt), max(mt)) + 0.5)]);
title('Adaptive Delta Modulation', 'FontSize', 14);
xlabel('Time');
ylabel('Amplitude');
legend('Input Signal, m(t)', 'Modulated Signal, m_q(t)');
grid on;

```

Try

Design a simple algorithm for adaptive modulation in LTE based on CQI feedback from the UE (User Equipment). How would you optimize this algorithm for better performance?

13. Exercises on Generation of Noises.

Noise is any unwanted signal, random or deterministic, which interferes with the faithful reproduction of the desired signal in a system. This interfering signal is usually noticed as random fluctuations in voltage or current tending to obscure and mask the desired signals.

13.1. Generation of White noise:

Hints

```

% White noise

% Parameters
duration = 5; % Duration of the white noise signal in seconds
fs = 44100; % Sampling frequency in Hz (adjust as needed)

% Generate white noise
white_noise = randn(1, duration * fs);

% Normalize to have zero mean and unit variance
white_noise = white_noise / std(white_noise);

```

```
% Plot the white noise signal
. . . . .
% Play the white noise (optional)
% sound(white_noise, fs)
```

Try

1. Consider a random signal with the mean of 1 and the variance of 2. What is the power of this signal? How do you calculate signal power in MATLAB.
2. How do you use `fftshift()` with `pwelch()` in MATLAB®? Give an example.

13.2. Generation of Brown noise:

Hints

```
% Brown noise

% Parameters
duration = 5; % Duration of the white noise signal in seconds
fs = 44100; % Sampling frequency in Hz (adjust as needed)

% Generate Brown noise
. . . . .
% Normalize to have zero mean and unit variance
. . . . .
% Plot and play the brown noise
. . . . .
% Play the brown noise (optional)
% sound(brown_noise, fs);
```

Try

1. Consider a random signal with the mean of 3 and the variance of 5. What is the power of this signal? How do you calculate signal power in MATLAB
2. How do you use `repmat ()` with `reshape ()` in MATLAB? Give an example.

13.3. Generation of Pink noise:

Hints:

```
% Pink noise
% Parameters
duration = 5; % Duration of the white noise signal in seconds
fs = 44100; % Sampling frequency in Hz (adjust as needed)
% Generate Pink noise
pink_noise = pinknoise(duration * fs);
% Normalize to have zero mean and unit variance
pink_noise = pink_noise / std(pink_noise);
% Plot and play the pink noise
. . . . .
% Play the pink noise (optional)
% sound(pink_noise, fs);
```

Try

1. Consider a random signal with the mean of 2 and the variance of 3. What is the power of this signal? How do you calculate signal power in MATLAB.
2. How do you use `hist ()` with `hist3 ()` in MATLAB? Give an example.

14. Exercises on case study: communication system project

Voice Control Home Automation System Using Arduino and HC-05

Hardware Setup:

1. Arduino Uno and Genuino Uno
2. HC-05 Bluetooth Module
3. Jumper Wires
4. Two AC Bulbs (Red, Yellow)
5. Relay

```
/* arduino*/
Transmitter (Node 1):
#include <SPI.h>

String voice;
    int RED = 2;
    int YELLOW = 3;
    void RedOn()
    { digitalWrite (RED, LOW);
    }
    void RedOff(){
        digitalWrite (RED, HIGH);
    }
    void YellowOn(){
        digitalWrite (YELLOW, LOW);
    }
    void YellowOff()
    {
        digitalWrite (YELLOW, HIGH);
    }
    void allon()
    {
        digitalWrite (RED, LOW);
        digitalWrite (YELLOW, LOW);
    }
    void alloff()
    {
        digitalWrite (RED, HIGH);
        digitalWrite (YELLOW, HIGH);
    }
    . . . . .
    . . . . .
    }
    . . . . .
```

Try

Write a MATLAB program to generate an AM signal with a carrier frequency of 1 kHz, a modulation index of 0.5, and a message signal frequency of 100 Hz. Plot the modulated signal in the time domain.

15. Final Notes

The only way to learn programming is program, program and program on challenging problems. The problems in this tutorial are certainly NOT challenging. There are tens of thousands of challenging problems available – used in training for various programming contests (such as International

Collegiate Programming Contest (ICPC), International Olympiad in Informatics (IOI)). Check out these sites:

- MathWorks Certification Program
- National Institute of Electronics and Information Technology (<https://reg.nielitchennai.edu.in>)

Student can have any one of the following certifications:

- NPTEL – Analog Communications
- NPTEL – Digital Communications

V. TEXT BOOKS:

1. Simon Haykin, “Analog and Digital Communications”, TMH, 3rd edition, 2003.
2. B.P.Lathi, “Analog and Digital Communications”, Wiley, 3rd edition, 2003

VI. REFERENCE BOOKS:

1. George Kenedy, “Communication Systems” TMH, 1st edition, 2004.
2. F.E. Terman, “Electronic and Radio Engineering”, McGraw-Hill, 4th edition, 1955.

VII. ELECTRONICS RESOURCES:

1. <https://www.slac.stanford.edu/pubs/slacpubs/7500/slac-pub-7731.pdf>
2. [https:// www.onlinecourses.nptel.ac.in/noc20_ee63/preview](https://www.onlinecourses.nptel.ac.in/noc20_ee63/preview)
3. <http://nptel.ac.in/courses/analogcommunications>

VIII. MATERIALS ONLINE

1. Course template
2. Lab manual