Sri Shanmugha College of Engineering and Technology
Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai
Accredited by NAAC and NBA (ECE/CSE/MECH)
Pullipalayam, Sankari, Salem (Dt.)

Department of Electronics and Communication Engineering

LAB MANUAL

EC8561 COMMUNICATION SYSTEMS LABORATORY

SEMESTER V /YEAR-III

(R-2017)

PREPARED BY

Mrs.M.SARANYA, AP/ECE

EC8561	**COMMUNICATION SYSTEMS LABORATORY**	**L**	**T**	**P**	**C**
		0	**0**	**4**	**2**

OBJECTIVES:

The student should be made:

- ❖ To visualize the effects of sampling and TDM
- ❖ To Implement AM & FM modulation and demodulation
- ❖ To implement PCM & DM
- ❖ To simulate Digital Modulation schemes
- ❖ To simulate Error control coding schemes

LIST OF EXPERIMENTS:

1. Signal Sampling and reconstruction
2. Time Division Multiplexing
3. AM Modulator and Demodulator
4. FM Modulator and Demodulator
5. Pulse Code Modulation and Demodulation
6. Delta Modulation and Demodulation
7. Line coding schemes
8. Simulation of ASK, FSK, and BPSK generation schemes
9. Simulation of DPSK, QPSK and QAM generation schemes
10. Simulation of signal constellations of BPSK, QPSK and QAM
11. Simulation of ASK, FSK and BPSK detection schemes
12. Simulation of Linear Block and Cyclic error control coding schemes
13. Simulation of Convolutional coding scheme
14. Communication link simulation

TOTAL: 60 PERIODS

OUTCOMES:

At the end of the course, the student should be able to:

- ❖ Simulate & validate the various functional modules of a communication system
- ❖ Demonstrate their knowledge in base band signaling schemes through implementation of digital modulation schemes
- ❖ Apply various channel coding schemes & demonstrate their capabilities towards the improvement of the noise performance of communication system

- Simulate end-to-end communication Link

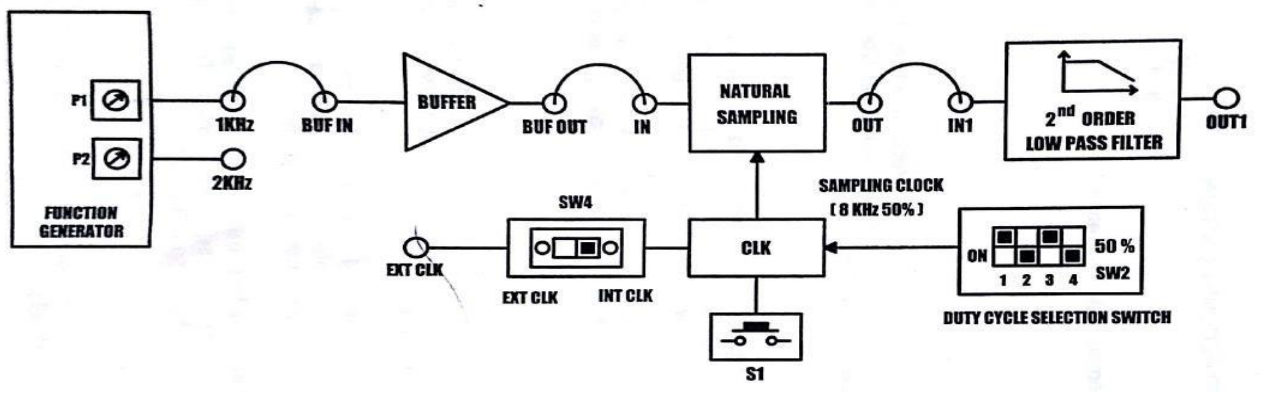
COURSE CODE	**EC8561**	**COURSE NAME**	**COMMUNICATION SYSTEMS LABORATORY**	**SEM**	**5**
On completion of the course, the students will be able to					
CO1	Demonstrate analog modulation and demodulation scheme.				
CO2	Demonstrate the digital modulation & Demodulation scheme.				
CO3	Demonstrate the concept of line coding techniques.				
CO4	Simulation of BPSK, QPSK & QAM signal constellation, ASK, FSK BPSK & DPSK digital modulation schemes using MATLAB.				
CO5	Simulation using communication link (MATLAB Simulink) & also the error control coding, convolutional coding.				

CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**
CO1	3	3	3	0	1			1	2	2		1	3	3
CO2	3	3	3	0	1			1	2	2		1	3	3
CO3	3	3	3	0	1			1	2	2		1	3	3
CO4	3	3	3	1	1			1	2	2		1	3	3
CO5	3	3	3	1	1			1	2	2		1	3	3
C406	3	3	3	0	1			1	2	2		1	3	3

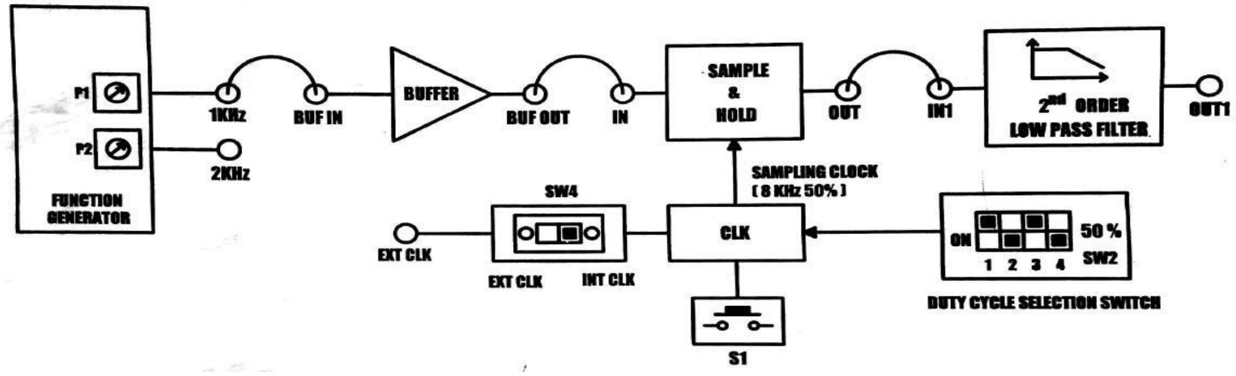
INDEX

CONNECTION DIAGRAM

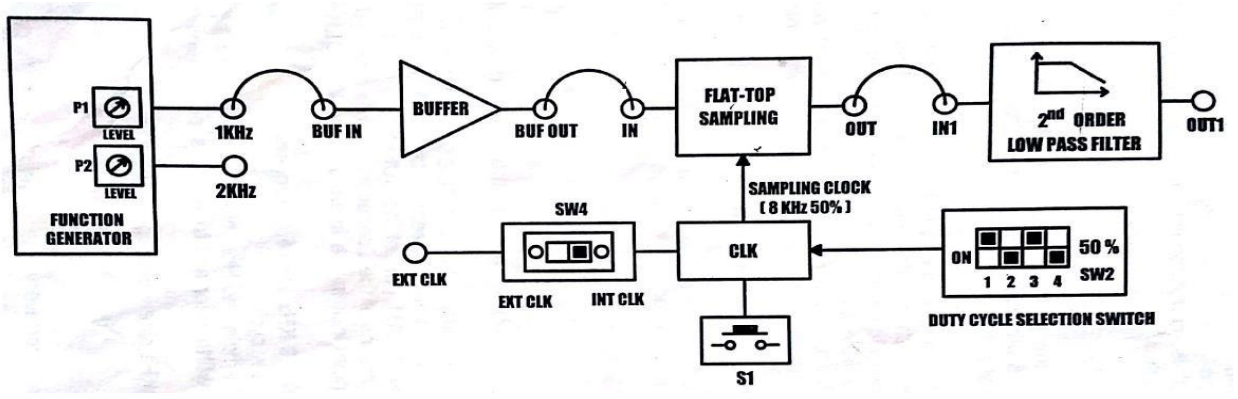
NATURAL SAMPLING

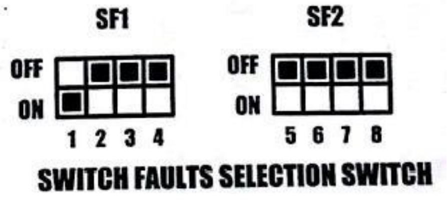


SAMPLE AND HOLD



FLAT – TOP SAMPLING





1. SIGNAL SAMPLING AND RECONSTRUCTION

AIM:

To sample and reconstruct the given signal using natural sampling, sample - hold and flat top sampling techniques using DCL-01Kit.

APPARATUS REQUIRED:

S.No.	Name of the Equipment/ Component	Range	Quantity
1.	DCL-01 trainer kit	-	1
2.	CRO	30 MHz	1
3.	Power supply	5V, ±12V	1
4.	Patch chords	-	Required

THEORY

Sampling is the process of splitting the given analog signal into different samples of equal amplitudes with respect to time. There are two types of sampling namely natural sampling, flat top sampling. Sampling should follow strictly the Nyquist Criterion i.e. the sampling frequency should be twice higher than that of the highest frequency signal.

$fs \geq 2fm$

Where,

fs= Minimum Nyquist Sampling rate (Hz), *fm*= Maximum analog input frequency (Hz).

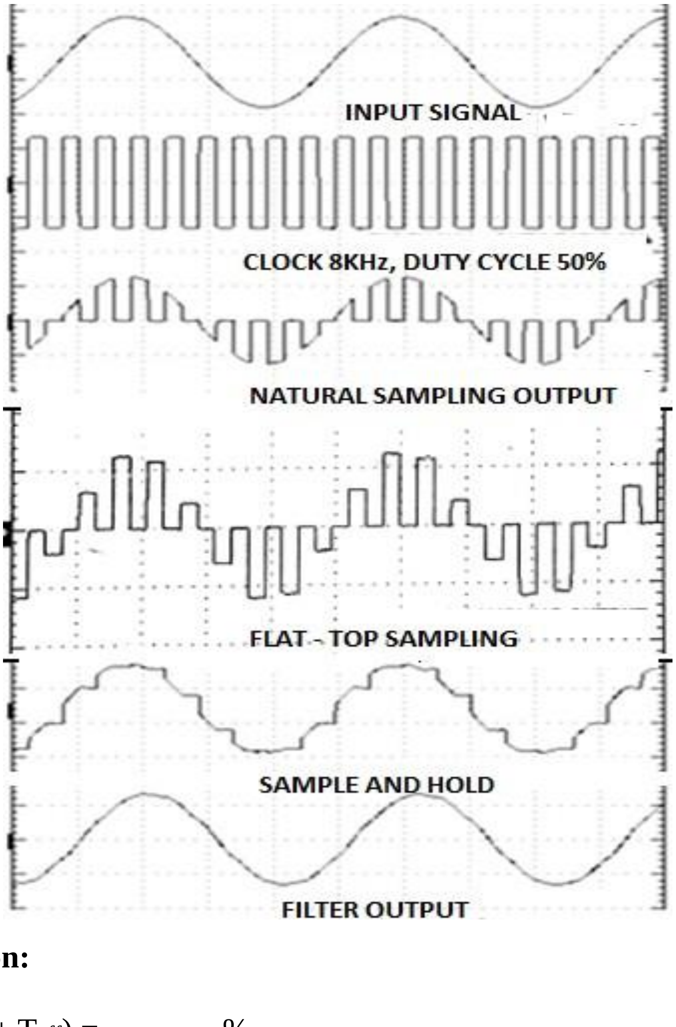
In natural sampling, the top of the sampled pulse follows the shape of the original signal. Since the natural sampling increases the system complexity, the flat top sampling is mostly preferred in practical case. The flat top sampling is achieved with the help of sample and hold circuit. Sample and Hold circuits are used internally in analog to digital conversion. Here, the sampled signal obtained at each sampling instant is hold until the next sampling instant. The sampling process must follow the sampling theorem for proper signal reconstruction. In other words, the sampling frequency must be equal to twice that of the highest frequency component present in the original signal.

The reconstructed signal is the succession of sine pulses weighted by x (nTs) these pulses are interpolated with the help of a LPF. It is also called reconstruction filter or interpolation filter Natural sampling is chopper sampling because the waveform of the sampled signal appears to be chopped off from the original signal waveform. The top of the samples remains constant and equal to instantaneous value of x(t) at start of sampling fs = 1/Ts.

