

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY



IT6313 DIGITAL COMMUNICATION LABORATORY
LAB MANUAL

Prepared by
Ms.B.Saravanya/AP-III *B.Saravanya*
Mrs.M.Seema/AP-III *M.Seema*

R.Perumal Raja
Verified by,
Dr.R.Perumal Raja
HoD/IT

List of Experiments

Exp.No.	Title of the Experiment
1	Signal sampling and reconstruction
2	Amplitude modulation and demodulation
3	Frequency modulation and demodulation
4	Pulse code modulation and demodulation
5	Delta and Adaptive Delta modulation
6	Line coding and decoding techniques
7	Generation and detection of FSK
8	Generation and detection of PSK
9	Time Division Multiplexing and demultiplexing (TDM)
10(a)	Simulation of FSK using MATLAB
10(b)	Simulation of PSK using MATLAB
10(c)	Simulation of DPSK using MATLAB
11	Simulation of Error Control coding schemes using MATLAB
12	Simulation of spread spectrum communication using MATLAB
13	Simulation of Frequency Division Multiplexing (FDM) using MATLAB
14	Simulation of communication link using SCILAB
15	Simulation of Linear Block codes using MATLAB

Exp. No. 1	SIGNAL SAMPLING AND RECONSTRUCTION
Date :	

AIM :

To study the process of sampling and to reconstruct the signals at the receiver.

APPARATUS REQUIRED:

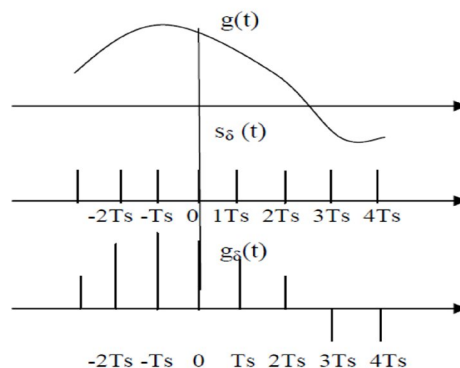
- 1, Sampling trainer kit
2. Power supply
3. Patch chords
4. CRO

THEORY:

A message signal may originate from a digital or analog source. If the message signal is analog in nature, then it has to be converted into digital form before it can be transmitted by digital means. The process by which the continuous-time signal is converted into a discrete-time signal is called Sampling. Sampling operation is performed in accordance with the sampling theorem.

Sampling Theorem For Low-Pass Signals:-

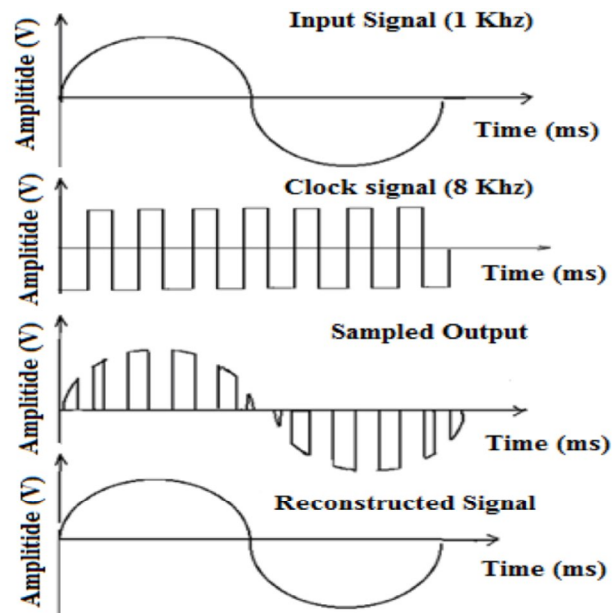
Statement:- "If a band-limited signal $g(t)$ contains no frequency components for $|f| > W$, then it is completely described by instantaneous values $g(kT_s)$ uniformly spaced in time with period $T_s \leq 1/2W$. If the sampling rate, f_s is equal to the Nyquist rate or greater ($f_s \geq 2W$), the signal $g(t)$ can be exactly reconstructed.

**PROCEDURE:**

1. Connections are made as per the kit diagram.
2. Connect the power supply cable at the POWER IN connector.
3. Switch ON the power.
4. Here POT-P1 is for frequency adjust (Sine and square1- signals), POT-P2 is for frequency adjust (square2 and triangle- signals).
5. ADD0, ADD1 and ADD2 test points are provided to monitor the addresses.
6. Obtain the output at output terminal.

TABULAR COLUMN:

Signal	Amplitude (V)	Time period(ms)	Frequency
Input Signal			
Clock signal			
Output			

MODEL GRAPH :**RESULT:**

Thus the sampling process was studied and the message signal was reconstructed.

Exp. No. 2	AMPLITUDE MODULATION AND DEMODULATION
Date:	

AIM:

To transmit a modulating signal after amplitude modulation using AM transmitter and receive the signal back after demodulating using AM receiver.

APPARATUS REQUIRED:

1. AM trainer kit
2. CRO
3. Patch cards
4. Probe

THEORY:

Modulation is defined as the process by which some characteristics of a carrier signal is varied in accordance with a modulating signal. The base band signal is referred to as the modulating signal and the output of the modulation process is called as the modulation signal. Amplitude Modulation is a process by which amplitude of the carrier signal is varied in accordance with the instantaneous value of the modulating signal, but frequency and phase of carrier wave remains constant. The envelope of the modulating wave has the same shape as the base band signal provided the following two requirements are satisfied.

1. The carrier frequency f_c must be much greater than the highest frequency components f_m of the message signal $m(t)$ i.e. $f_c \gg f_m$
2. The modulation index must be less than unity. if the modulation index is greater than unity, the carrier wave becomes over modulated.

The modulating and carrier signal are given by

$$V_m(t) = V_m \sin\omega_m t$$

$$V_C(t) = V_C \sin\omega_C t$$

The modulation index is given by, $m_a = V_m / V_C$.

$$V_m = V_{\max} - V_{\min} \text{ and } V_C = V_{\max} + V_{\min}$$

The amplitude of the modulated signal is given by,

$$V_{AM}(t) = V_C (1 + m_a \sin\omega_m t) \sin\omega_C t$$

Where

V_m = maximum amplitude of modulating signal

V_C = maximum amplitude of carrier signal

V_{\max} = maximum variation of AM signal

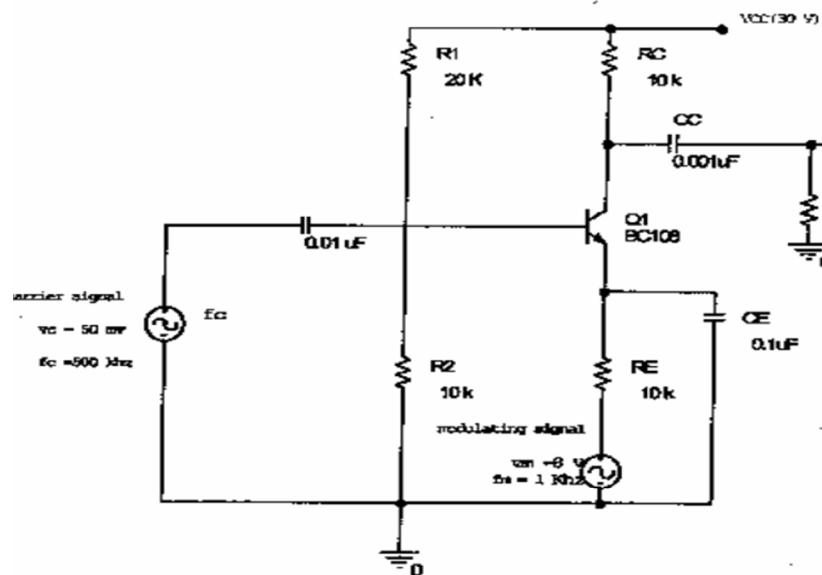
V_{\min} = minimum variation of AM signal

PROCEDURE:

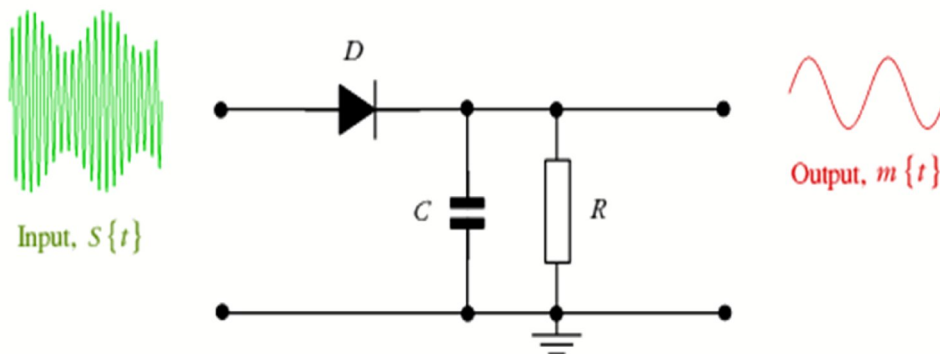
1. Connections are made as shown in kit diagram.
2. Connect power supply cable at the POWER IN connector.

3. Switch ON the power.
4. Here POT-P1 is for frequency adjust (Audio Sig), POT-P2@ is for amplitude adjust(AUDIO-SIG), POT-P3 is for carrier frequency and POT-P4 is for amplitude adjust(CARRIER).
5. Connect the AUDIO-SIG to AUDIO-IN, CARRIER -SIG to CARRIER-IN and monitor the modulated signal AM-TX of the MODULATON block.
6. Connect the modulated signal (AM-TX) to (AM-RX) for demodulation and monitor the demodulated signal at AUDIO-OUT and compare this signal with AUDIO-SIG.

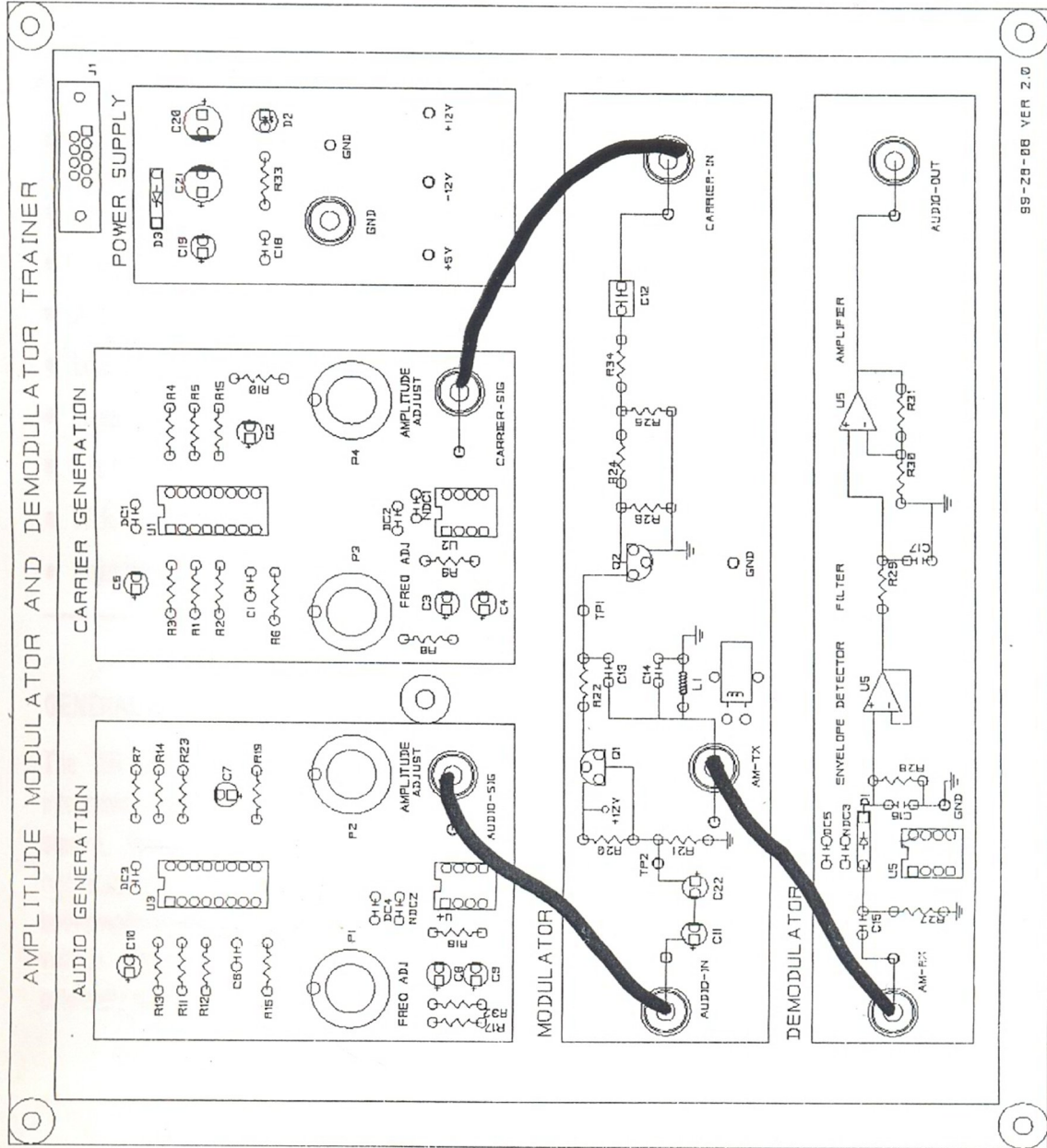
MODULATOR:



DEMODULATOR:



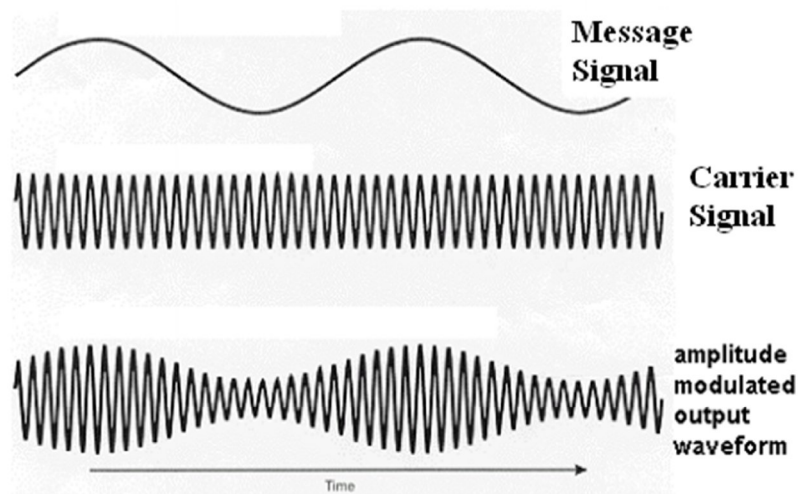
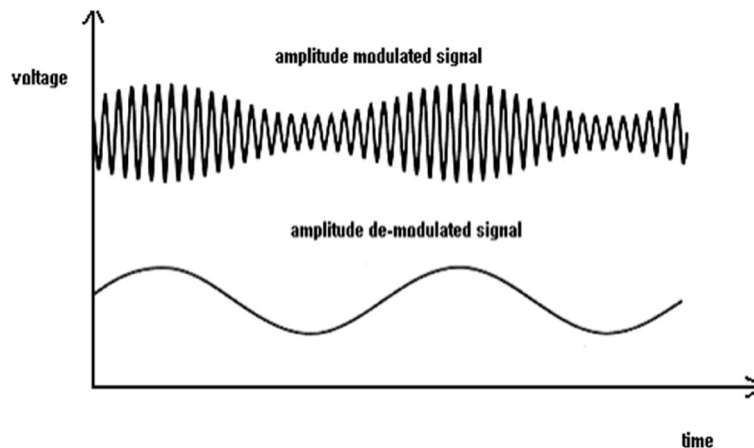
KIT DIAGRAM:



99-ZB-08 VER 2.0

TABULATION:

Waveform	Amplitude (V)	Time Period (msec)	Frequency(Hz)
Message			
Carrier			
Modulated			
Demodulated			

MODEL GRAPH:**Amplitude Modulation:****Amplitude Demodulation:****RESULT:**

Thus the AM signal was transmitted using AM trainer kit and the AM signal detected using AM detector kit.