TABULATION

Parameters	**Amplitude in Volts**	**Time Period in Seconds**
Input Signal		
Sampling Signal		Ton = Toff=
Natural sampling Output		Ton = Toff=
Flat-top sampling Output		Ton = Toff=
Sample and Hold output		
Reconstructed Signal		

MODEL GRAPH



Duty cycle calculation:

$D = T_{on} / (T_{on} + T_{off})$ = -----------%

PROCEDURE:

1. Give the connections as per the connection diagram.
2. Put the duty cycle selector switch in position 50%.
3. Connect the modulating signal of 1 KHz frequency to BUFIN and measure its amplitude and time period.
4. Connect the sampling frequency clock in the internal mode INT CLK using the sampling signal selector switch SW4.
5. Set the sampling frequency to 8 KHz and note down the amplitude and time period of the sampling signal.
6. Observe the natural sampling output at OUT terminal and note down the amplitude and time period of the sampled signal.
7. Give the sampled signal to the 2nd order low pass filter circuit and observe the reconstructed signal.
8. Note down the amplitude and time period of the reconstructed signal.
9. Repeat the same procedure for flat top sampling, Sample and hold and note down the readings.
10. Plot the readings in the graph.

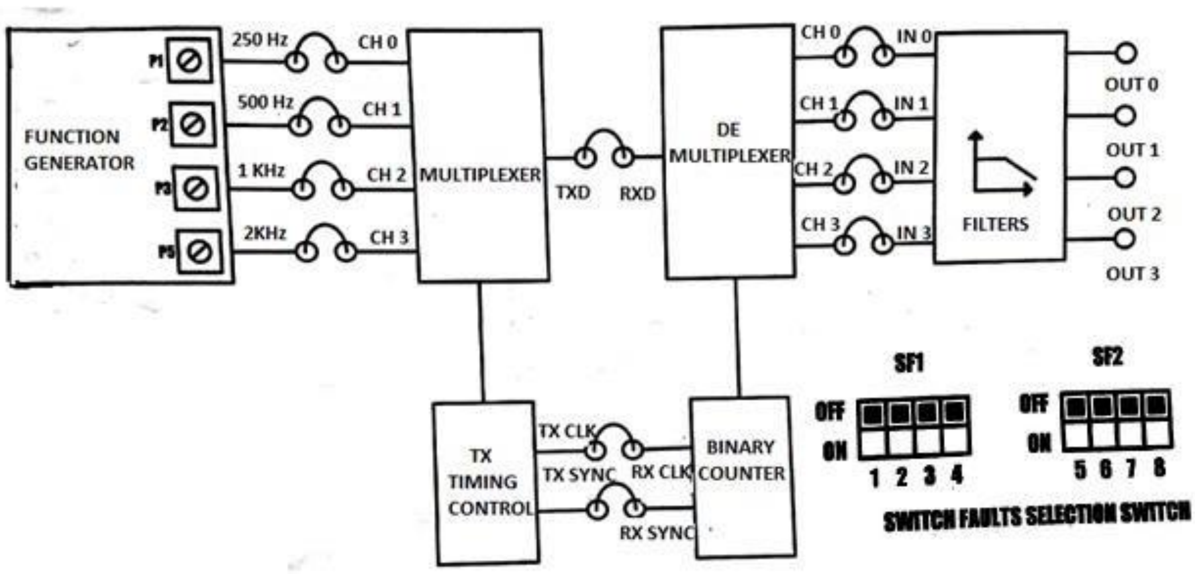
VIVA QUESTIONS:

1. What is the need of sampling?
2. What is sampling rate?
3. What is Nyquist rate of sampling?
4. Mention the types of sampling.
5. State the sampling theorem.
6. What is natural sampling?
7. Differentiate three types of sampling techniques.
8. What is the function sample & hold circuit?
9. What is aliasing effect and how it can be eliminated?
10. State the role of duty cycle in sampling circuit.
11. What is the function reconstruction filter?
12. What is the use of sampling and reconstruction technique in communication engineering?

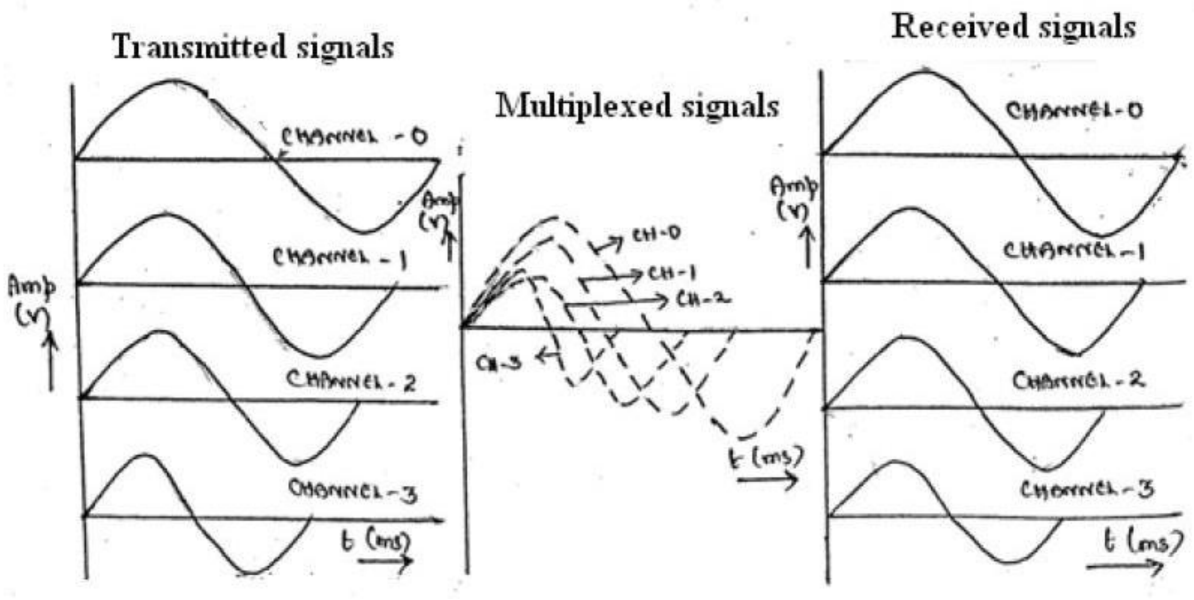
RESULT:

Thus the given signal is sampled and reconstructed for natural sampling,sample and hold and flat top sampling alsoplotted the observed waveforms.

CONNECTION DIAGRAM



MODEL GRAPH



2. TIME DIVISION MULTIPLEXING

AIM:

To perform four channel Time Division multiplexing and De multiplexing using DCL – 02 trainer kit.

APPARATUS REQUIRED:

S.No.	Name of the Equipment/ Component	Range	Quantity
1.	DCL-02 trainer kit	-	1
2.	CRO	30 MHz	1
3.	Power supply	5V, ±12V	1
4.	Patch chords	-	Required

THEORY:

Time Division Multiplexing (TDM) is a technique of transmitting different source signals on the same channel at different time slots. That is several information can be transmitted over a single channel by sending samples from different information sources at different moments. TDM is widely used in digital communication systems to increase the efficiency of the transmitting medium. TDM can be achieved by electronically switching the samples such that they interleave sequentially at correct instant in time without mutual interference.A major problem in any TDM system is the synchronization of the transmitter and receiver timing circuits. The transmitter and receiver must switch at the same time and frequency.TDM based on analog modulation, the time slots are separated by guard slots to prevent crosstalk between the channels.

In PAM, PPM the pulse is present for a short duration and for most of the timebetween the two pulses no signal is present. This free space between the pulses can be occupied by pulses from other channels. Thus, time division multiplexing makes maximum utilization of the transmission channel. Each channel to be transmitted is passed through the low pass filter. The outputs of the low pass filters are connected to the rotating sampling switch (or) commutator.

It takes the sample from each channel per revolution and rotates at the rate of f s. Thusthe sampling frequency becomes fs the single signal composed due to multiplexing of input channels. These channels signals are then passed through low pass reconstruction filters. If the highest signal frequency present in all the channels is fm, then by sampling theorem, the sampling frequency fs must be such that fs≥2fm. Therefore, the time space between successive samples from any one input will be Ts=1/fs, and Ts≤ 1/2fm.

TABULATION:

Parameters	**Amplitude in Volts**	**Time Period in Seconds**
Channel 0 (CH 0)		
Channel 1 (CH 1)		
Channel 2 (CH 2)		
Channel 3 (CH 3)		
TxD		
RxD		
Output 0		
Output 1		
Output 2		
Output 3		

PROCEDURE:

1. Give the connections as per the connection diagram.
2. Connect the power supply with proper polarity to the kit DCL-02 and switch it ON.
3. Connect 250 Hz, 500 Hz, 1 KHz, 2 KHz sine wave signal from the function generator to the multiplexer input channel CH0, CH1, CH2 and CH3 by means of connecting cords.
4. Connect the multiplexer OUT TXD to the transmitter section to the de-multiplexer input RXD of the receiver section.
5. Connect the output of the receiver section CH0, CH1, CH2 and CH3 to the IN0, IN1, IN2 and IN3 of the filter section.
6. Connect the sampling clock TXCLK and the channel identification clock SYNC of the receiver section respectively.
7. Set the amplitude of the input sine wave as desired.
8. Take the observations and draw the required graphs.

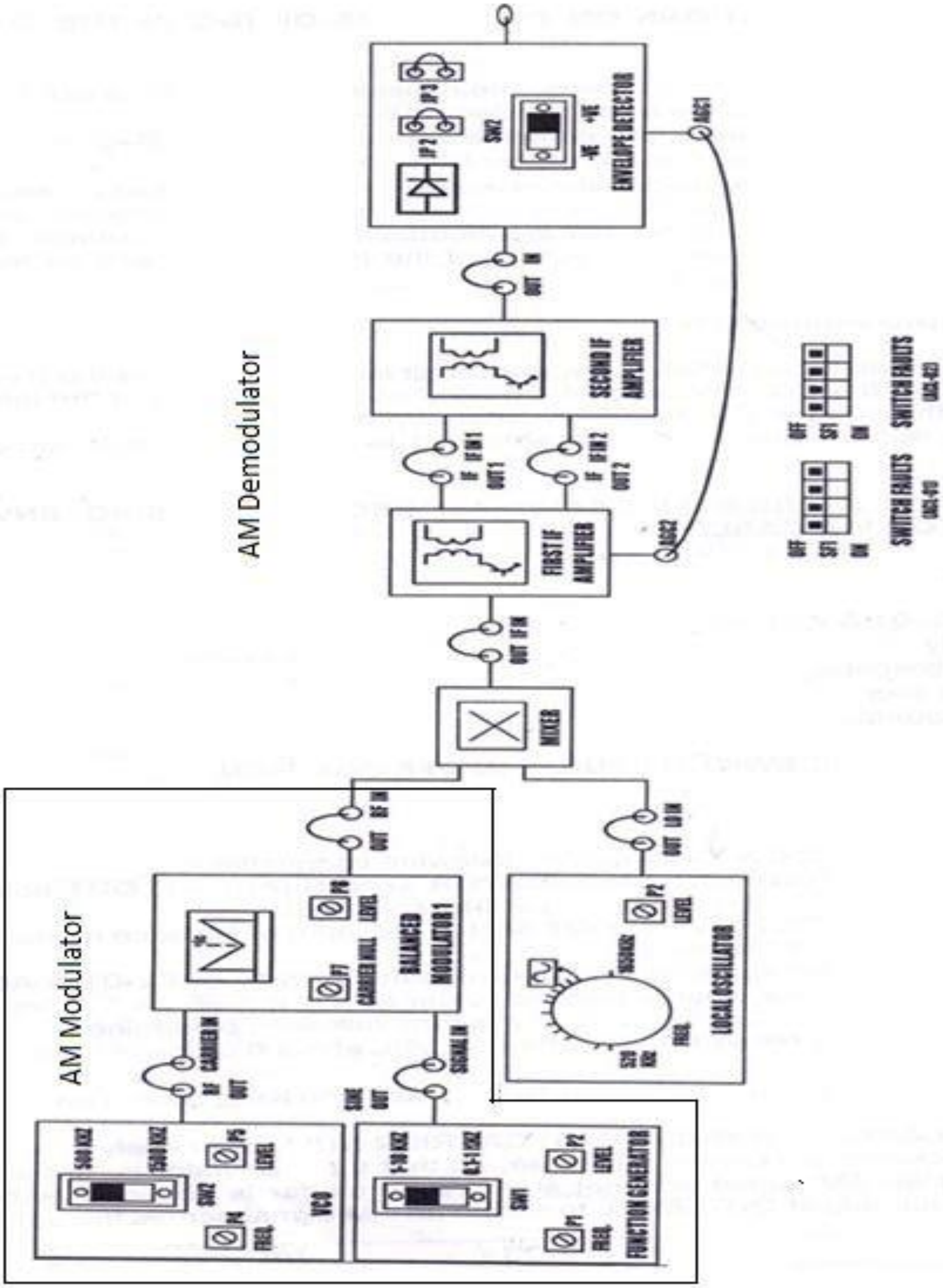
VIVA QUESTIONS:

1. What is meant by multiplexing technique?
2. What are the two types of multiplexing?
3. When do you prefer TDM to FDM?
4. When would you prefer FDM to TDM?
5. Differentiate TDM and FDM.
6. Define crosstalk effect in PAM/TDM system.
7. What is the transmission bandwidth of a PAM/TDM channel?
8. State the advantages of TDM.
9. State the disadvantages of TDM.
10. Why are the bands limiting filters used at the PAM-TDM transmitter?
11. In TDM, multiple signals share a channel by transmitting in different ……..
12. The minimum sampling rate for a voice channel is………..

RESULT:

Thus the four channel Time division multiplexing and de-multiplexing using DCL-02 kit was performed and obtained its waveform.

AM TRANSMITTER (MODULATOR)



3. AM MODULATOR AND DEMODULATOR

AIM:

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

APPARATUS REQUIRED:

S.No.	Name of the Equipment/ Component	Range	Quantity
1.	AM Transmitter kit - ACL 01	-	1
2.	AM Receiver kit - ACL 01	-	1
3.	CRO	30 MHz	1
4.	Power supply	5V, ±12V	1
5.	Patch chords	-	Required

THEORY:

MODULATION 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. The carrier frequency fc must be much greater than the highest frequency components fm of the message signal m (t) i.e. fc>>fm. The modulation index must be less than unity. If the modulation index is greater than unity, the carrier wave becomes over modulated.

The modulating, carrier and modulated signals are given by

Vm(t) =Vmsinωmt ; VC(t) = VCsinωCt ; VAM(t) = VC (1+ma sinωmt) sinωCt

The modulation index is given by, ma = Vm / VC.

Vm = Vmax – Vmin and VC = Vmax + Vmin

The amplitude of the modulated signal is given by,

Where Vm = maximum amplitude of modulating signal, VC = maximum amplitude of carrier signal, Vmax = maximum variation of AM signal, Vmin = minimum variation of AM signal

DEMODULATION THEORY:

Demodulation is the reverse process of modulation. The detector circuit is employed to separate the carrier wave and eliminate the side bands. Since the envelope of an AM wave has the same shape as the message, independent of the carrier frequency and phase, demodulation can be accomplished by extracting envelope. The depth of modulation at the detector output greater than unity and circuit impedance is less than circuit load (Rl>Zm) results in clipping of negative peaks of modulating signal. It is called “negative clipping “.

TABULATION:

Parameter	Amplitude (V)	Time Period in seconds	Frequency in Hz
Message signal			
Carrier signal			
Modulated signal			
Demodulated signal			

Calculation of modulation index:

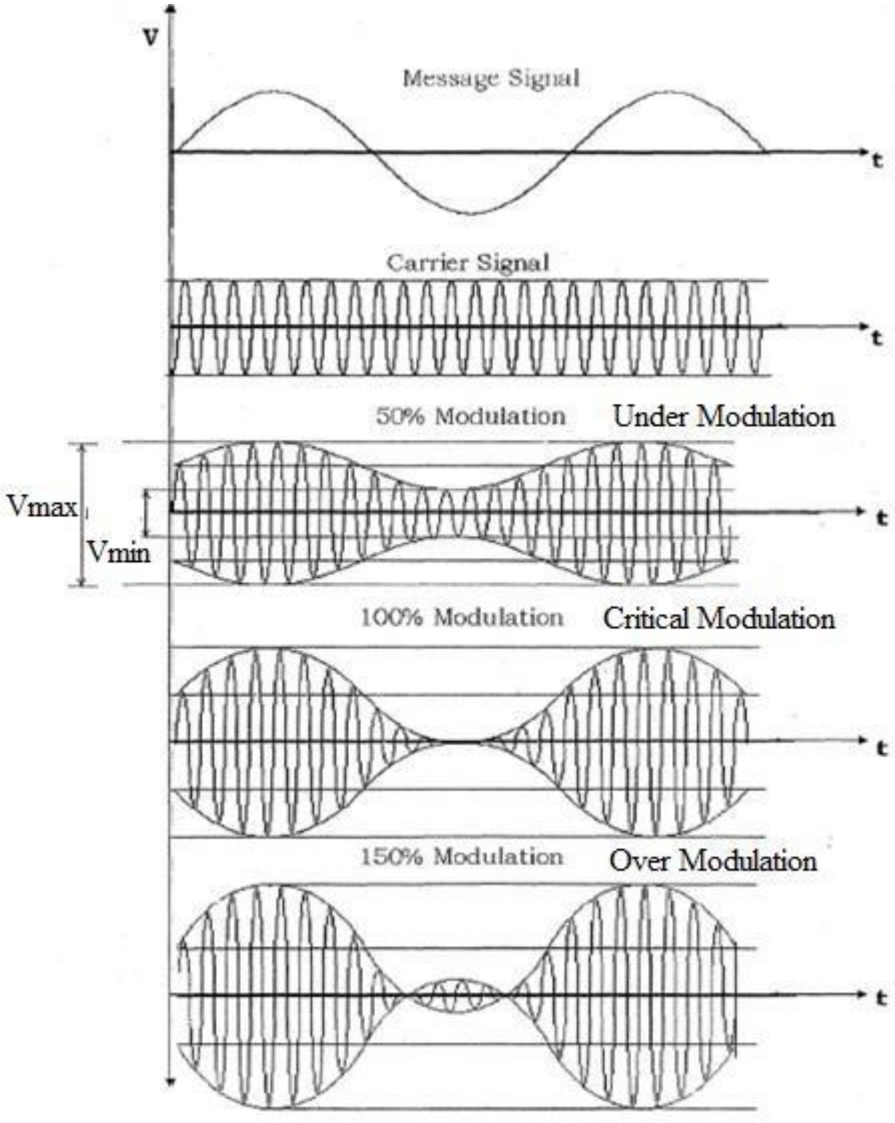
Practical calculation

$$MI = \frac{V\ max - Vmin}{Vmax + Vmin}$$

Theoretical calculation

$$MI = \left(\frac{Vc}{Vm}\right) * 100$$

MODEL GRAPH:



PROCEDURE:

1. The circuit wiring is done as shown in diagram
2. A modulating signal input given to the Amplitude modulator
3. Now increase the amplitude of the modulating signal to the required level.
4. The amplitude and the time duration of the modulating signal are observed using CRO.
5. Finally the amplitude modulated output is observed from the output of amplitude modulator stage and the amplitude and time duration of the AM wave are noted down.
6. Calculate the modulation index by using the formula and verify them.

7. The final demodulated signal is viewed using CRO at the output of audio power amplifier stage. Also the amplitude and time duration of the demodulated wave are noted down.

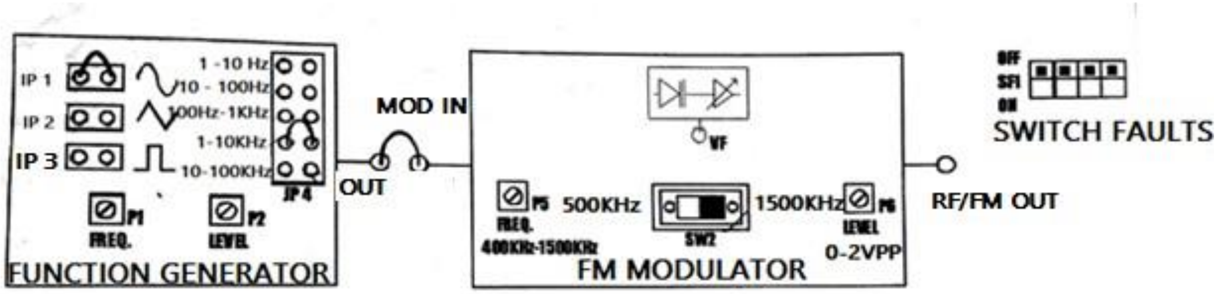
VIVA QUESTIONS:

1. What is the need of modulation and demodulation?
2. What is the range of frequency in commercial AM broadcasting?
3. What is analog modulation various techniques?
4. What is the difference between detector and demodulator?
5. Define the term Modulation index.
6. What are the main components of a RF receiver?
7. What is the Difference between Coherent and Non-coherent Demodulation?
8. Why the Intermediate Frequency Should is Carefully Chosen As?
9. What is the function of an Automatic Gain Control of the AM Receiver?
10. Define Amplitude Modulation?

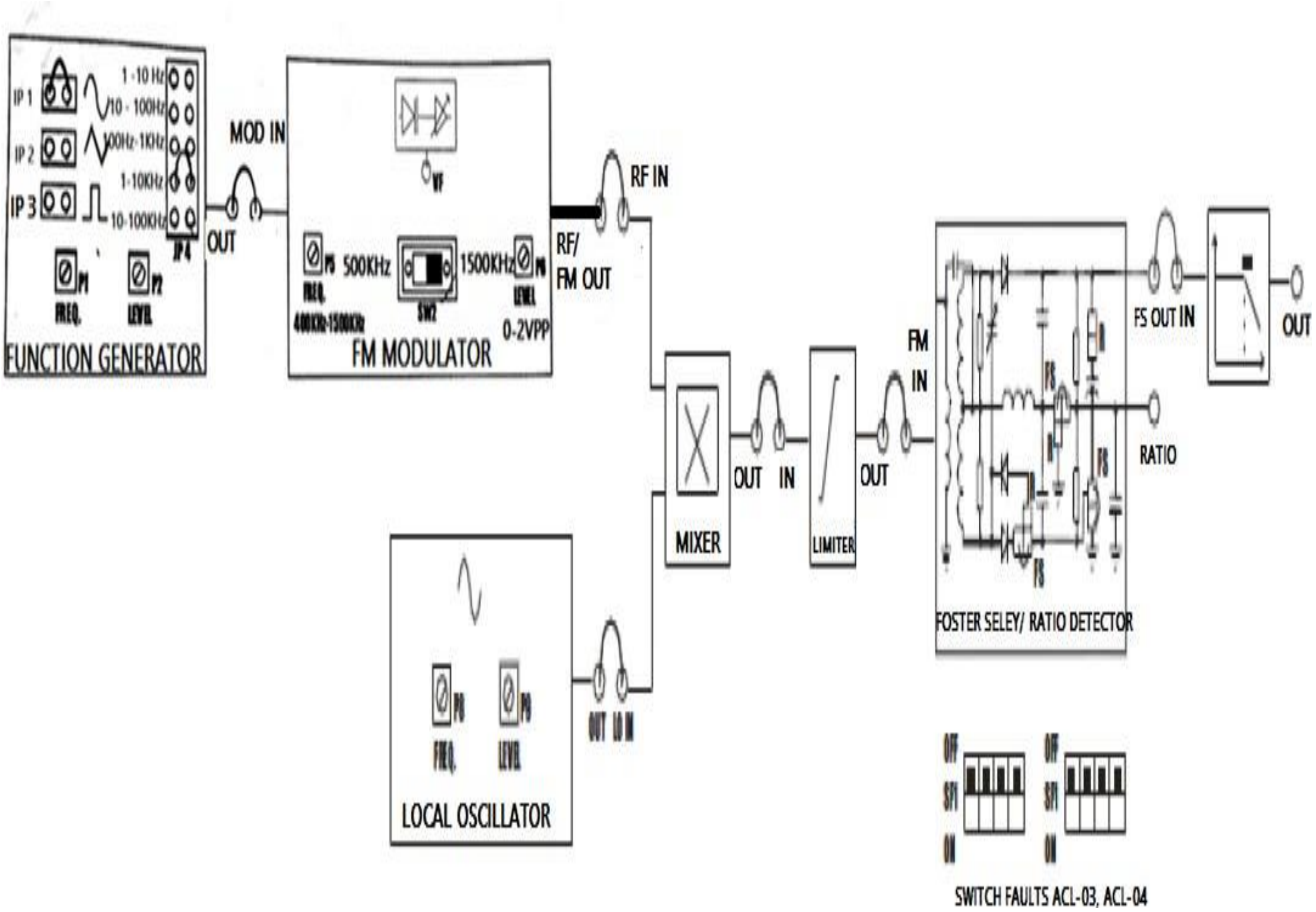
RESULT:

Thus the AM signal was transmitted using AM trainer kit and the AM signal detected using AM detector kit. The calculated modulation index ma =_____________.