Exp.No. :3	FREQUENCY MODULATION AND DEMODULATION
Date:	

AIM:

To construct a frequency modulated waveform using FM transmitter and demodulate it using FM receiver.

APPARATUS REQUIRED:

1. FM trainer kit
2. CRO
3. Patch cards
4. Probe

THEORY:

In an FM signal, the frequency of the signal is changed by the modulation (baseband) signal while its amplitude remains the same. Frequency modulation (FM) is the standard technique for high-fidelity communications as is evident in the received signals of the FM band (88-108 MHz) vs. the AM band (450-1650 KHz). The main reason for the improved fidelity is that FM detectors, when properly designed, are not sensitive to random amplitude variations which are the dominant part of electrical noise (heard as static on the AM radio). Frequency modulation is not only used in commercial radio broadcasts, but also in police and hospital communications, emergency channels, TV sound, wireless (cellular) telephone systems, and radio amateur bands above 30 MHz.

The bandwidth of an FM signal depends on the modulation index (m_f), and is approximated by the well-known Carson's Rule:

$$BW = 2 (\delta + f_m(\max))$$

where $f_m(\max)$ is the maximum frequency of the modulating signal.

The factor (2) in the equation is to account for both the upper and lower sidebands (left and right of the carrier).

There are types of FM : (1) Narrow band FM (2) Wide band FM

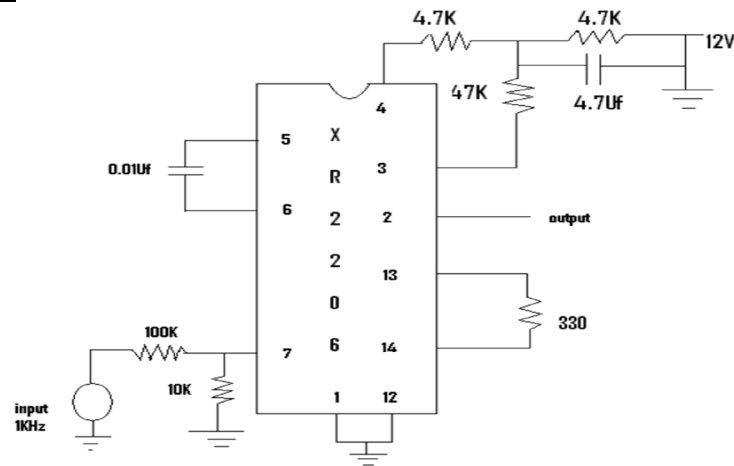
A narrow band FM is the FM wave with a small bandwidth. The modulation index m_f of narrow band FM is small as compared to one radian. Hence, the spectrum of narrow band FM consists of the carrier and upper sideband and a lower sideband. Narrow band systems are used in FM mobile communications such as police wireless, ambulances, taxicabs etc. FM signal can be detected using frequency discriminator, Foster Seeley and ratio detectors. For large values of modulation index m_f , the FM wave ideally contains the carrier and an infinite number of sidebands located symmetrically around the carrier. Such a FM wave has infinite bandwidth and hence called as wideband FM. The modulation index of wideband FM is higher than 1.

PROCEDURE:

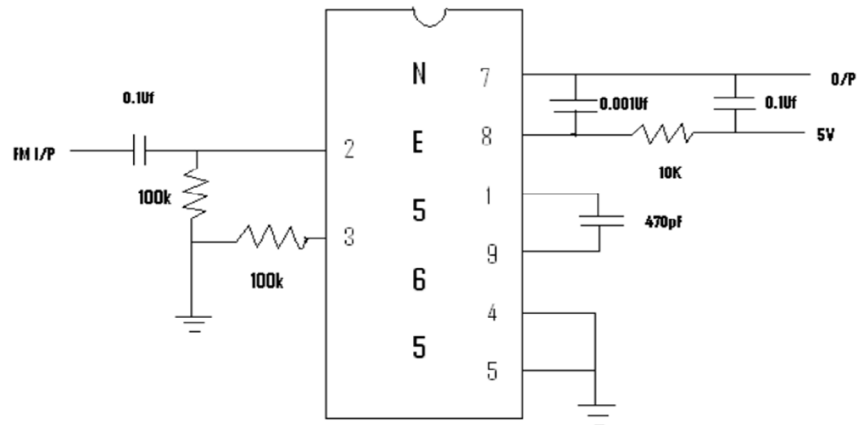
1. Connections are made as per the kit diagram.

2. Connect the power supply cable at the POWER IN connector.
3. Switch ON the power.
4. Here POT-P1 is for frequency adjust (AC-SIG), POT-P2 is for level adjust (DC-SIG) and POT-P3 is for VCO adjust.
5. Give the AC-SIG to SIGNAL -IN and the modulated signal may be monitored at PWM-TX in the MODULATOR block.
6. Give the modulated signal FM-TX for demodulation at FM-RX and monitor the demodulated signal at DEMOD-OUT in the DEMODULATOR block.
7. Give the demodulated signal to FILTER IN for filtering. Monitor the final output at AC SIG-OUT.
8. Similarly give the DC-SIGNAL and check the modulated signal at FM-TX, by varying the POT-P2. You can observe the frequency modulation at FM-TX and the demodulated signal can be monitored at DC SIG-OUT of FILTER block.

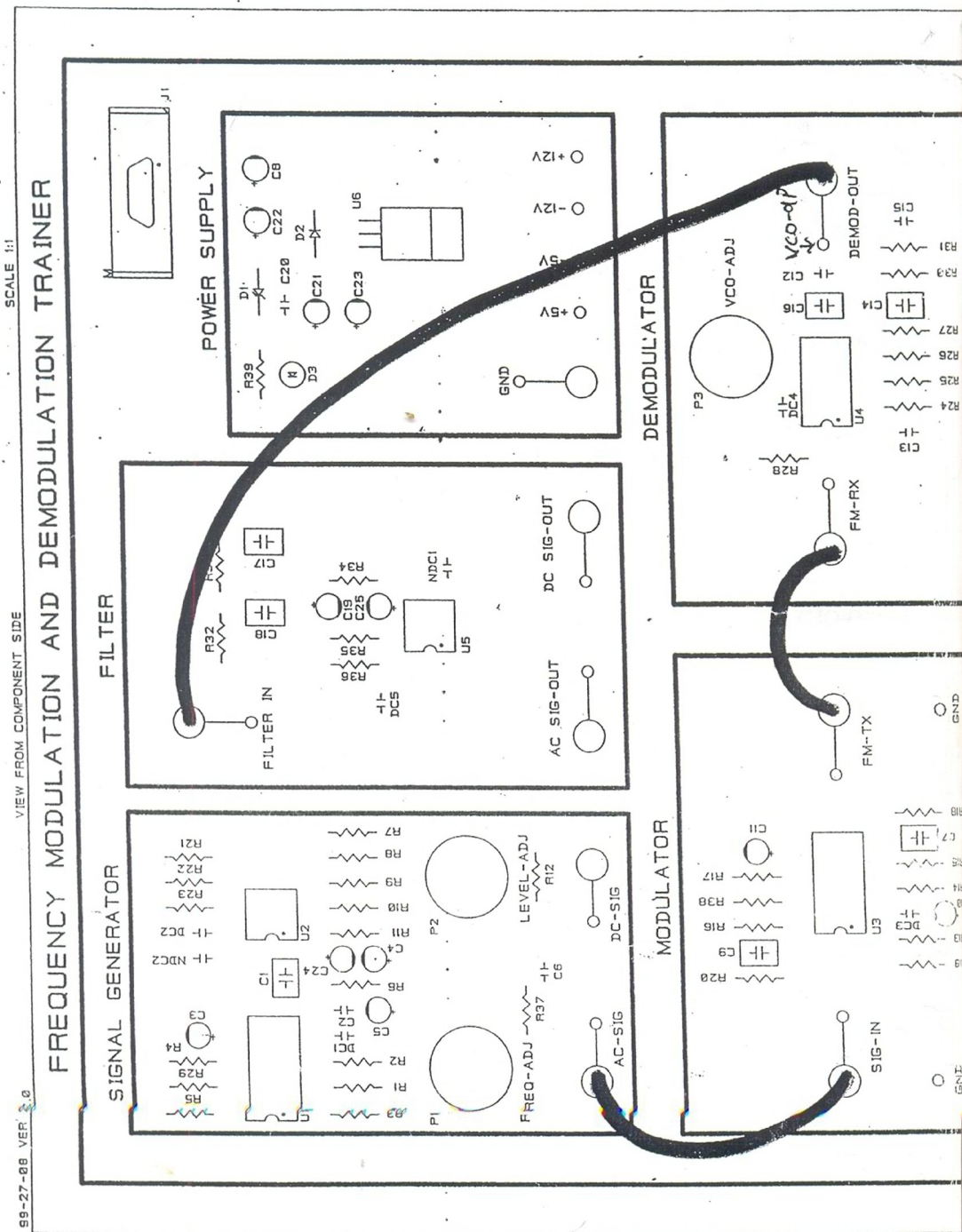
MODULATOR:



DEMODULATOR :

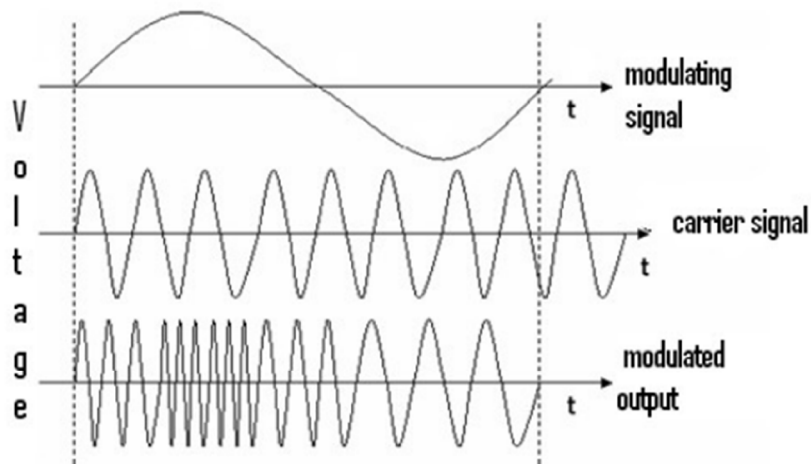
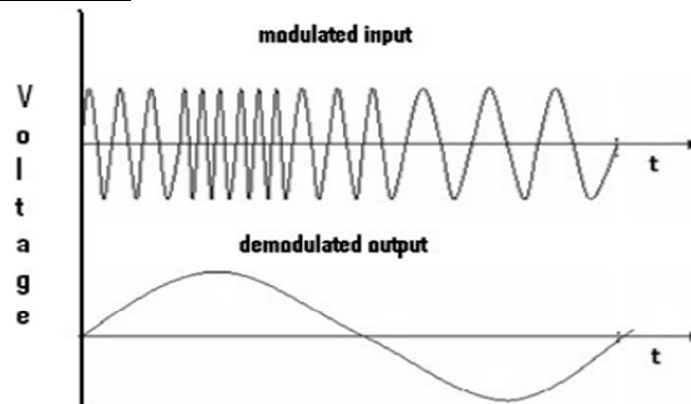


KIT DIAGRAM:



TABULATION:

Name of the signal	Amplitude (V)	Time Period (msec)	Frequency
Message			
Carrier			
Modulated			
Demodulated			

MODEL GRAPH:**Frequency Modulation:****Frequency demodulation:****RESULT:**

Thus the FM signal was transmitted using FM trainer kit and detected using FM detector kit.

Exp.No.:4	PULSE CODE MODULATION AND DEMODULATION
Date:	

AIM:

To generate a PCM signal using PCM modulator and detect the message signal using PCM demodulator.

APPARATUS REQUIRED:

1. PCM kit
2. CRO
3. Connecting probes

THEORY:

Pulse code modulation is a process of converting an analog signal into digital. The voice or any data input is first sampled using a sampler (which is a simple switch) and then quantized. Quantization is the process of converting a given signal amplitude to an equivalent binary number with fixed number of bits. This quantization can be either midtread or mid-raise and it can be uniform or non-uniform based on the requirements. For example in speech signals, the higher amplitudes will be less frequent than the low amplitudes. So higher amplitudes are given less step size than the lower amplitudes and thus quantization is performed non-uniformly. After quantization the signal is digital and the bits are passed through a parallel to serial converter and then launched into the channel serially.