FM MODULATOR



FM DEMODULATOR



4. FMMODULATION AND DEMODULATION

AIM:

To plot the modulation characteristics of FM modulator and demodulator and also to Observe and measure frequency deviation and modulation index of FM.

APPARATUS REQUIRED:

S.No.	Name of the Equipment/ Component	Range	Quantity
1.	FM Transmitter kit - ACL 03	-	1
2.	FM Receiver kit - ACL 04	-	1
3.	CRO	30 MHz	1
4.	Power supply	5V, ±12V	1
5.	Patch chords	-	Required

THEORY:

Frequency modulation is a type of modulation in which the frequency of the high frequency (carrier) is varied in accordance with the instantaneous value of the modulating signal.

FREQUENCY MODULATION GENERATION:

The circuits used to generate a frequency modulation must vary the frequency of a high frequency signal (carrier) as function of the amplitude of a low frequency signal (modulating signal). In practice there are two main methods used to generate FM.

DIRECT METHOD

An oscilloscope is used in which the reactance of one of the elements of the resonant circuit depends on the modulating voltage. The most common device with variable reactance is the Varactor or Varicap, which is a particular diode which capacity varies as function of the reverse bias voltage. The frequency of the carrier is established with AFC circuits (Automated frequency control) or PLL (Phase locked loop).

INDIRECT METHOD:

The FM is obtained in this case by a phase modulation, after the modulating signal has been integrated. In this phase modulator the carrier can be generated by a quartz oscillator, and so its frequency stabilization is easier. In the circuit used for the exercise, the frequency modulation is generated by a Hartley oscillator, which frequency is determined by a fixed inductance and by capacity (variable) supplied by varicap diodes.

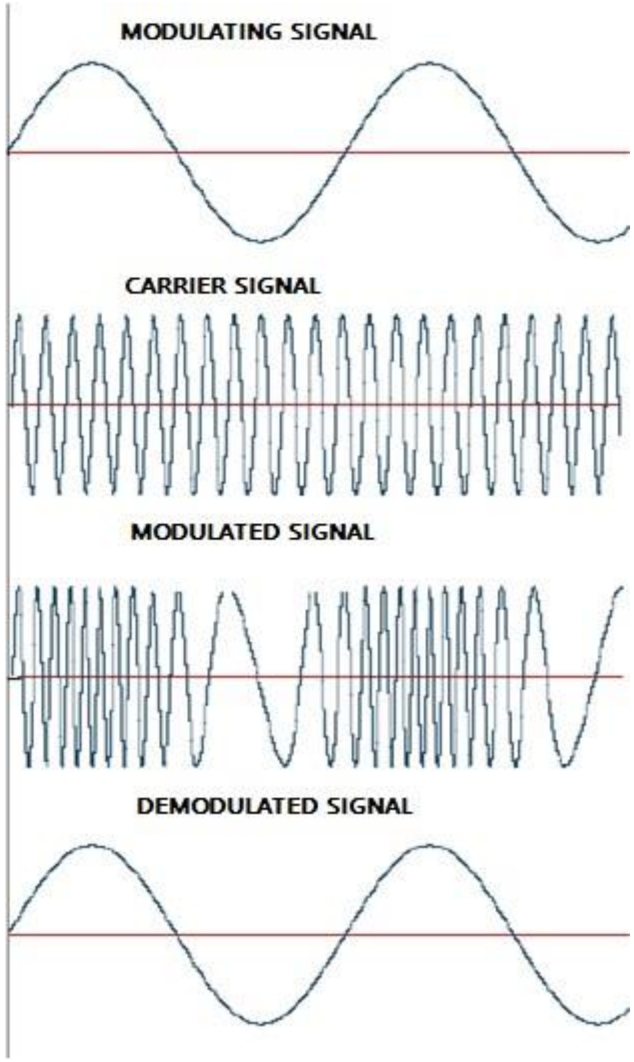
FREQUENCY DEVIATION □*f* and MODULATION INDEX *fm:*

The frequency deviation □*f* represents the maximum shift between the modulated signal frequency, over and under the frequency of the carrier.

TABULATION

Parameter	Amplitude (V)	Time Period in seconds	Frequency in Hz
Message signal			
Carrier signal			
Modulated signal		Tmin =	Fmin =
		Tmax =	Fmax =
Demodulated signal			

MODEL GRAPH



$$\Delta f = \frac{f_{\max} - f_{\max}}{2}$$

We define modulation index m $_f$ the ratio between □f and the modulating frequency f.

$$fm = \frac{\Delta f}{f}$$

PROCEDURE:

1. Connect the power supply with proper polarity to the kit. While connecting this ensures that the Power supply is OFF.
2. Switch on the power supply and carry out the following presetting as shownincircuit Diagram.
3. In the FM modulator set the level about 2Vpp and frequency knob to theminimum and switch on 1500 KHz.

4. Observe the FM modulated waveform from the RF/FM output of the FMmodulator measure frequency deviation and modulation index of FM.
5. For demodulation switch on the demodulator and carry out the followingdemodulation connection as shown in circuit diagram.
6. Observe the demodulated waveform and plot the graph.

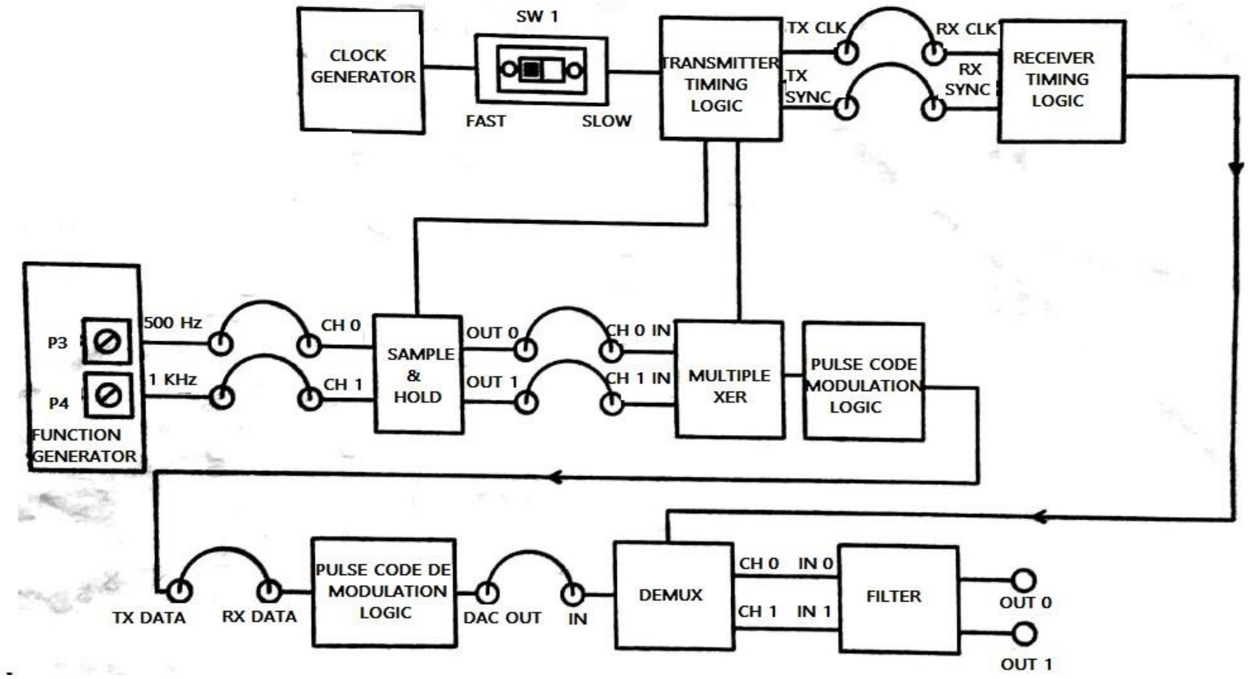
VIVA QUESTION

1. Why frequency modulation is better than amplitude modulation?
2. What is the application of FM?
3. What is PLL?
4. What is FM modulation?
5. What is FM demodulation?
6. Define frequency deviation in FM?
7. Difference between narrowband and wideband FM?
8. What are the advantages of FM over AM?
9. What are the disadvantages of FM over AM?
10. Mention the types of digital modulation.

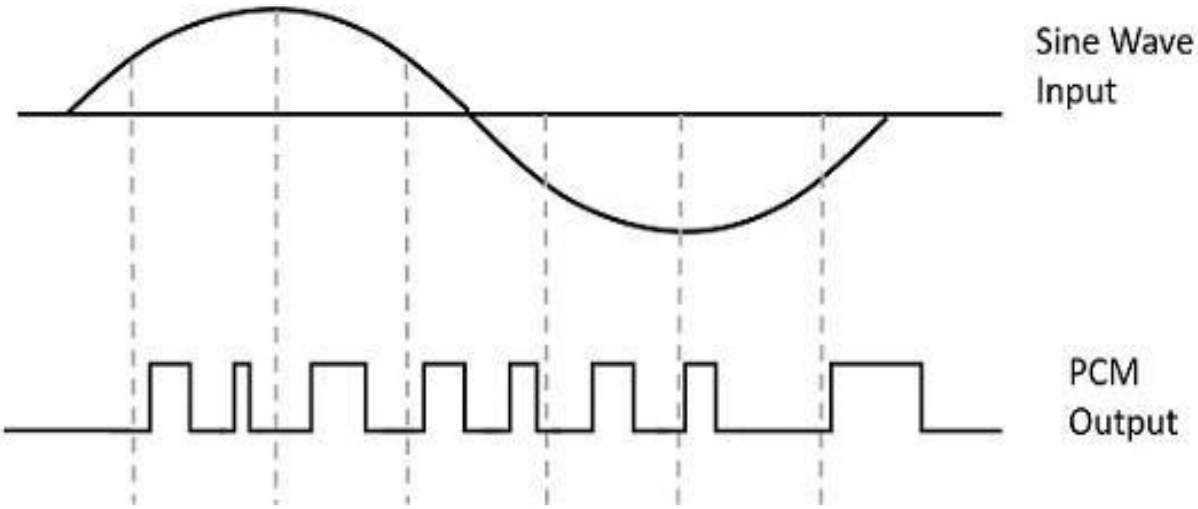
RESULT:

Thus the modulation characteristics of FM modulator and demodulator are observed and plotted.

CONNECTION DIAGRAM:


CLOCK GENERATOR
SW 1
FAST
SLOW
TRANSMITTER TIMING LOGIC
TX CLK
TX SYNC
RX CLK
RX SYNC
RECEIVER TIMING LOGIC
P3
P4
500 Hz
1 KHz
FUNCTION GENERATOR
CH 0
CH 1
SAMPLE & HOLD
OUT 0
OUT 1
CH 0 IN
CH 1 IN
MULTIPLE XER
PULSE CODE MODULATION LOGIC
TX DATA
RX DATA
PULSE CODE DE MODULATION LOGIC
DAC OUT
IN
DEMUX
CH 0
CH 1
IN 0
IN 1
FILTER
OUT 0
OUT 1

MODEL GRAPH:


Sine Wave Input
PCM Output

5. PULSE CODE MODULATION AND DEMODULATION

AIM:

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

APPARATUS REQUIRED:

S.No.	Name of the Equipment/ Component	Range	Quantity
1.	PCM Trainer kit - DCL 03, DCL 04	-	1
2.	CRO	30 MHz	1
3.	Power supply	5V, ±12V	1
4.	Patch chords	-	Required

THEORY:

Pulse code modulation is known as digital pulse modulation technique. It is the process in which the message signal is sampled and the amplitude of each sample is rounded off to the nearest one of the finite set of allowable values. It consists of three main parts transmitter, transmitter path and receiver. The essential operation in the transmitter of a PCM system are sampling, quantizing and encoding. The band pass filter limits the frequency of the analog input signal. The sample and hold circuit periodically samples the analog input signal and converts those to a multi-level PAM signal. The ADC converts PAM samples to parallel PCM codes which are converted to serial binary data in parallel to serial converter and then outputted on the transmission line as serial digital pulse. The transmission line repeaters are placed at prescribed distance to regenerate the digital pulse.

In the receiver serial to parallel converter converts serial pulse received from the transmission line to parallel PCM codes. The DAC converts the parallel PCM codes to multi-level PAM signals. The hold circuit is basically a Low Pass Filter that converts the PAM signal back to its original analog form.

ADVANTAGES:

1. Secrecy
2. Noise resistant and hence free from channel interference

DISADVANTAGE:

1. Requires more bandwidth

APPLICATION:

1. Compact DISC for storage2. Military Applications.

TABULATION

Parameters	Amplitude in V	Time period in Sec	
		TON	TOFF
Input Signal 1(500Hz)			
Input Signal 1(1KHz)			
Sample & Hold OUT 0			
Sample & Hold OUT 1			
Multiplexer CLK1			
Multiplexer CLK1			
Multiplexed Data MUX OUT			
TX Data			
TX CLK			
TX SYNC			
RX Data			
RX CLK			
RX SYNC			
DAC OUT			
Demultiplexer CLK1			
Demultiplexer CLK2			
Demultiplexed Data CH 0			
Demultiplexed Data CH 1			
Received Signal OUT 0			
Received Signal OUT 1			

PROCEDURE:

1. Refer the block diagram and carry out the following connections and switch settings.
2. Connect the power supply with power polarity to the kit and switch it ‘ON’.
3. Put the switch sw1 to ‘FAST’ mode.
4. Select 500Hz and 1 KHz sine wave signals generated on board.
5. Connect the signals to ‘CH0’ and ‘CH1’ of sample and hold circuit.
6. The output of sample and hold circuit ‘OUT0’ and ‘OUT1’ are given to the multiplexer, then to the pulse code modulation logic.
7. The output ‘TXDATA’ is connected to ‘RXDATA’ of pulse code demodulation logic.
8. The output ‘DAC OUT’ is given to ‘IN’ of ‘DEMUX’.
9. Then the output of ‘DEMUX’ –‘OUT0’ and ‘OUT1’ are given to ‘IN0’ and ‘IN1’ of filter.
10. Observe the pulse code demodulated output at ‘OUT0’ and ‘OUT1’.

VIVA QUESTIONS:

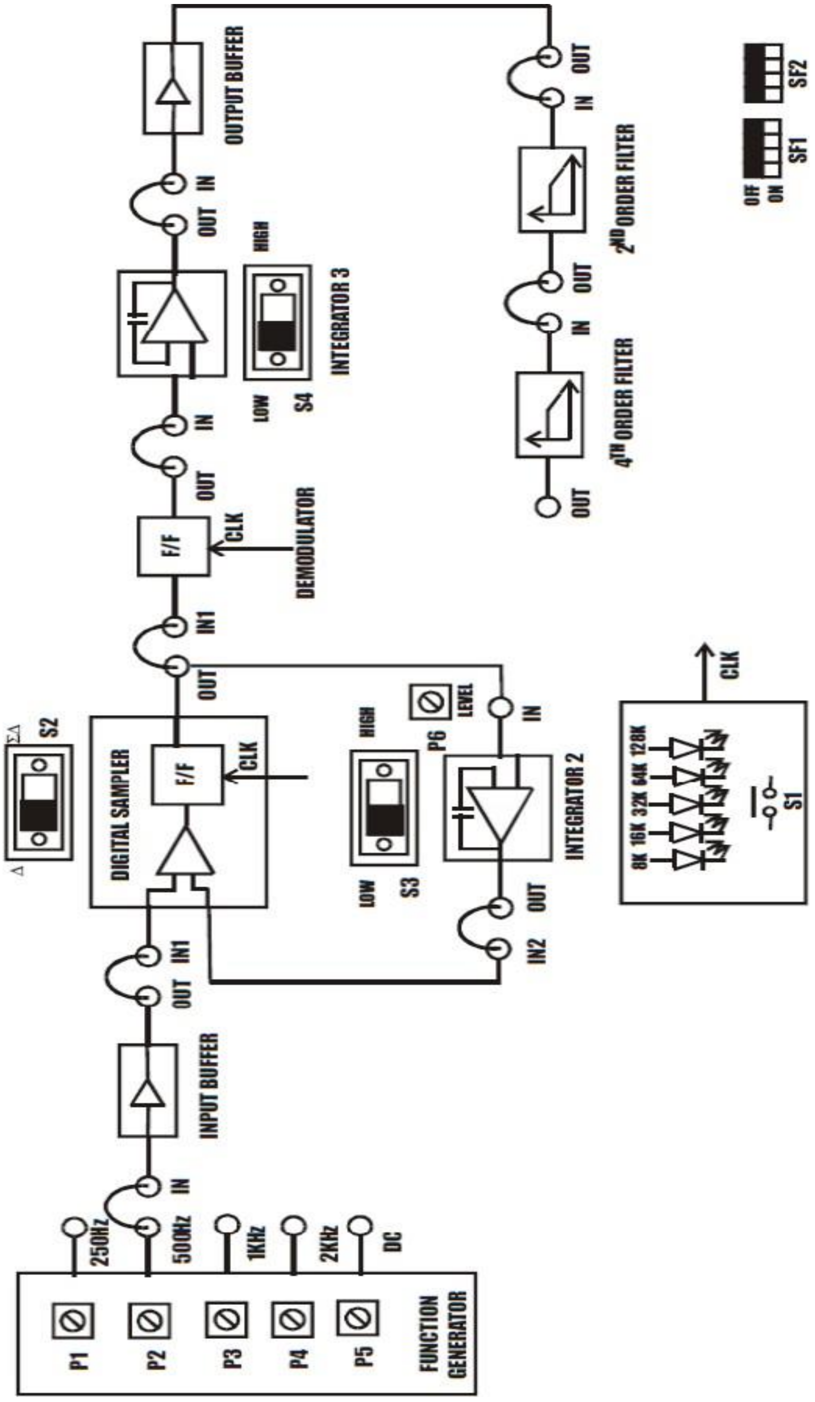
1. What is PCM?
2. How bits are needed to encode N different levels?
3. Define step size?
4. How to calculate Step size in PCM?
5. Define Quantization error.
6. What is the max value of Quantization error?
7. What are the applications of PCM?
8. What are the disadvantages of Pulse code modulation?
9. Define Differential pulse code modulation?
10. Why DPCM is better than PCM?

RESULT

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

CONNECTION DIAGRAM

DELTA MODULATION AND DEMODULATION



6. DELTA MODULATION AND DEMODULATION

AIM

To transmit an analog message signal in its digital form and again reconstruct back the original analog message signal at receiver by using Delta modulator.

APPARATUS REQUIRED:

S.No	Name of the Equipment/ Component	Range	Quantity
1.	Delta modulation/ demodulation trainer kit, DCL 07	-	1
2.	CRO	30 MHz	1
3.	Power supply	5V, ±12V	1
4.	Patch chords	-	Required

THEORY:

Delta modulation uses a single bit PCM code to achieve digital transmission of analogsignal. With conventional PCM, each code is a binary representative of both the sign and magnitude of a particular sample. The algorithm of delta modulation is simple if the current sample is smaller than the previous sample a logic0 is transmitted. If the current sample is larger than the previous sample logic 1 is transmitted.

ADVANTAGES:

1. Simple system/circuitry - Cheap
2. Single bit encoding allows us to increase the sampling rate or to transmit more information at some sampling rate for the given system BW.

DISADVANTAGES:

1. Noise and distortion.
2. Major drawback is that it is unable to pass DC information.

APPLICATIONS:

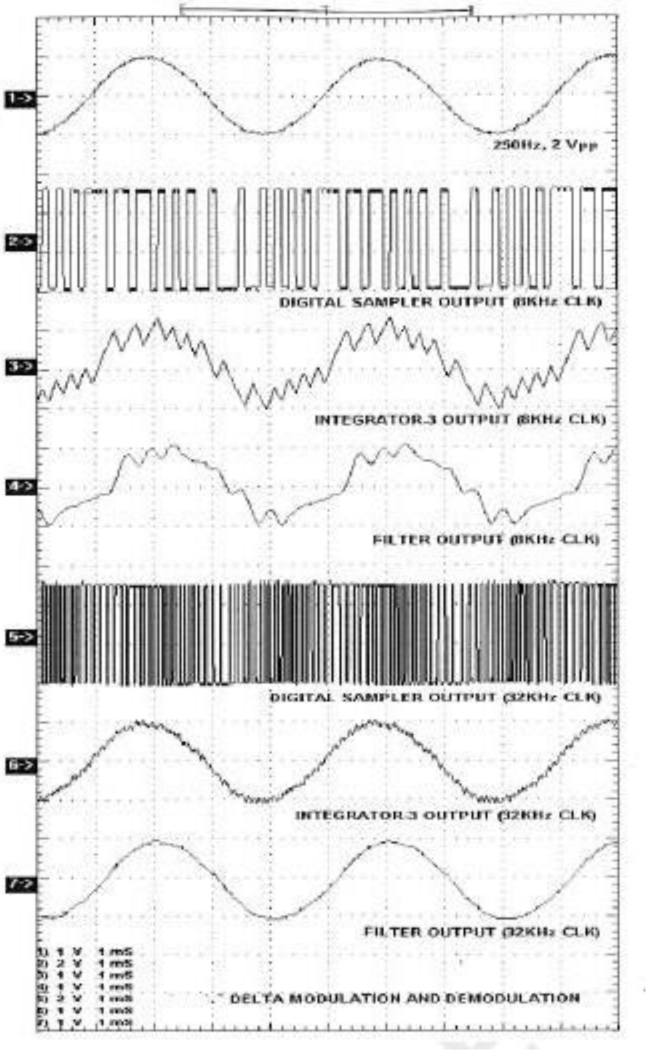
Digital voice storage, Voice transmission, Radio communication devices such TV remotes.

Adaptive delta modulation is delta modulation system where the step size of DAC isautomatically varied, depending on the amplitude characteristics of the analog input signal. A common algorithm for an adaptive delta modulator is when three consecutive 1s or 0soccurs the step size of the DAC is increased or decreased by a factor of 1.5

APPLICATION:

Audio communication system

MODEL GRAPH



TABULATION

Parameter	Amplitude (V)	Time Period in seconds	Frequency in Hz
Input signal			
Integrator 1 output			
Sampler output			
Integrator 1 output			
Filter output			
Demodulated output			

PROCEDURE:

1. Refer to the block diagram and carry out the following connections
2. Connect the power supply with proper polarity to the kit DCL-07 and switch it ON.
3. Select the sine wave input 250Hz or through port P1 and connect port 250Hz to port IN of input buffer.
4. Connect the output of buffer ‘OUT’ to digital samples input port ‘IN1’.
5. Then select clock rate of 8 KHz by pressing switch S1. Selected clock is indicated by LED glow.
6. Keep switch S2 in (Δ) deltaposition.
7. Connect output of a digital sampler port ‘OUT’ to input port ‘IN’ of integrator.
8. Connect output of integrator 1 port ‘OUT’ to input port ‘IN2’ of digital sampler.
9. The digital sampler ‘OUT’ is given to the input of the output buffer.
10. The output of the buffer is given to the second order input of the fourth order Butterworth filter.