At the demodulator the received bits are first converted into parallel frames and each frame is de-quantized to an equivalent analog value. This analog value is thus equivalent to a sampler output. This is the demodulated signal.

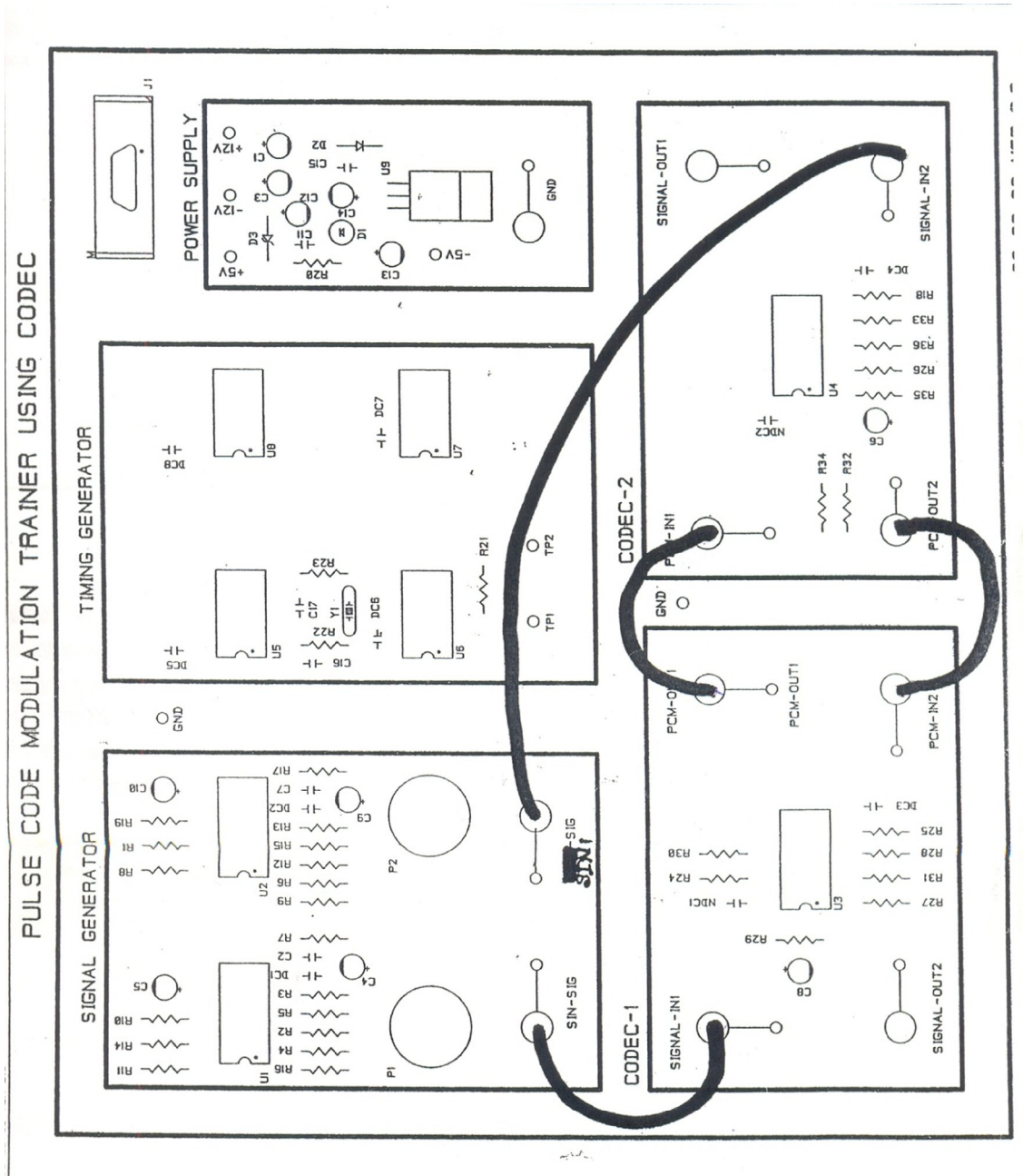
In the kit this is implemented differently. The analog signal is passed through a ADC (Analog to Digital Converter) and then the digital codeword is passed through a parallel to serial converter block. This is modulated PCM. This is taken by the Serial to Parallel converter and then through a DAC to get the demodulated signal. The clock is given to all these blocks for synchronization. The input signal can be either DC or AC according to the kit. The waveforms can be observed on a CRO for DC without problem.AC also can be observed but with poor resolution.

PROCEDURE:

1. Connections are made as per the kit diagram.
2. Connect the power supply cable at the POWER IN connector.
3. Switch ON the power.
4. Here POT-P1 is for frequency adjust of SIN-SIG1 and POT-P2 is for frequency adjust of SIN-SIG2.

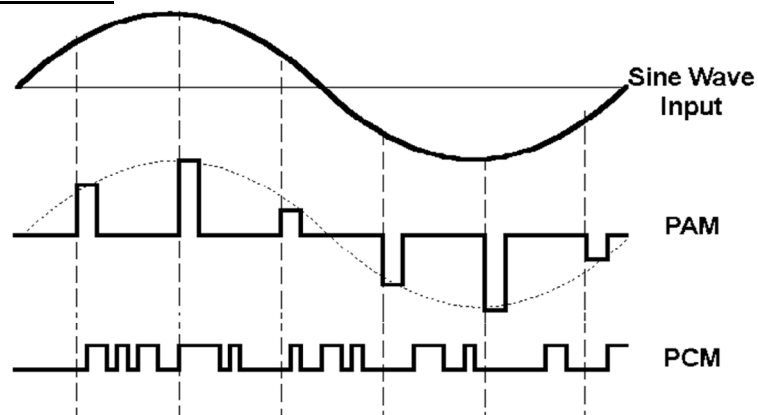
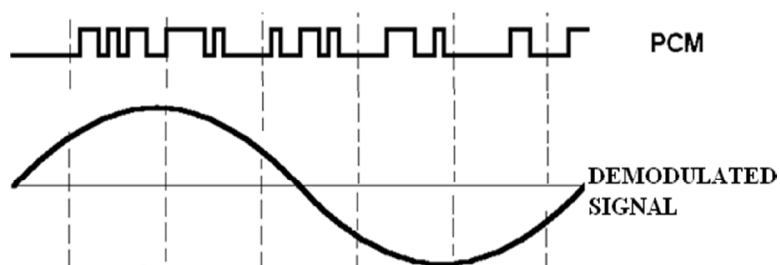
5. Connect the SIN-SIG to SIGNAL-IN1 and monitor the PCM output of channel 1 at PCM-OUT1 in the CODEC-1 block.
6. Connect the PCM output to PCM-IN1 for demodulation and monitor the demodulated signal at SIGNAL-OUT1 in the CODEC-2 BLOCK. Compare the input SIN-SIG with demodulated signal.
7. Similarly connect the SIN-SIG2 and check the PCM OUTPUT and also see the demodulated output for the same.

KIT DIAGRAM:



TABULATION:

S.No	Name of the signal	Amplitude (V)	Time period (msec)	Frequency
1	Message			
2	Carrier			
3	Modulated			
4	Demodulated			

MODEL GRAPH**Pulse code modulation:****Pulse Code demodulation:****RESULT:**

Thus the PCM signal was generated using PCM modulator and the message signal was detected from PCM signal by using PCM demodulator.

Exp.No.:5	DELTA AND ADAPTIVE DELTA MODULATION
Date:	

AIM:

To convert original analog signal to digital signal and reconstruct the analog signal back using delta and adaptive delta modulation.

APPARATUS REQUIRED:

1. DM kit
2. CRO
3. Connecting probes

THEORY:

Delta modulation is the DPCM technique of converting an analog message signal to a digital sequence. The difference signal between two successive samples is encoded into a single bit code. A present sample of the analog signal $m(t)$ is compared with a previous sample and the difference output is level shifted, i.e. a positive level (corresponding to bit 1) is given if difference is positive and negative level (corresponding to bit 0) if it is negative. The comparison of samples is accomplished by converting the digital to analog form and then comparing with the present sample. This is done using an Up counter and DAC as shown in block diagram. The delta modulated signal is given to up counter and then a DAC and the analog input is given to OPAMP and a LPF to obtain the demodulated output. When DM is not able to keep up with the input signal a phenomenon called slope overload is exhibited.

Adaptive delta modulation (ADM) reduces the slope-overload and granular distortions encountered in delta modulators (DM) to a greater extent. In each of the step-size algorithms of ADM, the processor detects the pattern to see if the delta modulator is operating in the granular noise region, in which case it produces an alternating1010..... Pattern, or in the slope over load region in which case it produces an all-1 or all-0 pattern.

- If the ADM senses a ...1010.... pattern, it decreases the step-size, and
- If it senses1111.... or0000.... pattern, it increases the step-size.

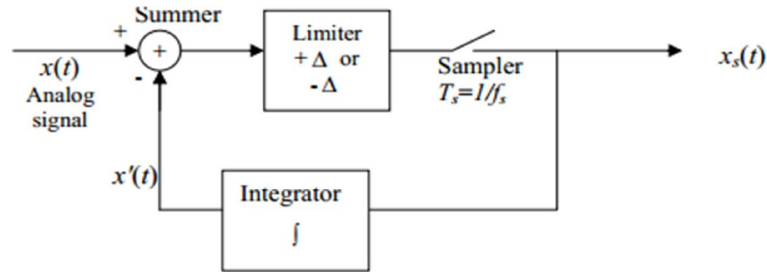
PROCEDURE:

1. Switch on the kit. Connect the clock signal and the modulating input signal to the modulator block. Observe the modulated signal in the CRO.
2. Connect the DM output to the demodulator circuit. Observe the demodulator output on the CRO.
3. Also observe the DAC output on the CRO.
4. Change the amplitude of the modulating signal and observe the DAC output.

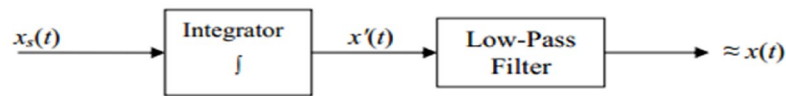
Notice the slope overload distortion. Keep the tuning knob so that the distortion is gone. Note this value of the amplitude. This is the minimum required value of the amplitude to overcome slope overload distortion.

5. Calculate the sampling frequency required for no slope overload distortion. Compare the calculated and measured values of the sampling frequency.

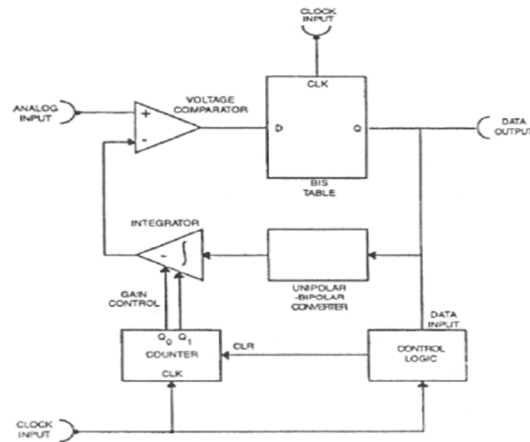
DELTA MODULATOR:



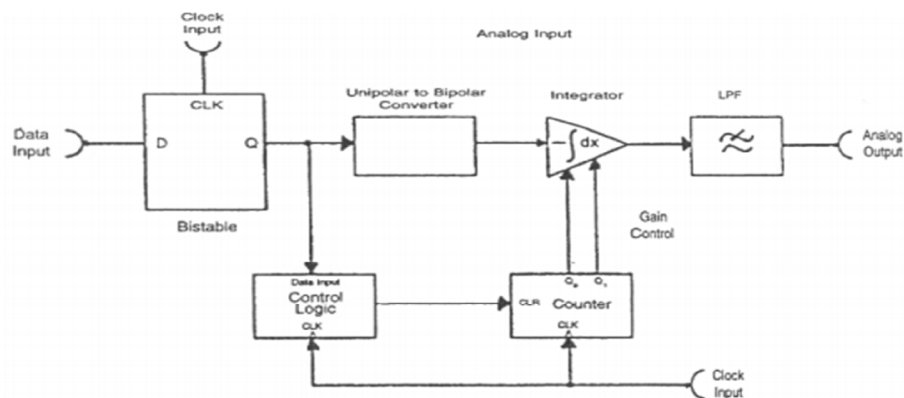
DELTA DEMODULATOR:



ADAPTIVE DELTA MODULATOR:

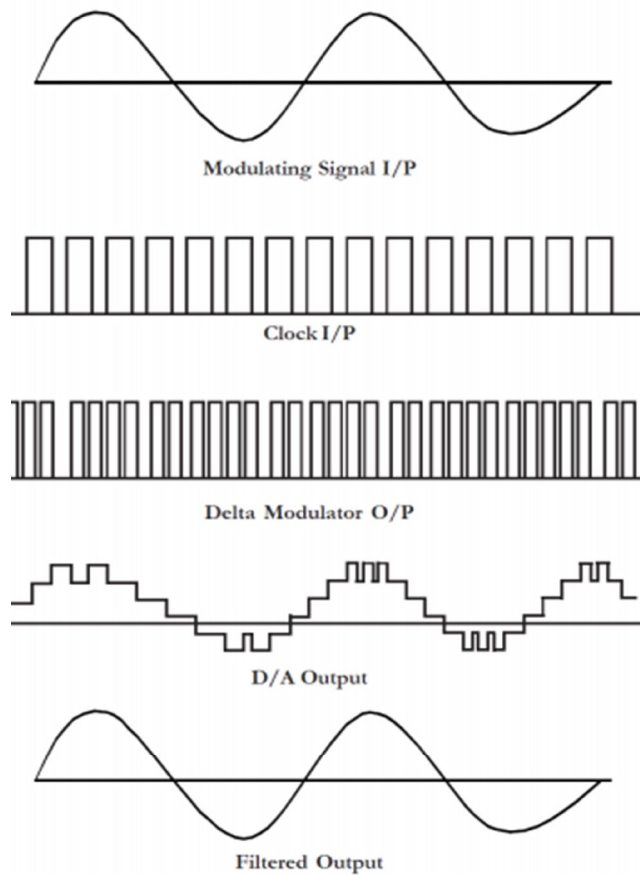


ADAPTIVE DELTA DEMODULATOR:



TABULATION:

S.No	Name of the signal	Amplitude (V)	Time period (msec)	Frequency
1	Message			
2	Carrier			
3	Modulated			
4	Demodulated			

MODEL GRAPH:**RESULT:**

Thus the delta and adaptive modulation and demodulation were performed successfully.