VIVA QUESTIONS:

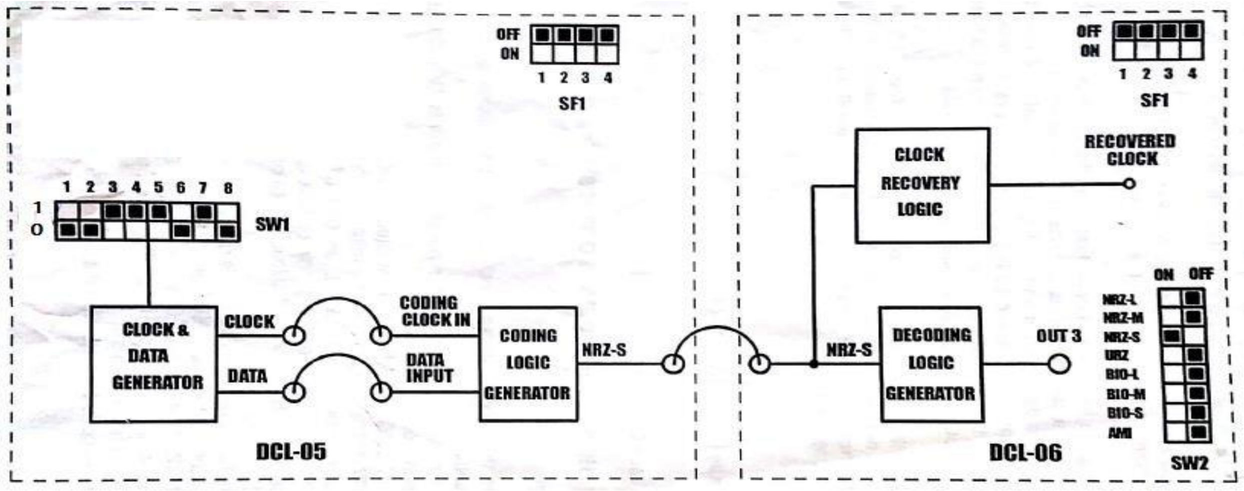
1. Define Delta modulation? Why it is better?
2. What is granular noise? Define slope overload?
3. When granular noise and slope overload occur in Delta modulation?
4. What is Adaptive Delta Modulation and what are the advantages?
5. Compare all Digital pulse modulation techniques (PCM, DPCM, DM and ADM)
6. In digital transmission, the modulation technique that requires minimum bandwidth is …………
7. In Delta Modulation, the bit rate is ………….
8. What are the applications of DM?
9. Mention the advantage of DM?
10. What is temporal waveform coding?

RESULT:

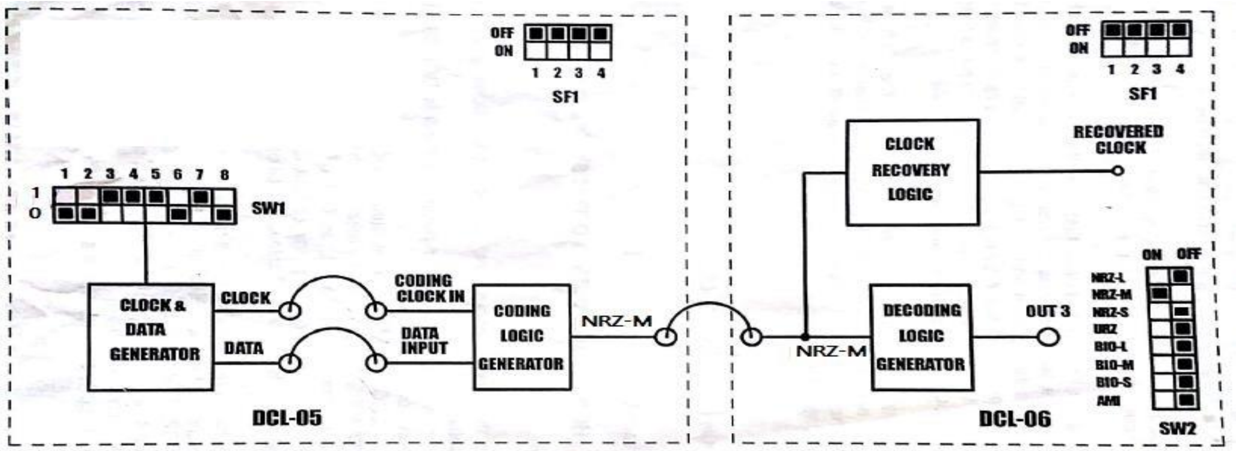
Thus the analog message signal in its digital form was transmitted and again the original analog message signal was reconstructed at receiver by using Delta modulator and Demodulator.

CONNECTION DIAGRAM

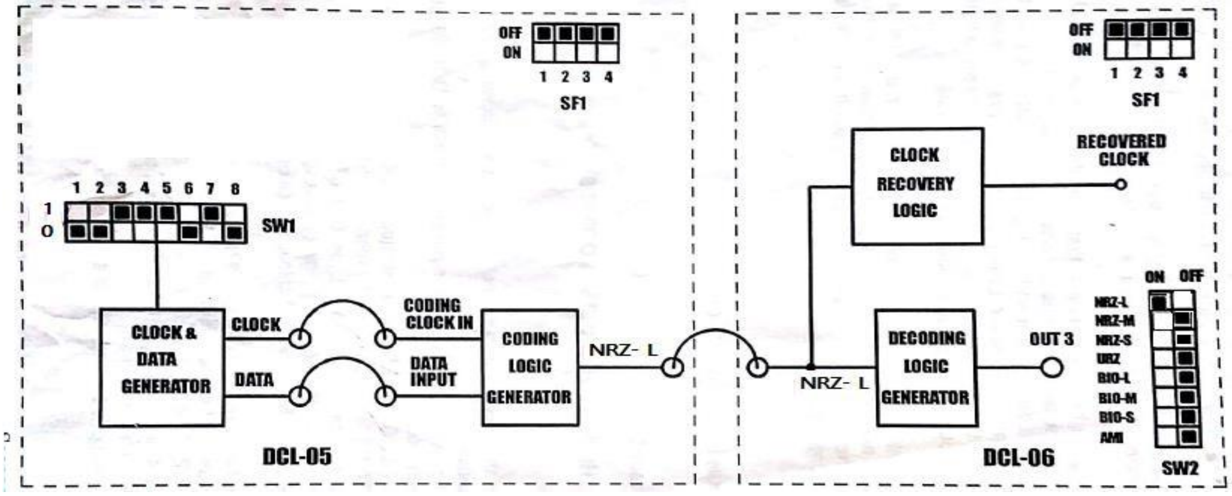
NON-RETURN TO ZERO-SPACE (NRZ-S)



NON-RETURN TO ZERO-MARK (NRZ-M)



NON-RETURN TO ZERO-LEVEL (NRZ-L)



7. LINE CODING SCHEMES

AIM

To study the various encoding and decoding techniques and observe the output waveforms of NRZ-S, NRZ-M, NRZ-L, BIPHASE -L, BIPHASE –M, BIPHASE –S, URZ,AMI using trainer kit DCL-05 and DCL-06.

APPARATUS REQUIRED:

S.No	Name of the Equipment/ Component	Range	Quantity
1.	Line encoding trainer kit DCL-05	-	1
2.	Line decoding trainer kit DCL-06	-	1
3.	CRO	30 MHz	1
4.	Power supply	5V, ±12V	1
5.	Patch chords	-	Required

THEORY:

NON-RETURN TO ZERO signal are the easiest formats that can be generated. These signals do not return to zero with the clock. The frequency component associated with these signals are half that of the clock frequency. The following data formats come under this category. Non-return to zero encoding is commonly used in slow speed communications interfaces for both synchronous and asynchronous transmission. Using NRZ, logic 1 bit is sent as a high value and logic 0 bit is sent as a low value.

a) NON-RETURN TO ZERO-LEVEL (NRZ-L)

This is the most extensively used waveform in digital logics. All “ones” are represented by “high” and all “zeros” by“low”. The data format is directly available at the output of all digital data generation logics and hence very easy to generate. Here all the transitions take place at the rising edge of the clock.

b) NON-RETURN TO ZERO-MARK (NRZ-M)

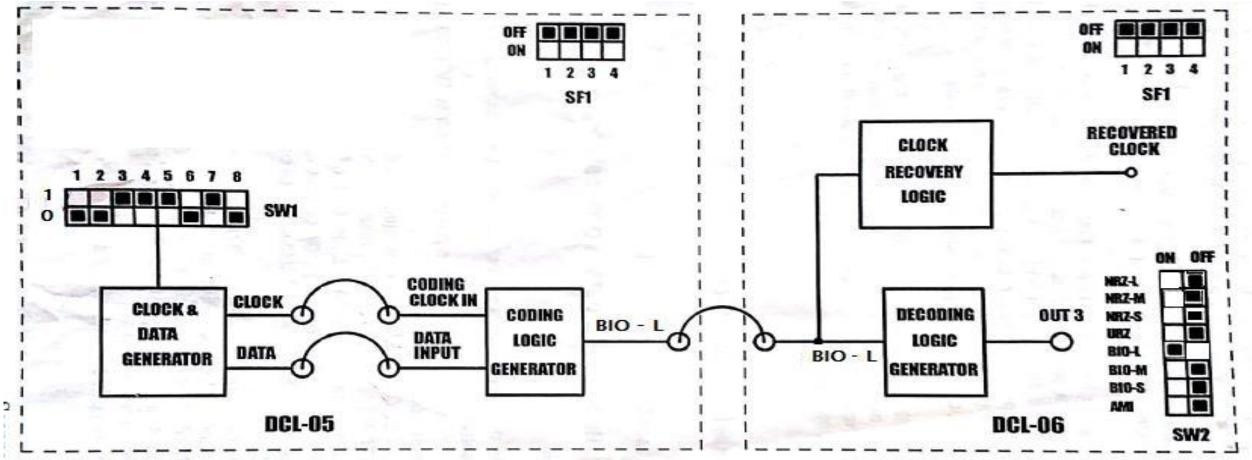
These waveforms are extensively used in tape recording. All „ones‟ are marked by change inlevels and all “zeros” by no transitions, and the transitions take place at the rising edge of the clock.

c) UNIPOLAR AND BIPOLAR

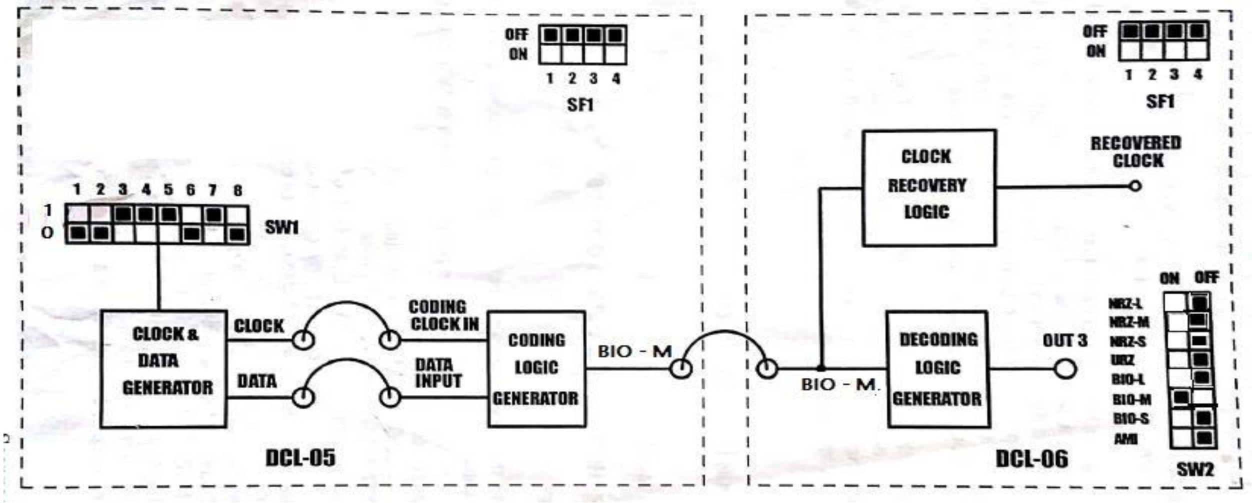
Unipolar signals are those signals, which have transition between 0 to +VCC. Bipolar signals are those signals, which have transition between +VCC to –VCC.

CONNECTION DIAGRAM

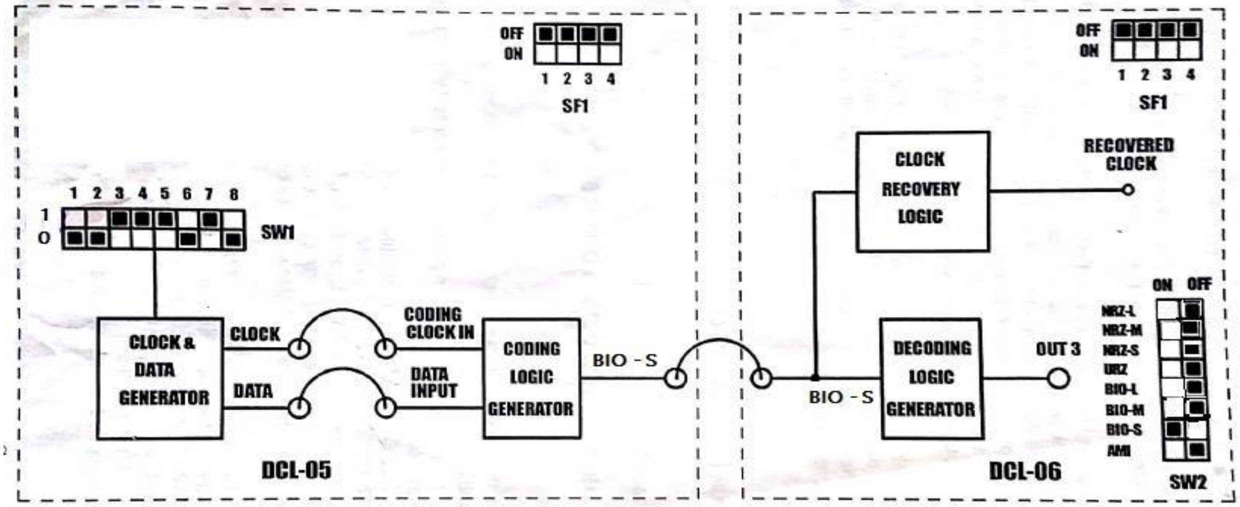
BIPHASE – LINE CODING



BIPHASE MARK CODING



BIPHASE SPACE CODING



d) NON-RETURN TO ZERO-SPACE (NRZ-S)

This type of waveform is marked by change in levels for „zeros" and no transition for "ones" and the transitions take place at the rising edge of the clock. This format is also used in magnetic tape recording.

e) BIPHASE – LINE CODING (BIPHASE -L):

With the Biphase – L one is represented by a half bit wide pulse positioned during thefirst half of the bit interval and a zero is represented by a half bit wide pulse positioned during the second half of the bit interval.

f) BIPHASE MARK CODING(BIPHASE-M):

With the Biphase-M, a transition occurs at the beginning of every bit interval. A "one" isrepresented by a second transition, half bit later, whereas a zero has no second transition.

g) BIPHASE SPACE CODING(BIPHASE-S):

With a Biphase-S, a transition occurs at the beginning of every bit interval. A "zero" ismarked by a second transition, one half bit later; „one" has no second transition.

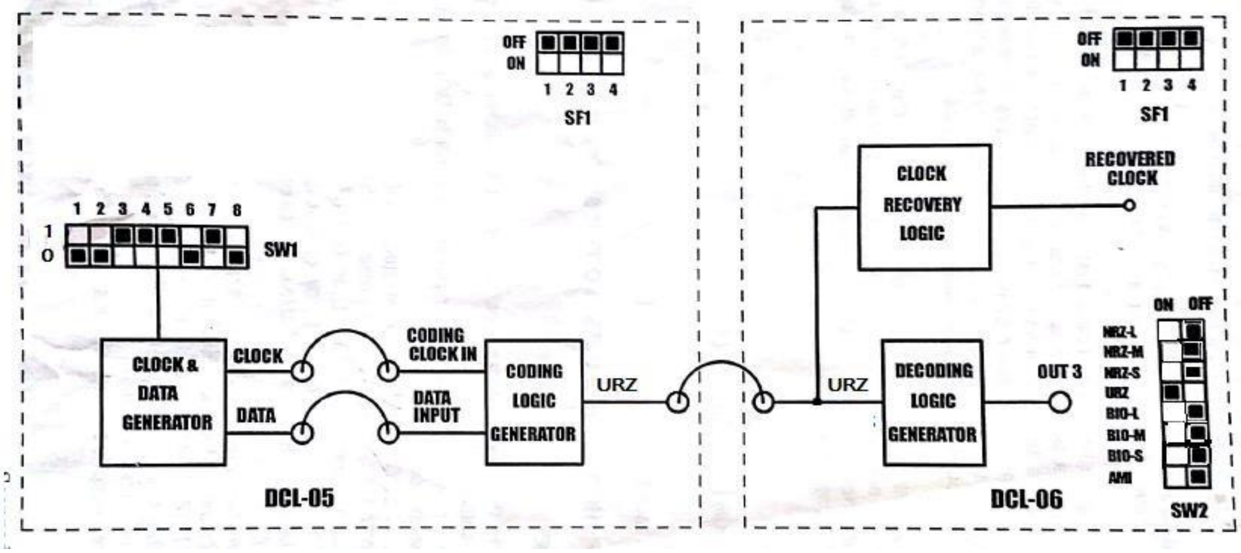
h) RETURN TO ZERO SIGNALS:

These signals are called "Return to Zero signals" since they return to "zero" with theclock. In this category, only one data format, i.e, the unipolar return to zero (URZ); With the URZ a "one" is represented by a half bit wide pulse and a "zero" is represented by the absence of pulse.

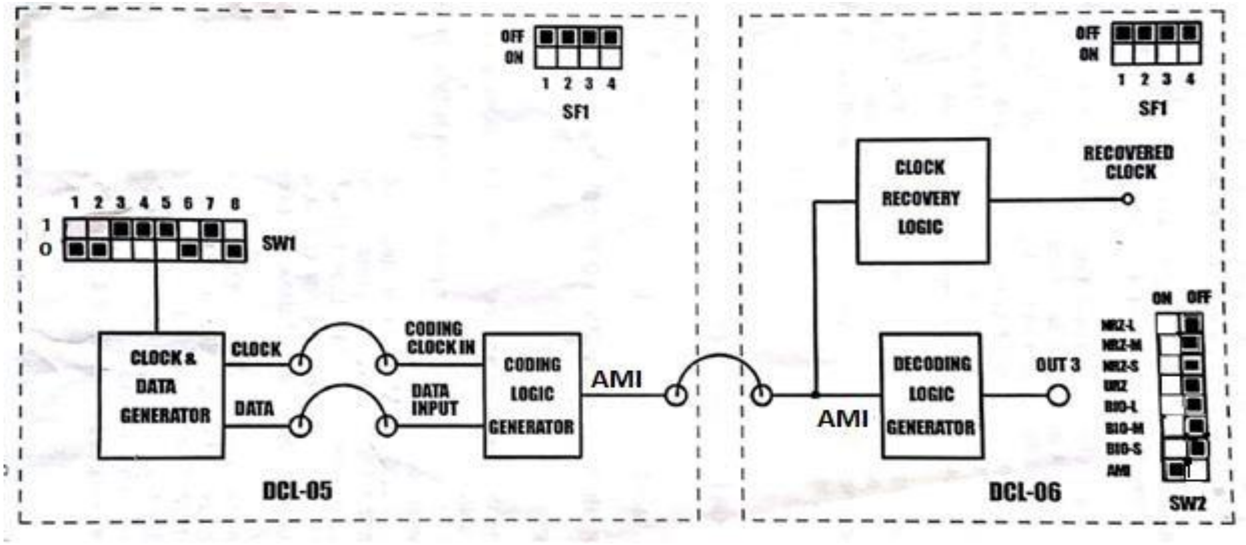
i) MULTILEVEL SIGNALS:

Multilevel signals use three or more levels of voltages to represent the binary digits, "one" and "zero" – instead of normal "highs" and "lows" Return to zero – alternative mark inversion (RZ - AMI) is the most commonly used multilevel signal. This coding scheme is most often used in telemetry systems. In this scheme, "one" are represented by equal amplitude ofalternative pulses, which alternate between a +5 and -5. These alternating pulses return to 0volt, after every half bit interval. The "Zeros" are marked by absence of pulses.

UNIPOLAR RETURN TO ZERO



ALTERNATIVE MARK INVERSION



TABULATION

Type of line coding	Amplitude in V	Time period in Sec	
		TON	TOFF
DATA INPUT			
CLOCK			
NRZ-S			
NRZ-M			
NRZ-L			
BIPHASE -L			
BIPHASE –M			
BIPHASE –S			

URZ			
AMI			

PROCEDURE:

1. Connect power supply in proper polarity to the kits **DCL-05** and **DCL-06** and switch it on.

2. Connect **CLOCK** and **DATA** generated on **DCL-05** to **CODING CLOCK IN** and **DATA INPUT** respectively by means of the patch-chords provided.

3. Connect the coded data **NRZ-L** on **DCL-05** to the corresponding **DATA INPUT NRZ-L,** of the decoding logic on **DCL-06**.

4. Keep the switch **SW2** for **NRZ-L** to **ON** position for decoding logic as shown in the block diagram.

5. Observe the coded and decoded signal on the oscilloscope.

6. Connect the coded data **NRZ-M** on **DCL-05** to the corresponding **DATA INPUT NRZ-M,** of the decoding logic on **DCL-06.**

7. Keep the switch **SW2** for **NRZ –M** to **ON** position for decoding logic as shown in the block diagram.

8. Observe the coded and decoded signal on the oscilloscope.

9. Connect the code data **NRZ-S** on **DCL-05** to the corresponding **DATA INPUT NRZ-S,** of the decoding logic on **DCL-06**.

10. Keep the switch **SW2** for **NRZ-S** to **ON** position for decoding logic as shown in the block diagram.

11. Observe the coded and decoded signal on the oscilloscope.

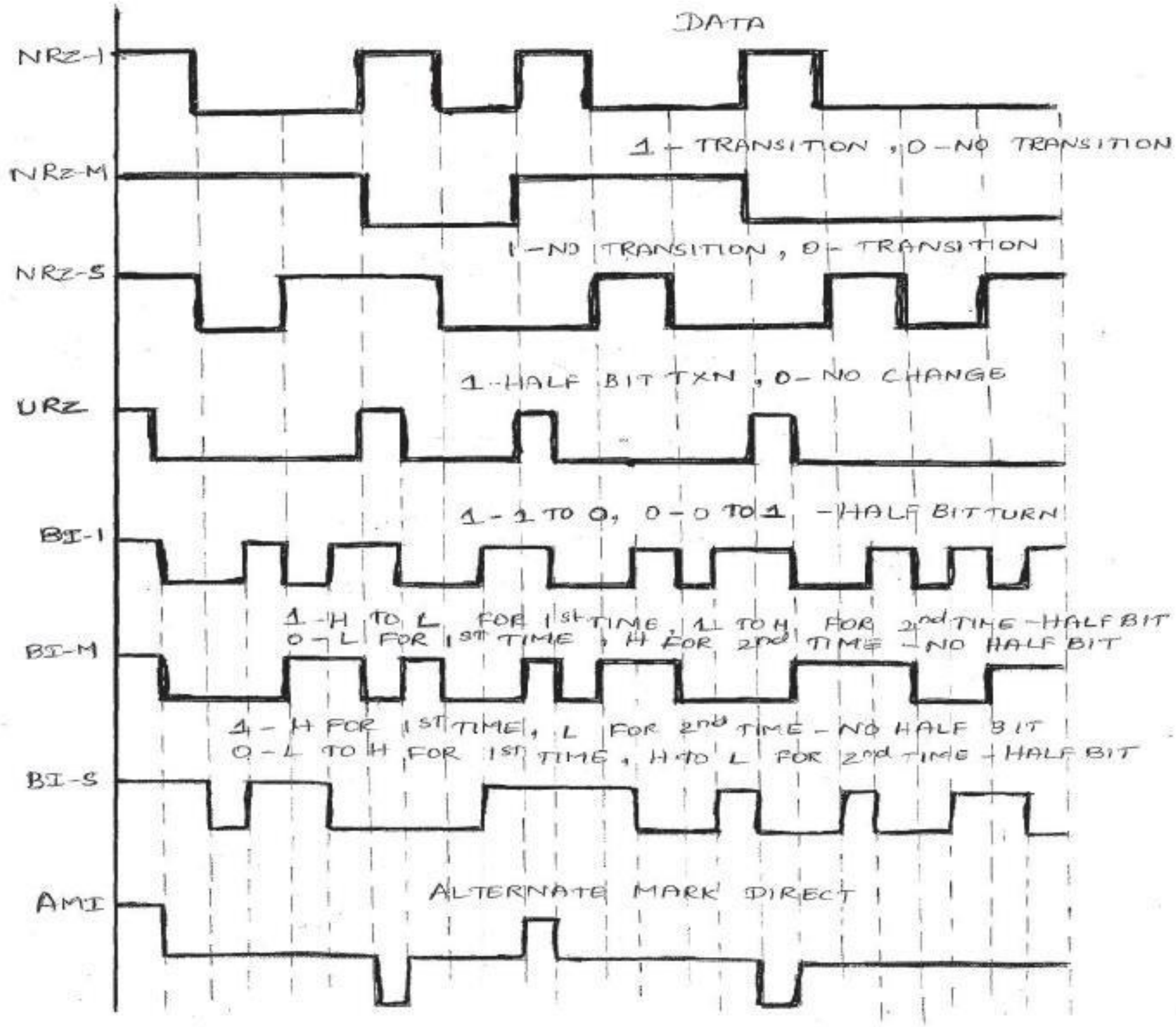
12. Use **RESET** switch for clear data observation if necessary.