Exp.No.:6	LINE CODING AND DECODING TECHNIQUES
Date:	

AIM:

To study the different line coding techniques used in digital transmission.

APPARATUS REQUIRED:

1. Communication trainer kit
2. Multi Output Power Supply.
3. Patch cords.
4. DSO/CRO

THEORY:

PCM binary digits should be represented by electrical pulses in order to transmit them through a base band channel. The most commonly used PCM popular data formats are being realized here. Line coding refers to the process of representing the bit stream (**1**'s and **0**'s) in the form of voltage or current variations optimally tuned for the specific properties of the physical channel being used. The selection of a proper line code can help in so many ways: One possibility is to aid in clock recovery at the receiver. A clock signal is recovered by observing transitions in the received bit sequence, and if enough transitions exist, a good recovery of the clock is guaranteed, and the signal is said to be self-clocking.

Another advantage is to get rid of DC shifts. The DC component in a line code is called the bias or the DC coefficient. Unfortunately, most long-distance communication channels cannot transport a DC component. This is why most line codes try to eliminate the DC component before being transmitted on the channel. Such codes are called DC balanced, zero-DC, zero-bias, or DC equalized. Some common types of line encoding in common-use nowadays are unipolar, polar, bipolar, Manchester, MLT-3 and Duobinary encoding. These codes are explained here:

Unipolar (Unipolar NRZ and Unipolar RZ):

Unipolar is the simplest line coding scheme possible. It has the advantage of being compatible with TTL logic. Unipolar coding uses a positive rectangular pulse $p(t)$ to represent binary **1**, and the absence of a pulse (i.e., zero voltage) to represent a binary **0**. Two possibilities for the pulse $p(t)$ exist: Non-Return-to-Zero (NRZ) rectangular pulse and Return-to-Zero (RZ) rectangular pulse. The difference between Unipolar NRZ and Unipolar RZ codes is that the rectangular pulse in NRZ stays at a positive value (e.g., +5V) for the full duration of the logic **1** bit, while the pulse in RZ drops from +5V to 0V in the middle of the bit time.

A drawback of unipolar (RZ and NRZ) is that its average value is not zero, which means it creates a significant DC-component at the receiver (see the impulse at zero frequency in the corresponding power spectral density (PSD) of this line code. The disadvantage of unipolar RZ compared to unipolar NRZ is that each rectangular pulse in RZ

is only half the length of NRZ pulse. This means that unipolar RZ requires twice the bandwidth of the NRZ code.

Bipolar(Bipolar NRZ and Bipolar RZ):

Binary 1's are represented by alternating positive or negative values. The binary 0 is represented by a zero level. The term pseudoternary refers to the use of 3 encoded signal levels to represent two-level (binary) data. This is also called alternate mark inversion (AMI) signalling.

Bipolar RZ: Also called RZ-AMI, where AMI denotes alternate mark (binary 1) inversion

Bipolar NRZ: Also called NRZ-M, where M denotes inversion on mark (binary 1)

Manchester Encoding:

In Manchester code each bit of data is signified by at least one transition. Manchester encoding is therefore considered to be self-clocking, which means that accurate clock recovery from a data stream is possible. In addition, the DC component of the encoded signal is zero. Although transitions allow the signal to be self-clocking, it carries significant overhead as there is a need for essentially twice the bandwidth of a simple NRZ or NRZI encoding

PROCEDURE:

Unipolar RZ encoding and decoding:

1. Connect the PRBS (test point P5) to test point P7.
2. Connect test point P8 to P18.
3. Set switch SW1 in RZ position.
4. Set the potentiometer P1 in minimum position.
5. Switch ON the power supply.
6. Press switch SW2 once.
7. Display the encoded signal at test point P8 on one channel of CRO and decoded signal at test point P20 on second channel of CRO.

Unipolar NRZ encoding and decoding:

1. Connect the PRBS (test point P3) to test point P7.
2. Connect test point P8 to P18.
3. Set switch SW1 in NRZ position.
4. Set the potentiometer P1 in minimum position.
5. Switch ON the power supply.
6. Press switch SW2 once.
7. Display the encoded signal at test point P8 on one channel of CRO and decoded signal at test point P20 on second channel of CRO.

Bipolar RZ encoding and decoding:

1. Connect the PRBS (P8) to point P12 and CLK (P3) to test point P13.
2. Connect test point P14 to P25.

3. Set switch SW1 in RZ position.
4. Set the potentiometer P1 in minimum position.
5. Switch ON the power supply.
6. Press switch SW2 once.
7. Display the encoded signal at test point P14 on channel 1 of CRO and decoded signal at test point P27 on second channel of CRO.

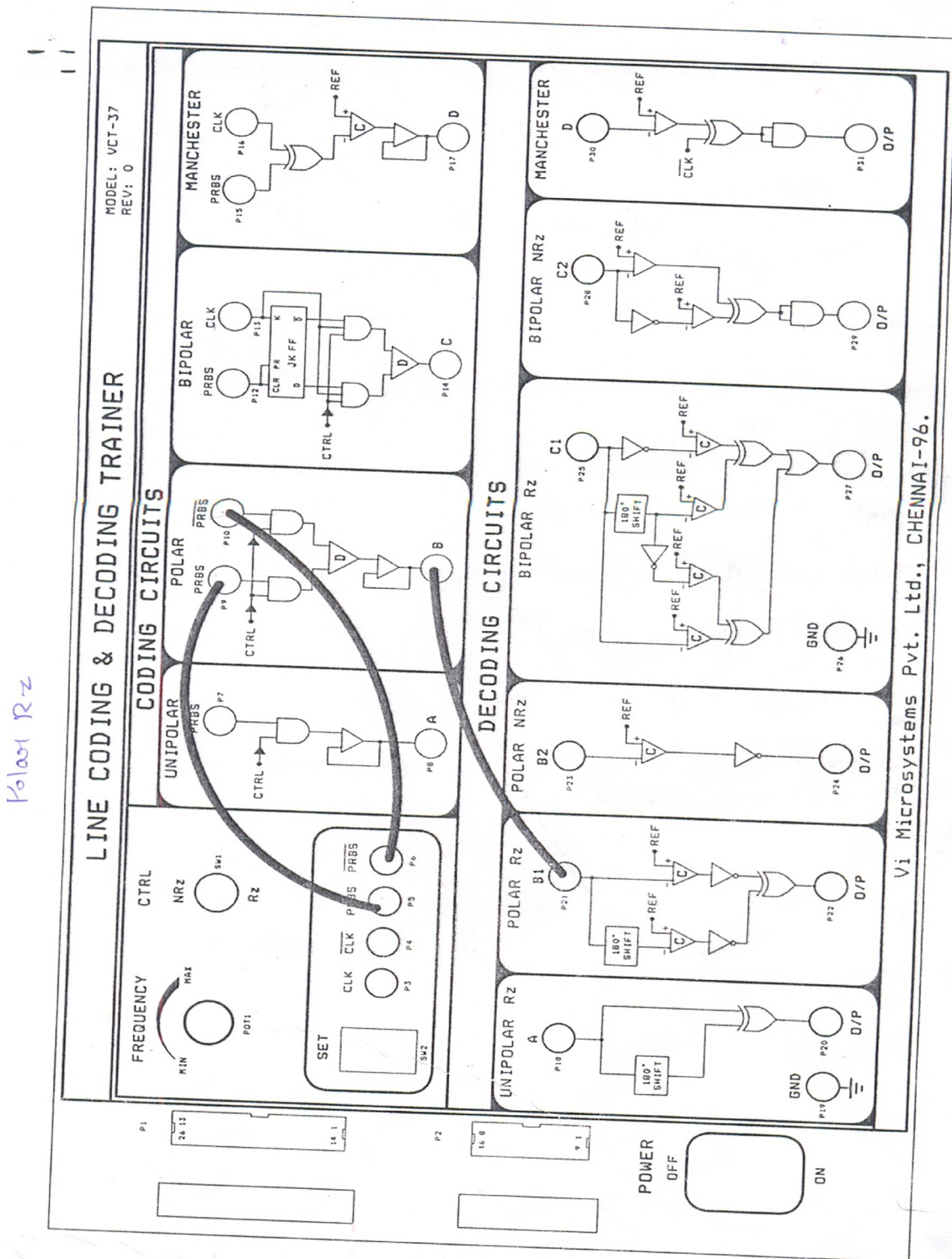
Bipolar NRZ encoding and decoding:

1. Connect the PRBS to point P12 and CLK (P3) to test point P13.
2. Connect test point P14 to P28.
3. Set switch SW1 in NRZ position.
4. Set the potentiometer P1 in minimum position.
5. Switch ON the power supply.
6. Press switch SW2 once.
7. Display the encoded signal at test point P14 on channel 1 of CRO and decoded signal at test point P29 on second channel of CRO.

Manchester encoding and decoding:

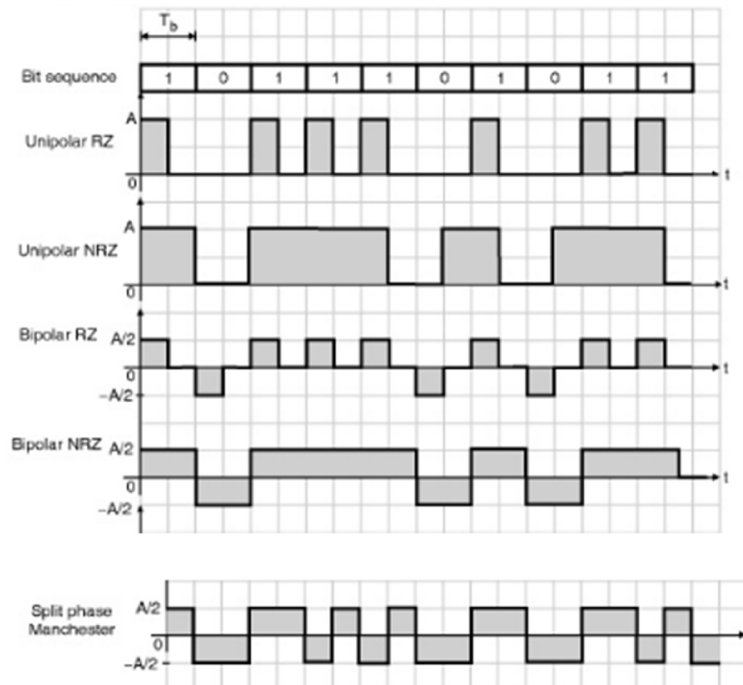
1. Connect the PRBS (P8) to point P15 and CLK (P3) to test point P16.
2. Connect test point P17 to P30.
3. Set the potentiometer P1 in minimum position.
4. Switch ON the power supply.
5. Press switch SW2 once.
6. Display the encoded signal at test point P17 on channel 1 of CRO and decoded signal at test point P31 on second channel of CRO.

KIT DIAGRAM:



TABULATION:

S.No	Name of the signal	Amplitude (V)	Time period (msec)	Frequency
1	Unipolar RZ			
2	Unipolar NRZ			
3	Bipolar RZ			
4	Bipolar NRZ			
5	Manchester			

MODEL GRAPH:**RESULT:**

Thus the different line coding techniques was studied successfully.

Exp.No.:7	GENERATION AND DETECTION OF FSK
Date:	

AIM:

To modulate a message signal using Frequency Shift Keying(FSK) modulator and recover the message signal using Frequency Shift Keying detector.

APPARATUS REQUIRED:

1. FSK kit
2. CRO and connecting probes

THEORY:

Frequency Shift Keying is the process generating a modulated signal from a digital data input. If the incoming bit is 1, a signal with frequency f_1 is sent for the duration of the bit. If the bit is 0, a signal with frequency f_2 is sent for the duration of this bit. This is the basic principle behind FSK modulation.

The operation of FSK modulator is as follows:

- 1, The input binary data is passed through bipolar NRZ encoder
- 2, Its output is applied to two independent balance modulators M1 and M2
- 3, The other input to M1 and M2 are carrier oscillator signals at f_1 and f_2 frequency.
- 4, The output of balance modulator are added together in linear adder circuit.
- 5, The resultant FSK signal with either have frequency f_1 or f_2 .

FSK signal can also be generated using multivibrator as follows:

Basically a 555 timer is used as an Astablemultivibrator, which generates a clock pulse of frequency determined by the values of R and C in this circuit. This is divided by 2, 4, 8 and 16 using 74163 IC, and two of these outputs are used in a NAND logic gates circuit, to generate a FSK modulated wave. To this NAND gates circuit a binary data sequence is also supplied. The circuit operation causes a frequency f_1 for bit 1, and f_2 for bit 0.

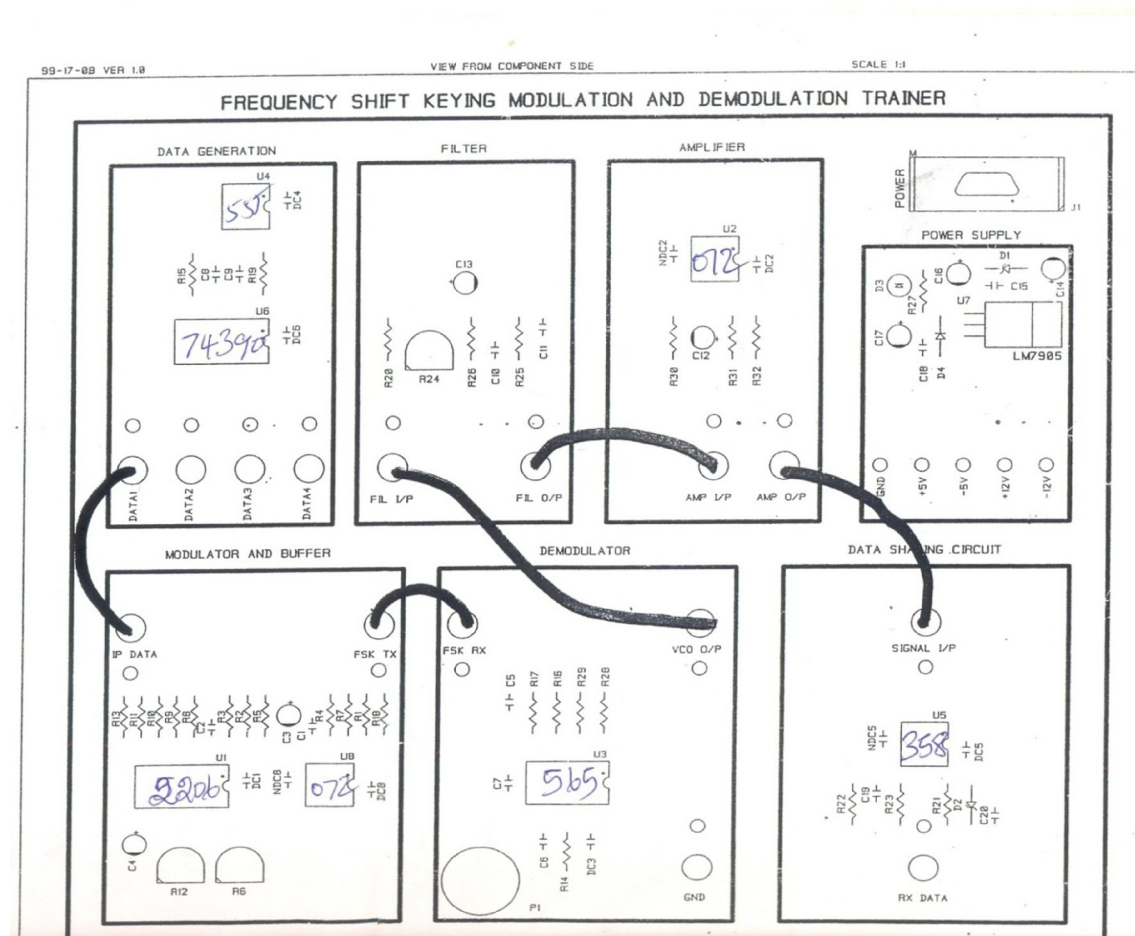
FSK signal can be demodulated either coherently, non-coherently or using PLL. In coherent demodulation, FSK signal is applied to two correlators, comprising of Balance modulators and integrator. The output of two correlator are then subtracted which is then applied to decision device. The decision device then compares the input signal with the preset, then produces the output. In noncoherent demodulator, the received signal is applied to two BPF operating at f_1 and f_2 respectively. The filtered signal is then applied to its corresponding envelope detector and sampling switches. The comparator responds to larger of two signal inputs and detected output is produced. The FSK signal is also applied to PLL circuit. The phase comparator gets locked to input frequency. A corresponding DC signal is produced which is used as DC error voltage to correct the VCO output. Thus the VCO tracks the input frequency between two frequencies f_1 and f_2 . Because there are only frequencies f_1 and f_2 corresponding to binary data 1 and 0. Thus binary message signal is recovered. FSK is mainly used in low speed modems (up to 1200bps) in analog voice band telephone lines. It is also

used in pager systems, HF radio tele type transmission systems, and LAN using coaxial cables.

PROCEDURE:

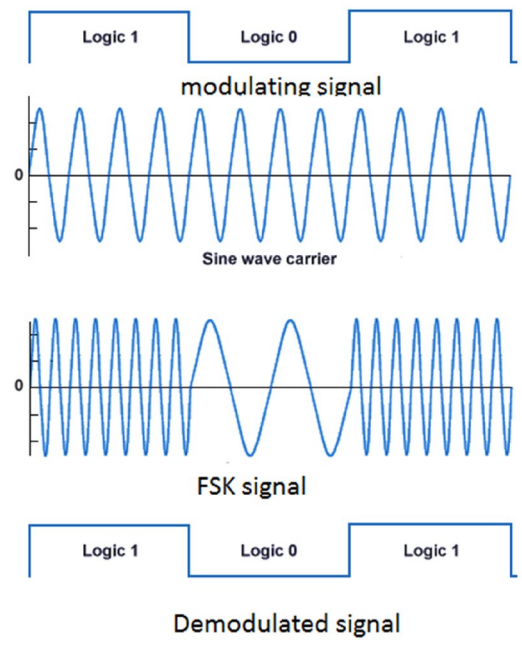
1. Connections are made as per the kit diagram.
2. Connect the power supply cable at the POWER IN connector.
3. Switch ON the power.
4. Here POT-P1 is for frequency tuning.
5. Give the DATA 1 to IP DATA and the modulated signal may be monitored at FSK-TX in the MODULATOR AND BUFFER block.
6. Give the modulated signal (FSK-TX) for demodulation at (FSK-RX) and monitor the demodulated signal at VCO OP of DEMODULATOR block.
7. Give the VCO O/P for filter to FIL I/P of FILTER block and monitor the amplified signal at AMP O/P.
8. Now give the amplified signal for data shaping to SIGNAL I/P and monitor the received signal at RX DATA of DATA SHAPING CIRCUIT block and compare the input signal with received signal.

KIT DIAGRAM:



TABULATION:

S.No	Name of the signal	Amplitude (V)	Time period (msec)	Frequency
1	Message			
2	Carrier			
3	Modulated			
4	Demodulated			

MODEL GRAPH:**RESULT:**

Thus the Frequency Shift Keying signal was generated using FSK modulator and the message signal was detected using FSK detector.

Exp.No.:8	GENERATION AND DETECTION OF PSK
Date:	

AIM:

To modulate a message signal using Phase Shift Keying (PSK) modulator and recover the message signal using Phase Shift Keying detector.

APPARATUS REQUIRED:

1. PSK kit
2. CRO and connecting probes

THEORY:

Phase-shift keying (PSK) is a digital modulation scheme that conveys data by changing (modulating) the phase of a reference signal (the carrier wave). The modulation is impressed by varying the sine and cosine inputs at a precise time. It is widely used for wireless LANs, RFID and Bluetooth communication. BPSK is also called as phase reversal keying or biphasic modulation. When the input binary data changes from 1 to 0 or vice versa, the BPSK output signal phase shifts from 0° to 180° or vice versa.

In a PSK modulator, the binary data sequence of 0s and 1s is converted into bipolar NRZ signal. Its output is then applied to a balance modulator whose other input is from local carrier oscillator. The output of the balance modulator will be

- Inphase with the reference carrier oscillator phase for binary data 1
- 180° out of phase with the reference carrier oscillator phase for binary data 0.

The operation of coherent BPSK demodulator is as follows:

1. The incoming BPSK signal is applied to correlator which consists of a balance modulator and an integrator.
2. The other input signal to the balance modulator is from a locally generated carrier oscillator.
3. The output of the correlator is compared with preset threshold level usually zero volt by the decision device.
 - If the input to the decision device is greater than 0v, then the detected output is 1.
 - If the input to the decision device is less than 0v, then the detected output is 0.

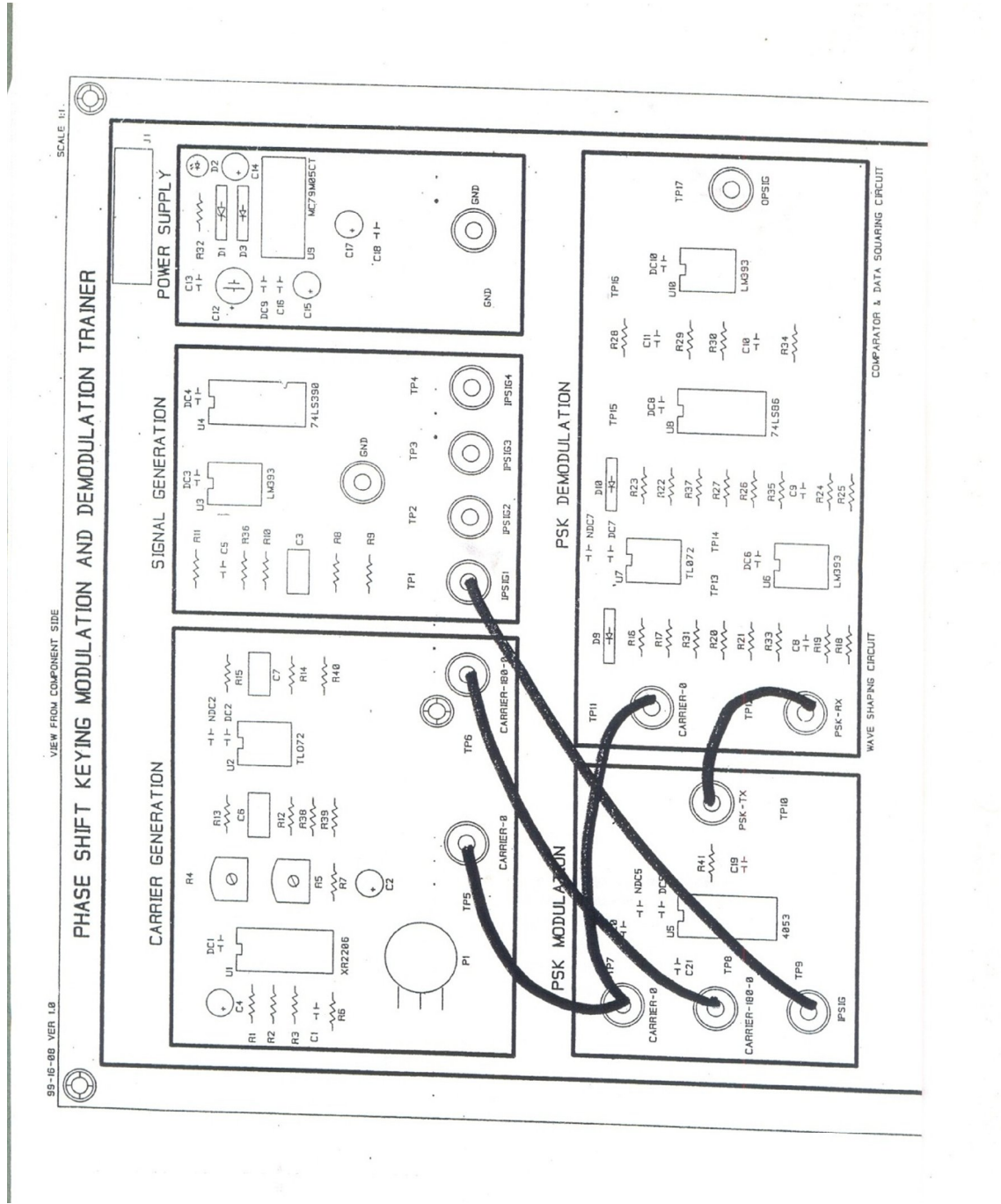
The main advantage of using PSK is that it has good noise immunity, transmission bandwidth is less than FSK, and bandwidth efficiency is better.

PROCEDURE:

1. Connections are made as per the kit diagram.
2. Connect the power supply cable at the POWER IN connector.
3. Switch ON the power.
4. Here POT-P1 is for frequency adjust of carrier wave.

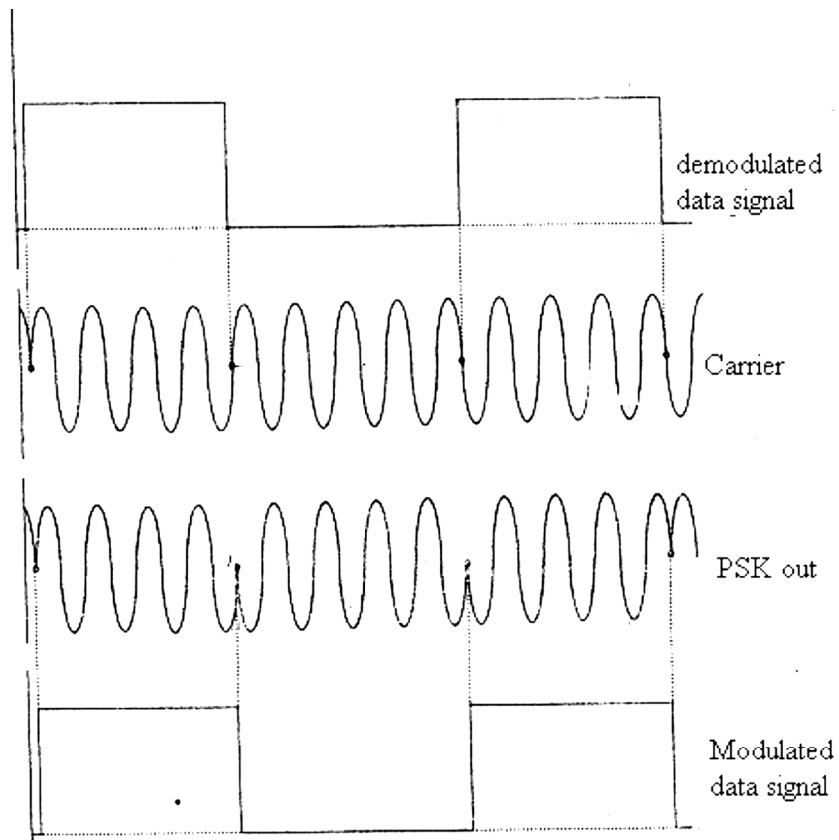
5. Give the IP-SIG 1 to IP-SIG for modulation. Give both carrier-0 and carrier-180 and monitor the modulated signal at PSK-TX of the PSK MODULATION block.
6. Give the modulated signal (PSK-TX) to PSK-RX and Carrier-0 and carrier-0 for demodulation and monitor the demodulated signal at OP SIG, compare the IP-SIG with demodulated signal.
7. Repeat the same for other input signals.

KIT DIAGRAM:



TABULATION:

S.No	Name of the signal	Amplitude (V)	Time period (msec)	Frequency
1	Message			
2	Carrier			
3	Modulated			
4	Demodulated			

MODEL GRAPH:**RESULT:**

Thus the Phase Shift Keying signal was generated using PSK modulator and the message signal was detected using PSK detector.

Exp.No.:9	TIME DIVISION MULTIPLEXING AND DEMULTIPLEXING(TDM)
Date:	

AIM:

To construct a multiplexed signal and demultiplexed signal using TDM technique and study its performance.

APPARATUS REQUIRED:

1. TDM Communication trainer
2. Multi Output Power Supply.
3. Patch cords.
4. CRO (60MHz)

THEORY:

Multiplexing is the process in which multiple Data Streams, coming from different Sources, are combined and Transmitted over a Single Data Channel or Data Stream. In Electronic Communications, the two basic forms of Multiplexing are Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM). In Time Division Multiplexing, Transmission Time on a Single Channel is divided into non-overlapped Time Slots. Data Streams from different Sources are divided into Units with same size and interleaved successively into the Time Slots. In Frequency Division Multiplexing, Data Streams are carried simultaneously on the same Transmission medium by allocating to each of them a different Frequency Band within the Bandwidth of the Single Channel. Multiplexing is done by an equipment called Multiplexer (MUX). It is placed at the Transmitting End of the communication link. At the Receiving End, the Composite Signal is separated by an equipment called Demultiplexer (DEMUX). Demultiplexer performs the reverse process of Multiplexing and routes the separated signals to their corresponding Receivers or Destinations. TDM is classified into two types: Asynchronous Time Division Multiplexing and synchronous Time Division Multiplexing.

Asynchronous Time Division Multiplexing:

Single data streams are classified in variable time segments and subsequently transmitted using the asynchronous time division multiplexing procedure. Thereby transmission occurs in no definite order. Each time segment receives a channel information number to separate them again in the demultiplexer procedure.

Synchronous Time Division Multiplexing:

Using the synchronous method, single data streams are classified in defined time segments for subsequent transmission in predetermined order.

Applications:

GSM Telephone system, TV Cable lines.

PROCEDURE:

1. Connections are made as per the kit diagram.
2. Connect the power supply cable at the POWER IN connector.

3. Switch ON the power.
4. Here POT-P1 is for frequency adjust (Sine and square1- signals), POT-P2 is for frequency adjust (square2 and triangle- signals).
5. ADD0, ADD1 and ADD2 test points are provided to monitor the addresses.
6. Connect the square signal to the channel no.1
7. With respect to ADD0 monitor the output at the MUX output buffer.
8. See that the channel is selected for every fourth zero state of ADD0
9. With respect to ADD1 monitor the output at the MUX output buffer.
10. See that the channel is selected for every second zero state of ADD1
11. With respect to ADD2 monitor the output at the MUX output buffer.
12. See that the channel is selected for every zero state of ADD2.

TABULAR COLUMN:

Name of the Signal	Amplitude(V)	Time period(msec)	Frequency
Multiplexing			
Demultiplexing			
Sine wave			
Square wave			
Triangle wave			

MODEL GRAPH:

RESULT:

Thus multiplexed signal and demultiplexed signal was constructed using TDM technique and its performance was studied.

Exp.No.:10(a)	SIMULATION OF FSK USING MATLAB
Date:	

AIM:

To perform the simulation of FSK modulation and demodulation using MATLAB.

EQUIPMENTS REQUIRED:

1. Operating system:Windows XP
2. Software:MATLAB

THEORY:

Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier signal. The technology is used for communication systems such as amateur radio, caller ID and emergency broadcasts. The simplest FSK is binary FSK (BFSK). BFSK uses a pair of discrete frequencies to transmit binary (0s and 1s) information. With this scheme, the "1" is called the mark frequency and the "0" is called the space frequency. This is simulated in MATLAB.

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

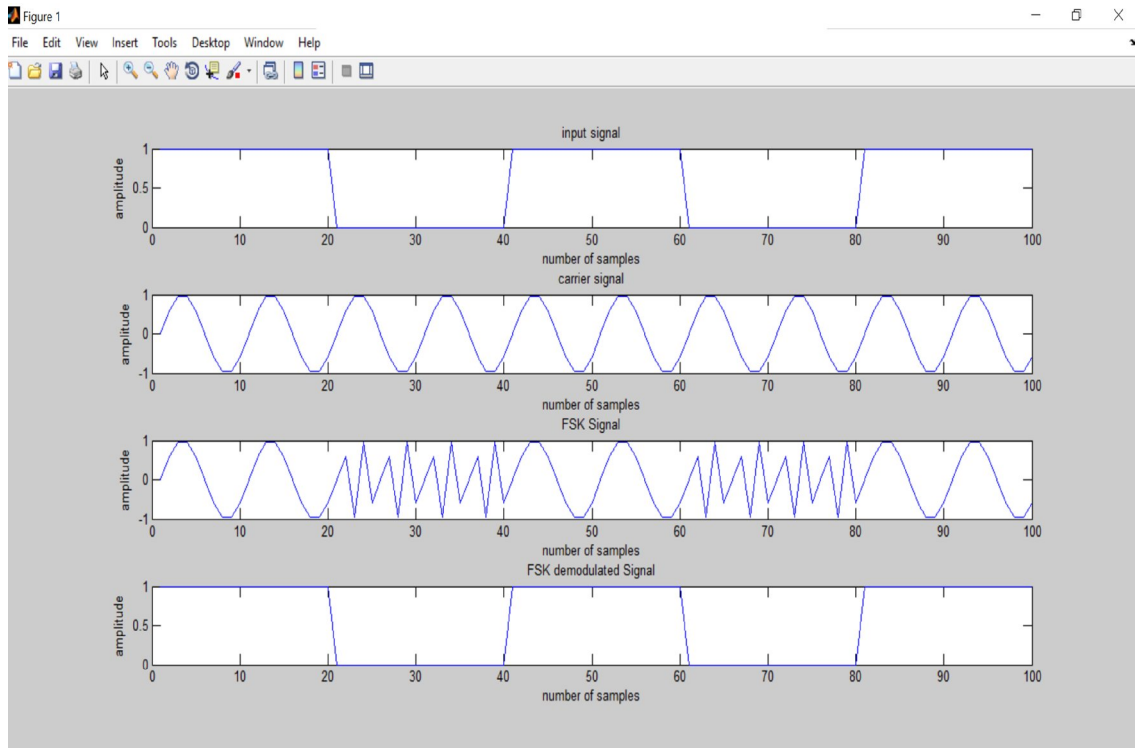
Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems. MATLAB has a number of competitors. Commercial competitors include Mathematica, TK Solver, Maple, and IDL. There are also free open source alternatives to MATLAB, in particular GNU Octave, Scilab, FreeMat, Julia, and Sage which are intended to be mostly compatible with the MATLAB language.

ALGORITHM:

1. Define the number of input samples.
2. Generate the binary input sequence.
3. Generate the carrier signal with frequency f1 and f2.
4. Generate FSK signal.
5. Reconstruct the original message signal
6. Plot the message, carrier, modulated and demodulated signal.

PROGRAM:

```
clc;
clearall;
closeall;
n=100;
x=[ones(1,20) zeros(1,20) ones(1,20) zeros(1,20) ones(1,20)];
subplot(4,1,1);
plot(x);
title('input signal');
xlabel('number of samples');
ylabel('amplitude');
f=1*10^6;
fs=10*10^6;
for i=0:n-1
d(i+1)=sin(2*pi*(f/fs)*i);
end
subplot(4,1,2);
plot(d);
title('carrier signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==1)
x(i+1)=sin((2*pi*(f/fs)*i));
else
x(i+1)=sin((2*pi*((4*(f/fs))*i)));
end
end
subplot(4,1,3);
plot(x);
title('FSK Signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==sin((2*pi*(f/fs)*i)))
x(i+1)=1;
else
x(i+1)=0;
end
end
subplot(4,1,4);
plot(x);
title('FSK demodulated Signal');
xlabel('number of samples');
ylabel('amplitude');
```

OUTPUT:**RESULT:**

Thus the simulation of FSK was performed in MATLAB and corresponding waveforms were plotted successfully.

Exp.No.:10(b)	SIMULATION OF PSK USING MATLAB
Date:	

AIM:

To perform the simulation of PSK modulation and demodulation using MATLAB.

EQUIPMENTS REQUIRED:

1. Operating system:Windows XP
2. Software:MATLAB

THEORY:

The simplest PSK technique is called binary phase-shift keying (BPSK). It uses two opposite signal phases (0 and 180 degrees). The digital signal is broken up timewise into individual bits (binary digits). The state of each bit is determined according to the state of the preceding bit. If the phase of the wave does not change, then the signal state stays the same (0 or 1). If the phase of the wave changes by 180 degrees -- that is, if the phase reverses -- then the signal state changes (from 0 to 1, or from 1 to 0). Because there are two possible wave phases, BPSK is sometimes called biphase modulation.

MATLAB is a programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job. Alternatives to MATLAB exist including open source software packages.

MATLAB is interesting in that it is dynamically compiled. In other words, when you're using it, you won't run all your code through a compiler, generate an executable, and then run the executable file to obtain a result. Instead, MATLAB simply goes line by line and performs the calculations without the need for an executable.

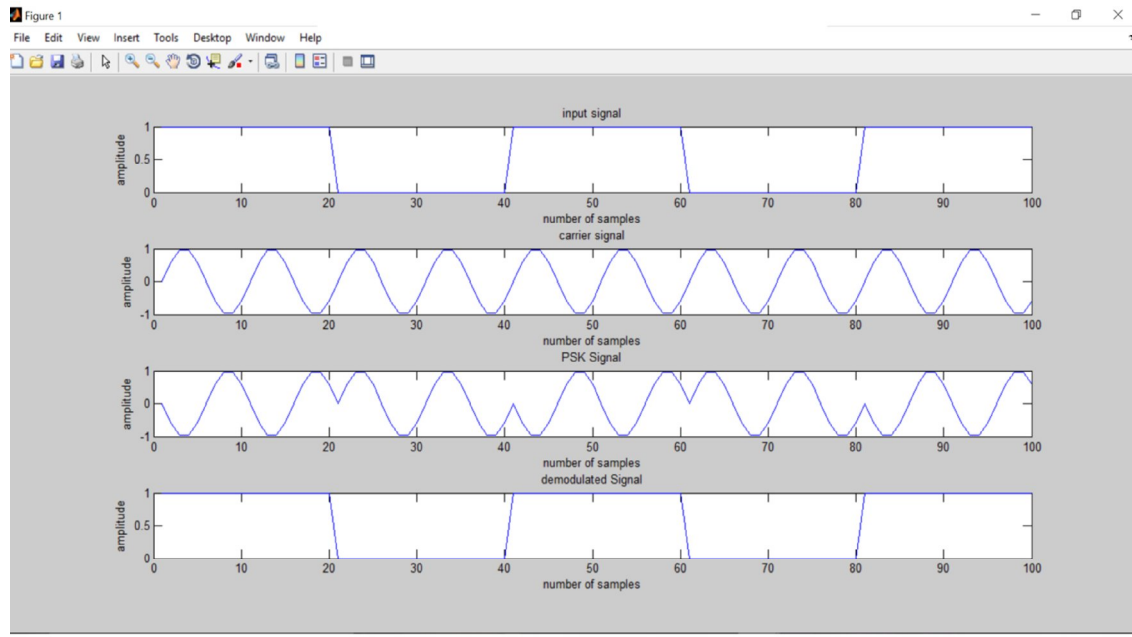
Partly because of this, it is possible to do calculations one line at a time at the command line using the same syntax as would be used in a file. It's even possible to write loops and branches at the command line if you want to. Of course this would often lead to a lot of wasted efforts, so doing anything beyond very simple calculations, testing to see if a certain function, syntax, etc. works, or calling a function you put into an .m file should be done within an .m file.

ALGORITHM:

1. Define the number of input samples.
2. Generate the binary input sequence.
3. Generate the carrier signal with frequency f1
4. Generate PSK signal.
5. Reconstruct the original message signal
6. Plot the message, carrier, modulated and demodulated signal.

PROGRAM:

```
clc;
clearall;
closeall;
n=100;
x=[ones(1,20) zeros(1,20) ones(1,20) zeros(1,20) ones(1,20)];
subplot(4,1,1);
plot(x);
title('input signal');
xlabel('number of samples');
ylabel('amplitude');
f=1*10^6;
fs=10*10^6;
for i=0:n-1
d(i+1)=sin(2*pi*(f/fs)*i);
end
subplot(4,1,2);
plot(d);
title('carrier signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==0)
x(i+1)=sin(2*pi*(f/fs)*i);
else
x(i+1)=sin(2*pi*(f/fs)*i+pi);
end
end
subplot(4,1,3);
plot(x);
title('PSK Signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==sin(2*pi*(f/fs)*i))
x(i+1)=0;
else
x(i+1)=1;
end
end
subplot(4,1,4);
plot(x);
title('PSK Signal');
xlabel('number of samples');
ylabel('amplitude');
```

OUTPUT:**RESULT:**

Thus the simulation of PSK was performed in MATLAB and corresponding waveforms were plotted successfully.

Exp.No.:10(c)	SIMULATION OF DPSK USING MATLAB
Date:	

AIM:

To perform the simulation of DPSK modulation and demodulation using MATLAB.

EQUIPMENTS REQUIRED:

1. Operating system: Windows XP
2. Software: MATLAB

THEORY:

Differential phase shift keying (DPSK), a common form of phase modulation conveys data by changing the phase of carrier wave. In **DPSK**, there is no absolute carrier phase reference, instead transmitted signal itself used as phase reference. For example, in differentially encoded BPSK a binary '1' may be transmitted by adding 180° to the current phase and a binary '0' by adding 0° to the current phase. For a signal that has been differentially encoded, there is an obvious alternative method of demodulation. Instead of demodulating as usual and ignoring carrier-phase ambiguity, the phase between two successive received symbols is compared and used to determine what the data must have been. When differential encoding is used in this manner, the scheme is known as differential phase-shift keying (DPSK).

ALGORITHM:

1. Define the number of input samples.
2. Generate the binary input sequence.
3. Generate the carrier signal with frequency f1
4. Generate DPSK signal.
5. Reconstruct the original message signal
6. Plot the message, carrier, modulated and demodulated signal.

PROGRAM:

```

clc;
clear all;
close all;
n=100;
x=[ones(1,20) zeros(1,20) ones(1,20) zeros(1,20) ones(1,20)];
subplot(4,1,1);
plot(x);
title('input signal');
xlabel('number of samples');
ylabel('amplitude');
f=1*10^6;
fs=10*10^6;
for i=0:n-1

```

```
d(i+1)=sin(2*pi*(f/fs)*i);
end
subplot(4,1,2);
plot(d);
title('carrier signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==0)
x(i+1)=sin(2*pi*(f/fs)*i);
else
x(i+1)=sin(2*pi*(f/fs)*i+pi);
end
end
subplot(4,1,3);
plot(x);
title('PSK Signal');
xlabel('number of samples');
ylabel('amplitude');
for i=0:n-1
if(x(i+1)==sin(2*pi*(f/fs)*i))
x(i+1)=0;
else
x(i+1)=1;
end
end
subplot(4,1,4);
plot(x);
title('demodulated Signal');
xlabel('number of samples');
ylabel('amplitude');
```

RESULT:

Thus the simulation of DPSK was performed in MATLAB and corresponding waveforms were plotted successfully.

Exp.No.:11	SIMULATION OF ERROR CONTROL CODING SCHEMES USING MATLAB
Date:	

AIM:

To perform the simulation of convolutional coding scheme and hamming code scheme using MATLAB.

EQUIPMENTS REQUIRED:

3. Operating system: Windows XP
4. Software: MATLAB

THEORY:**Convolutional Code:**

Convolutional codes are commonly described using two parameters: the code rate and the constraint length. The code rate, k/n , is expressed as a ratio of the number of bits into the convolutional encoder (k) to the number of channel symbols output by the convolutional encoder (n) in a given encoder cycle. The constraint length parameter, K , denotes the "length" of the convolutional encoder, i.e. how many k -bit stages are available to feed the combinatorial logic that produces the output symbols. Closely related to K is the parameter m , which indicates how many encoder cycles an input bit is retained and used for encoding after it first appears at the input to the convolutional encoder. The m parameter can be thought of as the memory length of the encoder. Convolutional codes are widely used as channel codes in practical communication systems for error correction. The encoded bits depend on the current k input bits and a few past input bits. The main decoding strategy for convolutional codes is based on the widely used Viterbi algorithm. As a result of the wide acceptance of convolutional codes, there have been several approaches to modify and extend this basic coding scheme. Trellis coded modulation (TCM) and turbo codes are two such examples. The operation of a convolutional encoder can be explained in several but equivalent ways such as, by a) state diagram representation, b) tree diagram representation and c) trellis diagram representation.

Hamming code:

The most common types of error-correcting codes used in RAM are based on the codes devised by R. W. Hamming. In the Hamming code, k parity bits are added to an n -bit data word, forming a new word of $n + k$ bits. The bit positions are numbered in sequence from 1 to $n + k$. Those positions numbered with powers of two are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.

An extended Hamming code having one extra parity bit is often used. Extended Hamming codes achieve a Hamming distance of four, which allows the decoder to distinguish between when at most one one-bit error occurs and when any two-bit errors occur. In this sense,

extended Hamming codes are single-error correcting and double-error detecting, abbreviated as **SECDED**.

ALGORITHM:

Convolutional Code:

1. Define the length of the message
2. Define the input binary message
3. Define the codeword length
4. Convert the convolutional polynomial to trellis structure.
5. Finally encode the binary data convolutionally.
6. Obtain the codeword for a given message.

Hamming code:

1. Set the code parameters.
2. Create a binary message having length k.
3. Encode the message.
4. Introduce an error in the 4th bit of the encoded sequence.
5. Decode the corrupted sequence. Observe that the decoder has correctly recovered the message.

PROGRAM:

Convolutional Codes

```
clc;
clearall;
closeall;
k=input('Enter the message length');
n=input('Enter the codeword length');
x=input('Enter the input');
t=poly2trellis(3,[n,k]);
code=convenc(x,t);
```

Hamming code:

```
clc;
clearall;
closeall;
n = 15;           % Code length
k = 11;
data = randi([0 1], k, 1);
encData = encode(data, n, k, 'hamming/binary');
encData(4) = ~encData(4);
decData = decode(encData, n, k, 'hamming/binary');
numerr = biterr(data, decData)
```

OUTPUT:

Convolutional Codes:

```
Enter the input[1 0 0 1 1]
Enter the codeword length7
Enter the message length5
>>code = 1 1 1 0 1 1 1 1 0 1
```

Hamming code:

```
numerr =
```

```
0
```

```
>>data
```

```
data =
```

```
0
```

```
1
```

```
1
```

```
0
```

```
1
```

```
1
```

```
1
```

```
1
```

```
0
```

```
1
```

```
0
```

```
encData =
```

```
1
```

```
0
```

```
1
```

```
0
```

```
0
```

```
1
```

```
1
```

```
0
```

```
1
```

```
1
```

```
1
```

```
1
```

```
0
```

```
1
```

```
0
```

```
>>decData
```

```
decData =
```

```
0
```

```
1
```

```
1
```

```
0
```

```
1
```

1
1
1
0
1
0

RESULT:

Thus the simulation of convolutional coding scheme and hamming code scheme was performed in MATLAB successfully.

Exp.No.:12	SIMULATION OF SPREAD SPECTRUM COMMUNICATION USING MATLAB
Date:	

AIM:

To obtain a Direct Sequence Spread Spectrum (DSSS) for a given input signal by simulating in MATLAB.

EQUIPMENTS REQUIRED:

5. Operating system: Windows XP
6. Software: MATLAB

THEORY:

In telecommunication and radio communication, spread-spectrum techniques are methods by which a signal (e.g. an electrical, electromagnetic, or acoustic signal) generated with a particular bandwidth is deliberately spread in the frequency domain, resulting in a signal with a wider bandwidth. These techniques are used for a variety of reasons, including the establishment of secure communications, increasing resistance to natural interference, noise and jamming, to prevent detection, and to limit power flux density (e.g. in satellite downlinks).

Frequency-hopping spread spectrum (FHSS), direct-sequence spread spectrum (DSSS), time-hopping spread spectrum (THSS), chirp spread spectrum (CSS), and combinations of these techniques are forms of spread spectrum. Each of these techniques employs pseudorandom number sequences — created using pseudorandom number generators — to determine and control the spreading pattern of the signal across the allocated bandwidth. In DSSS the message signal is used to modulate a bit sequence known as the Pseudo Noise (PN) code; this PN code consists of pulses of a much shorter duration (larger bandwidth) than the pulse duration of the message signal, therefore the modulation by the message signal has the effect of chopping up the pulses of the message signal and thereby resulting in a signal which has a bandwidth nearly as large as that of the PN sequence. In this context the duration of the pulse of the PN code is referred to as the chip duration and the smaller this value, the larger the bandwidth of the resultant DSSS signal and the more immune to interference the resultant signal becomes.

Some of the uses of DSSS include the Code Division Multiple Access (CDMA) channel access method and the IEEE 802.11b specification used in Wi-Fi networks.

ALGORITHM:

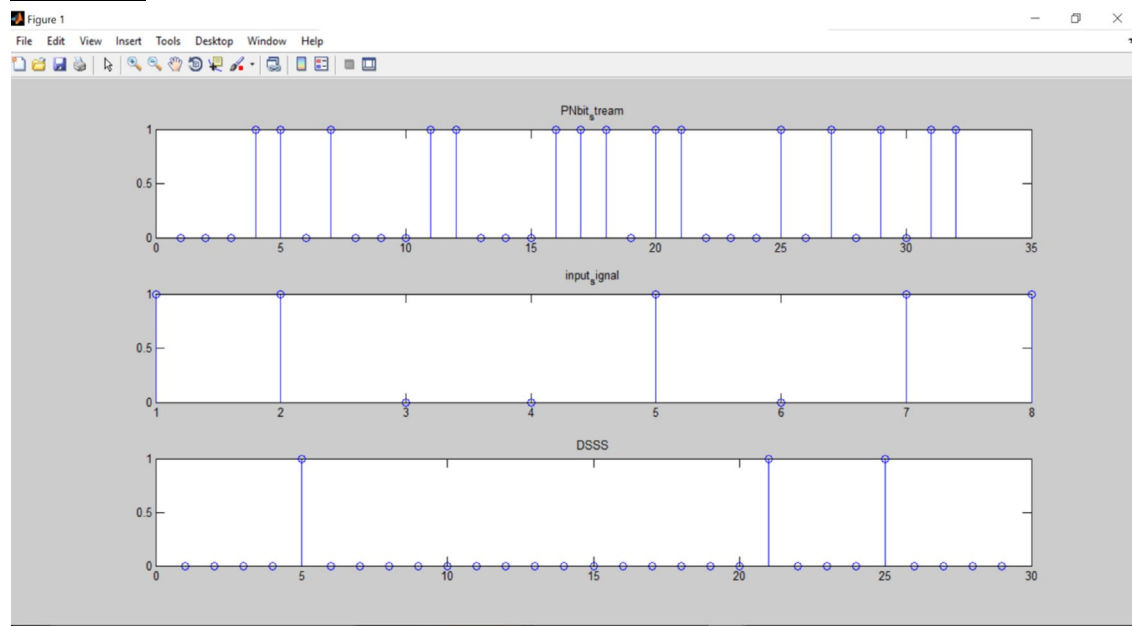
1. Define the PN bit stream
2. Define the input bit stream
3. Obtain the DSSS stream.
4. Plot the PN bit stream, input bit stream and DSSS stream.

PROGRAM:

```

clc;
clearall;
closeall;
PNbit_stream = round(rand(1,32));
input_signal=[1 1 0 0 1 0 1 1];
PNbit_stream;
for i=1:1:8
for j=1:4:32
for k=1:1:4
a(j)=xor(PNbit_stream(k+j-1),input_signal(i));
a(j);
end
end
end
subplot(3,1,1)
stem(PNbit_stream);
title('PNbit_stream');
subplot(3,1,2)
stem(input_signal);
title('input_signal');
subplot(3,1,3);
stem(a);
title('DSSS');

```

OUTPUT:**RESULT:**

Thus the simulation of Direct Sequence Spread Spectrum was performed in MATLAB and corresponding waveforms were plotted successfully.

Exp.No.:13	SIMULATION OF FREQUENCY DIVISION MULTIPLEXING (FDM) USING MATLAB
Date:	

AIM:

To perform the simulation of Frequency Division Multiplexing (FDM) using MATLAB.

EQUIPMENTS REQUIRED:

1. Operating system: Windows XP
2. Software: MATLAB

THEORY:

In telecommunications, frequency-division multiplexing (FDM) is a technique by which the total bandwidth available in a communication medium is divided into a series of non-overlapping frequency sub-bands, each of which is used to carry a separate signal. This allows a single transmission medium such as the radio spectrum, a cable or optical fiber to be shared by multiple independent signals. Another use is to carry separate serial bits or segments of a higher rate signal in parallel.

The most natural example of frequency-division multiplexing is radio and television broadcasting, in which multiple radio signals at different frequencies pass through the air at the same time. Another example is cable television, in which many television channels are carried simultaneously on a single cable. FDM is also used by telephone systems to transmit multiple telephone calls through high capacity trunk lines, communications satellites to transmit multiple channels of data on uplink and downlink radio beams, and broadband DSL modems to transmit large amounts of computer data through twisted pair telephone lines, among many other uses.

ALGORITHM:

1. Define the number of samples and number of users.
2. Define the message signal frequency and carrier signal frequency for each user.
3. Define frequency deviation
4. Generate modulating and modulated signal of each user
5. Pass the modulated signal through the channel
6. Demodulate the signal
7. Display transmitted and received signal.
8. Display the combination of all signal passed through the channel.

PROGRAM:

```
% Design and simulation of Frequency Division Multiplexing in MATLAB
clc;
clearall
closeall
samples=100;
```

```

nos=3;% number of users
mfreq=[30 40 50]; % modulating signal frequency in Hz
cfreq=[300 600 900];% carrier frequency allocated to the different users in
Hz
freqdev=100; % choose frequency deviation

% generate modulating signal
t=linspace(0,1000,samples);
parfor i=1:nos
m(i,:)=sin(2*pi*mfreq(1,i)*t);
end

% Generate the modulated signal
parfor i=1:nos
    y(i,:)=fmmod(m(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);
end

% pass the modulated signal through the channel

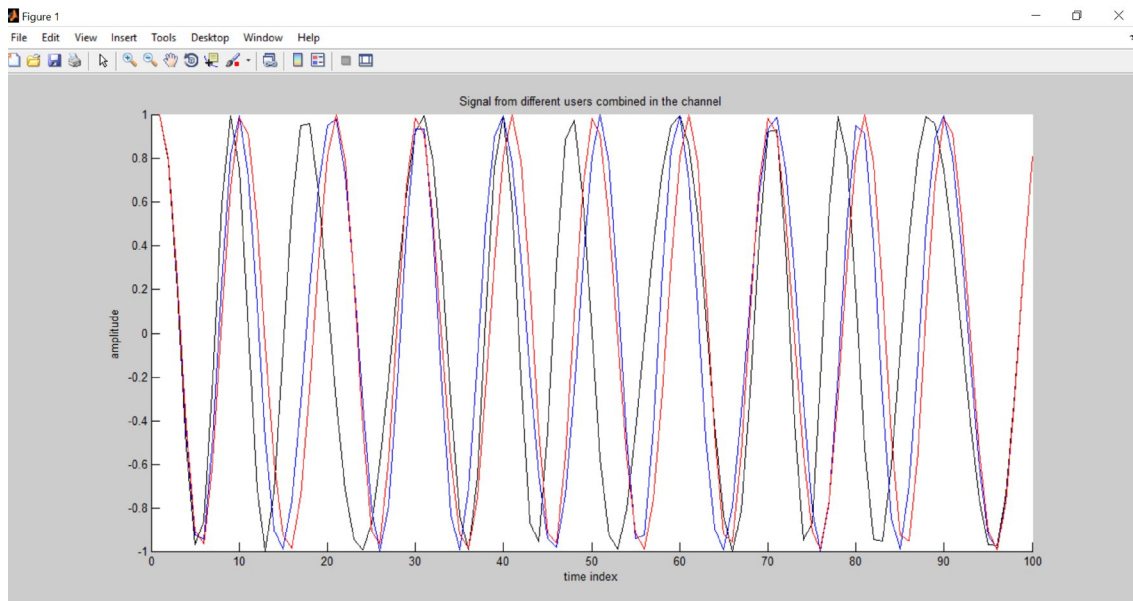
%ch_op=awgn(sum(y),0,'measured');

% demodulate the received signal at the base station
parfor i=1:nos
    z(i,:)=fmdemod(y(i,:),cfreq(1,i),10*cfreq(1,i),freqdev);
end

% display the transmitted signal and received signal at the base station
% figure

C = {'k','b','r','g','y',[.5 .6 .7],[.8 .2 .6],[.3 .2 .2]}; % Cell array of
colros.
for i=1:nos
figure (1)
holdon
plot(y(i,:), 'color',C{i});
xlabel('time index'); ylabel('amplitude'); title('Signal from different
users combined in the channel');
figure
subplot(3,1,1)
plot(m(i,:)) % modulating signal
xlabel('time index'); ylabel('amplitude'); title('modulating Signal from
user');
subplot(3,1,2)
plot(y(i,:), 'color',C{i}); % modulated signal
xlabel('time index'); ylabel('amplitude'); title('modulated Signal from
user');
subplot(3,1,3)
plot(z(i,:), 'color',C{i}); % demodulated signal
xlabel('time index'); ylabel('amplitude'); title('demodulated Signal from
user at the base station');
end
%figure
%plot(ch_op) % combination of all modulated signals passed through the
channel
%xlabel('time index'); ylabel('amplitude'); title('Signal after passing
through the channel');

```

OUTPUT:**RESULT:**

Thus simulation of Frequency Division Multiplexing has been done successfully using MATLAB.

Exp.No.:14	SIMULATION OF COMMUNICATION LINK USING SCILAB
Date:	

AIM:

To perform the simulation of communication link using SCILAB.

EQUIPMENTS REQUIRED:

1. Operating system: Windows XP
2. Software: SCILAB

THEORY:

SCILAB is open-source software that specializes in scientific computation. Xcos is a graphical editor to design hybrid dynamical systems models. Models can be designed, loaded, saved, compiled and simulated. SCILAB runs, and is available in binary format, for the main available platforms: Unix/Linux workstations (the main software development is performed on Linux workstations), Windows, and MacOSX. MacOSX users can also install SCILAB using fink.

Key Features of Xcos:

- **Graphically model, compile, and simulate dynamical systems**

It combines continuous and discrete-time behaviors in the same model and select model elements from Palettes of standard blocks. Programming new blocks is done in C, Fortran, or Scilab Language.

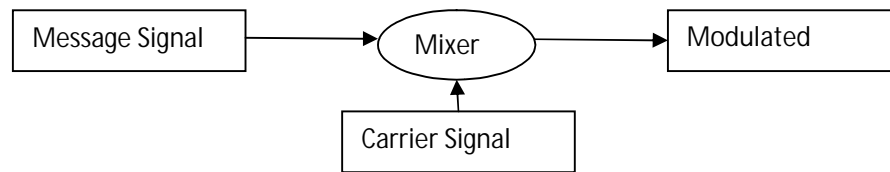
- **Use standards**

HDF5 standard which has been chosen to guarantee data exchanges between Scilab and Xcos Editor. A free Modelica compiler which enables the simulation of implicit diagrams. A brand new graphical user interface based on JGraphX.

Many digital modulation techniques are DSB-SC with a digital message signal. Therefore, it is appropriate to review this basic analog modulation. DSB-SC is the simplest form of amplitude modulation, generated by a mixer. This consists of an audio source combined with the frequency carrier, as shown in Figure. The message signal $m(t)$ is multiplied with a high frequency carrier,

$$c(t) = A \cos(2\pi f_c t)$$

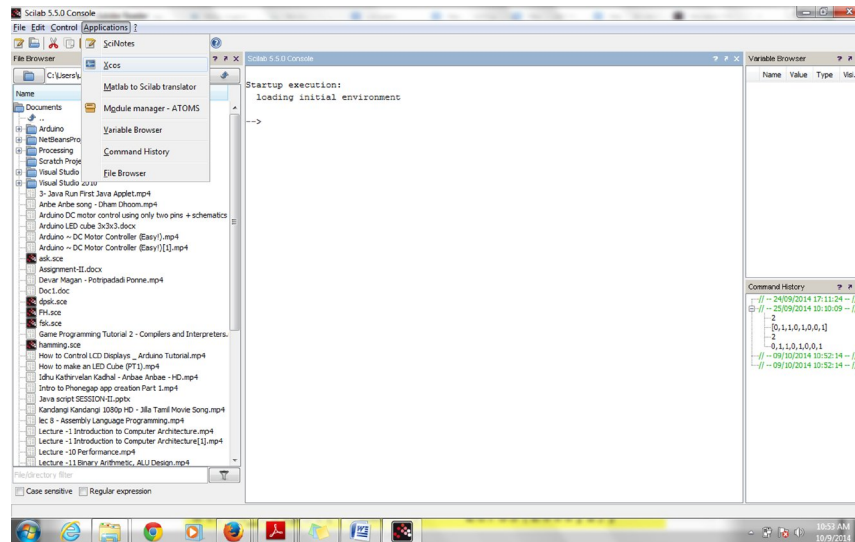
where A is the carrier amplitude and f_c is the carrier frequency, to provide a DSB-AM signal modulated output $s(t)$. Note that the amplitude of $s(t)$ is $A m(t)$, which varies according to the message signal $m(t)$, hence the name “amplitude modulation”. In other words, the message signal is now being carried in the amplitude of the DSB-AM signal. The generation of $s(t)$ is achieved by the modulator. The voltage multiplier shown in Figure, is commonly referred to as the mixer.



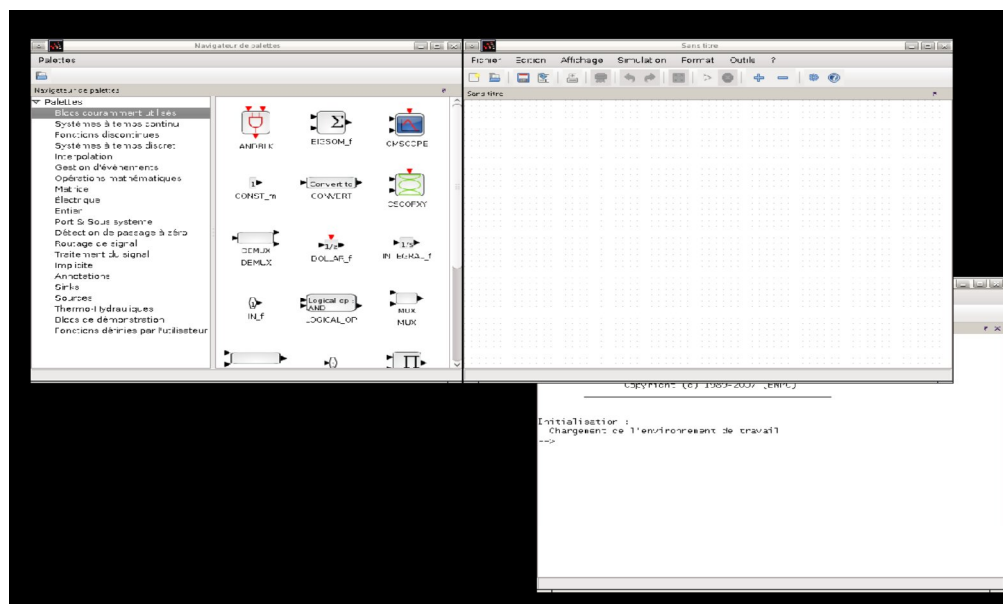
AM modulator

PROCEDURE:

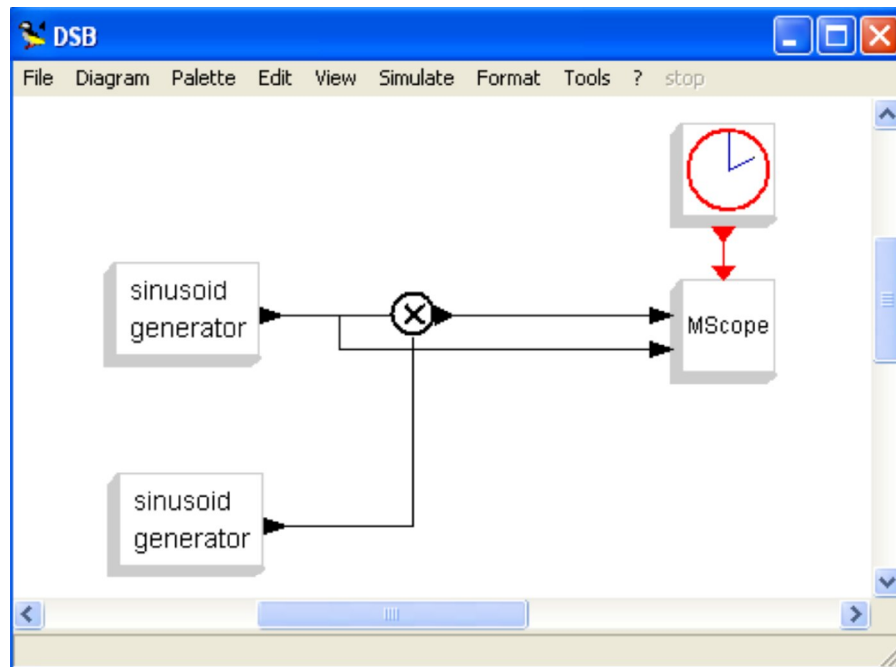
1. Install Scilab and goto Applications->Xcos->click



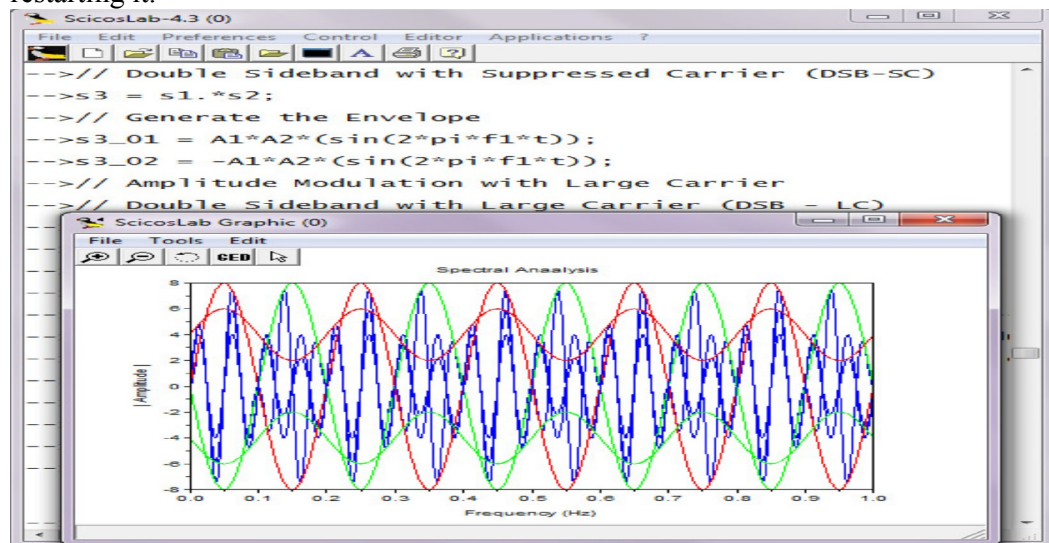
2. Then the palette appears and we can design any of the link.



3. To simulate AM Modulator, it is sufficient to select the Run operation from the Simulate menu.



4. Simulation parameters can be set by the Setup operation in the same menu.
5. Running the simulation for the system leads to the opening of a graphics window and the display of a sinusoidal signal.
6. This window is opened and updated by the FFT block.
7. The simulation result is given below. In this case, the Sinusoid Generator Frequency (rad/s) is 103, 106 and 109 respectively.
8. The simulation can be stopped using the stop button on the main Xcos window, subsequent to which the user has the option of continuing the simulation, ending the simulation, or restarting it.



RESULT:

Thus Communication link has been simulated successfully using SCILAB.

CONTENT BEYOND SYLLABUS

Exp.No.:15	SIMULATION OF LINEAR BLOCK CODES USING MATLAB
Date:	

AIM:

To perform the simulation of Linear Block coding scheme using MATLAB.

EQUIPMENTS REQUIRED:

1. Operating system: Windows XP
2. Software: MATLAB

THEORY:

Linear block coding involves encoding of block of source information bits into another block of bits with addition of error control bits to combat channel errors induced during transmission. An (n,k) linear block code encodes k information data bits into n bit codeword. The information data sequence is divided into sequential information data blocks. Each information data block is k bits long. The error control bits $(n-k)$ are derived from a block of k information data bits where n is total number of bits in encoded data block ($n > k$). The encoder adds $(n-k)$ error control bits to each k bit information data block. A block of n encoded bits at any instance depends only on the block of data consisting of k information bits present at that time. So there is no inbuilt memory. When the k information data bits appear at the beginning of the codeword, the code is called as systematic code. The n bit data block is called as code word.

ALGORITHM:

1. Define the code length and message length
2. Create a binary message having length k .
3. Create a cyclic generator polynomial. Then, create a parity-check matrix and convert it into a generator matrix.
4. Encode the message sequence by using the generator matrix.
5. Introduce an error in the 3rd bit of the encoded sequence.
6. Decode the corrupted sequence. Observe that the decoder has correctly recovered the message.

PROGRAM:

```

clc;
clearall;
closeall;
n = 7;           % Code length
k = 3;
data = randi([0 1],k,1);
pol = cyclpoly(n,k);

```

```
parmat = cyclgen(n,pol);
genmat = gen2par(parmat);
encData = encode(data,n,k,'linear/binary',genmat);
encData(3) = ~encData(3);
decData = decode(encData,n,k,'linear/binary',genmat);
numerr = biterr(data,decData);
```

OUTPUT:

Single-error patterns loaded in decoding table. 8 rows remaining.

2-error patterns loaded. 1 rows remaining.

3-error patterns loaded. 0 rows remaining.

```
>>data
```

```
data =
```

```
1
```

```
1
```

```
0
```

```
>>genmat
```

```
genmat =
```

```
1 0 1 1 1 0 0
```

```
1 1 1 0 0 1 0
```

```
0 1 1 1 0 0 1
```

```
>>parmat
```

```
parmat =
```

```
1 0 0 0 1 1 0
```

```
0 1 0 0 0 1 1
```

```
0 0 1 0 1 1 1
```

```
0 0 0 1 1 0 1
```

```
>>encData
```

```
encData =
```

```
0
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
0
```

```
>>decData
```

```
decData =
```

```
1
```

```
1
```

```
0
```

```
>>numerr
```

```
numerr =
```

```
0
```

RESULT:

Thus the simulation of Linear Block Coding scheme was performed in MATLAB successfully.