13. Unipolar to Bipolar/Bipolar to Unipolar:

a. connect **NRZ-L** signal from **DCL-05** to the input post **IN** Unipolar to Bipolar and Observe the Bipolar output at the post **OUT**.

b. Then connect bipolar output signal to the input post **IN** of Bipolar to Unipolar and Observe Unipolar out at post OUT.

MODEL GRAPH:


DATA
NRZ-I
1 - TRANSITION , 0 - NO TRANSITION
NRZ-M
1 - NO TRANSITION , 0 - TRANSITION
NRZ-S
1 - HALF BIT TXN , 0 - NO CHANGE
URZ
1 - 1 TO 0, 0 - 0 TO 1 - HALF BIT TURN
BI-I
1 - H TO L FOR 1st TIME, 1 L TO H FOR 2nd TIME - HALF BIT
0 - L FOR 1st TIME , H FOR 2nd TIME - NO HALF BIT
BI-M
1 - H FOR 1st TIME, L FOR 2nd TIME - NO HALF BIT
0 - L TO H FOR 1st TIME, H TO L FOR 2nd TIME - HALF BIT
BI-S
ALTERNATE MARK DIRECT
AMI

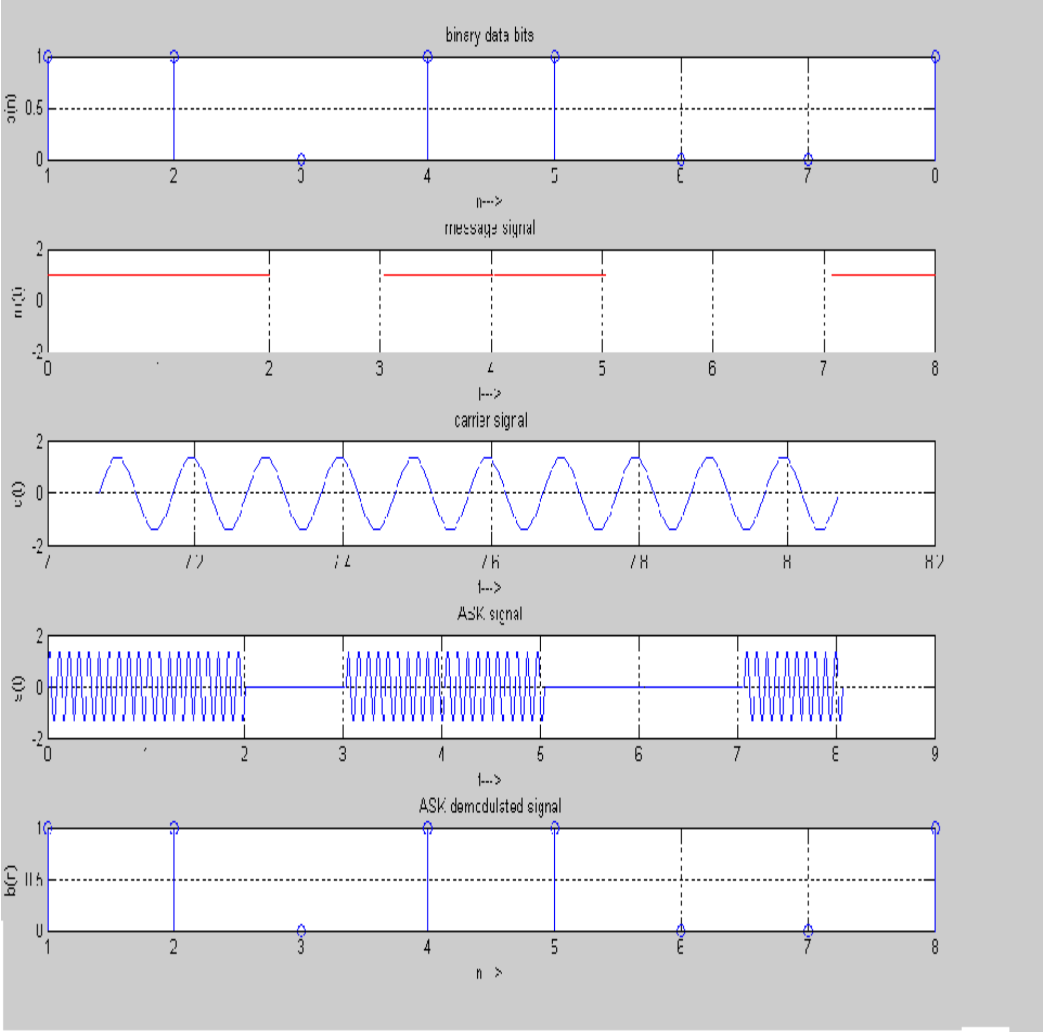
VIVA QUESTIONS

1. What is Line coding and what are the different line coding techniques?
2. What is the difference between Source coding and Line coding?
3. Define ISI (Inter symbol Interference)?
4. What is Matched filter?
5. What is the cause of inter symbol Interference (ISI) and it can be reduced?
6. Define polar encoding?
7. Define bipolar encoding?
8. Define Manchester encoding?
9. What are the Properties of Line Coding?
10. State any two requirements of line code.

RESULT:

Thus the various decoding and encoding schemes are studied and observed its output waveforms.

SIMULATION WAVEFORM



8. SIMULATION OF ASK GENERATION AND DETECTION SCHEME

Aim:

To generate and demodulate amplitude shift keyed (ASK) signal using MATLAB.

Theory

Generation of ASK

Amplitude shift keying - ASK - is a modulation process, which imparts to a sinusoid two or more discrete amplitude levels. These are related to the number of levels adopted by the digital message. For a binary message sequence there are two levels, one of which is typically zero. The data rate is a sub-multiple of the carrier frequency. Thus the modulated waveform consists of bursts of a sinusoid. One of the disadvantages of ASK, compared with FSK and PSK, for example, is that it has not got a constant envelope. This makes its processing (eg, power amplification) more difficult, since linearity becomes an important factor. However, it does make for ease of demodulation with an envelope detector.

Demodulation

ASK signal has a well defined envelope. Thus it is amenable to demodulation by an envelope detector. Some sort of decision-making circuitry is necessary for detecting the message. The signal is recovered by using a correlator and decision making circuitry is used to recover the binary sequence.

Algorithm

Initialization commands

ASK modulation

1. Generate carrier signal.
2. Start FOR loop
3. Generate binary data, message signal(on-off form)
4. Generate ASK modulated signal.
5. Plot message signal and ASK modulated signal.
6. End FOR loop.
7. Plot the binary data and carrier.

ASK demodulation

1. Start FOR loop
2. Perform correlation of ASK signal with carrier to get decision variable
3. Make decision to get demodulated binary data. If x>0, choose ‘1’ else choose ‘0’
4. Plot the demodulated binary data.

Program

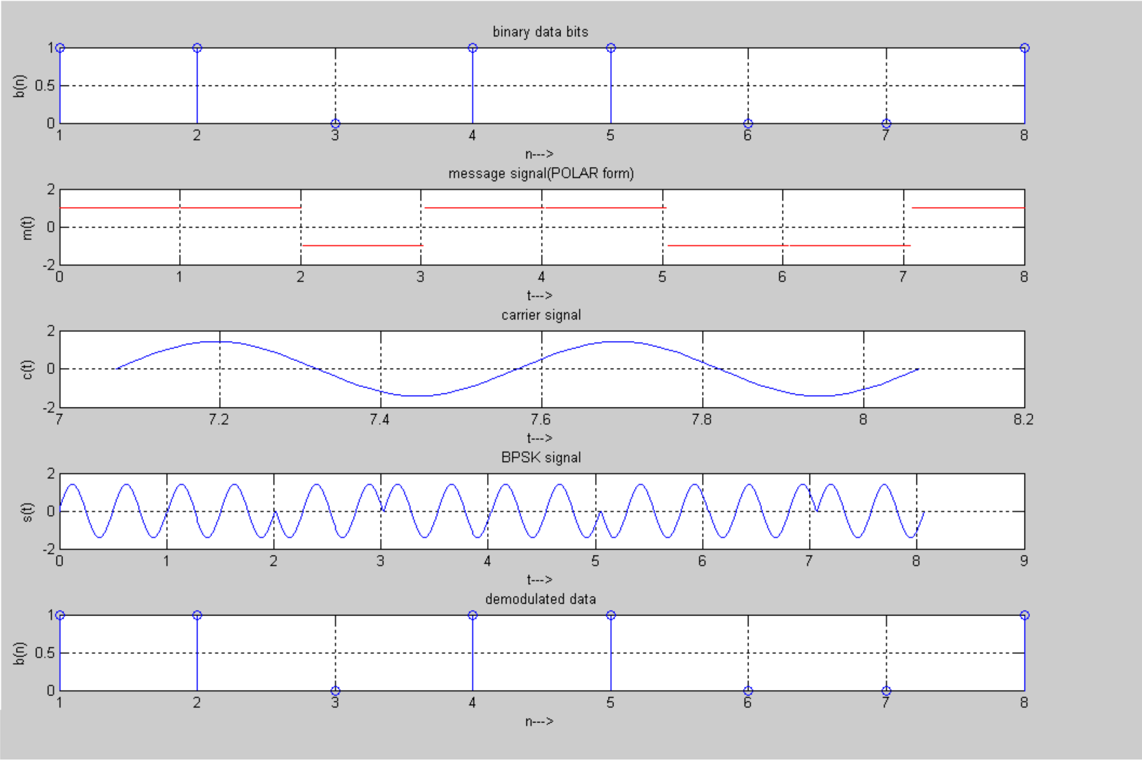
```
%ASK Modulation
clc;
clear all;
close all;
%GENERATE CARRIER SIGNAL
Tb=1; fc=10;
t=0:Tb/100:1;
c=sqrt(2/Tb)*sin(2*pi*fc*t);
%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb
for i=1:N
t=[t1:.01:t2]
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=zeros(1,length(t));
end
message(i,:)=m_s;
%product of carrier and message
ask_sig(i,:)=c.*m_s;
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
%plot the message and ASK signal
subplot(5,1,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
title('message signal');xlabel('t--->');ylabel('m(t)');grid on
hold on
subplot(5,1,4);plot(t,ask_sig(i,:));
title('ASK signal');xlabel('t--->');ylabel('s(t)');grid on
hold on
```

```
end
hold off
%Plot the carrier signal and input binary data
subplot(5,1,3);plot(t,c);
title('carrier signal');xlabel('t--->');ylabel('c(t)');grid on
subplot(5,1,1);stem(m);
title('binary data bits');xlabel('n--->');ylabel('b(n)');grid on
% ASK Demodulation
t1=0;t2=Tb
for i=1:N
t=[t1:Tb/100:t2]
%correlator
x=sum(c.*ask_sig(i,:));
%decision device
if x>0
demod(i)=1;
else
demod(i)=0;
end
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end
%plot demodulated binary data bits
subplot(5,1,5);stem(demod);
title('ASK demodulated signal'); xlabel('n--->');ylabel('b(n)');grid on
```

Result

The program for ASK modulation and demodulation has been simulated in MATLAB and necessary graphs are plotted.

SIMULATION WAVEFORM



9. SIMULATION OF BPSK GENERATION AND DETECTION SCHEMES

Aim:

To generate and demodulate Binary phase shift keyed (BPSK) signal using MATLAB

Generation of PSK signal

PSK is a digital modulation scheme that conveys data by changing, or modulating, the phase of a reference signal (the carrier wave). PSK uses a finite number of phases, each assigned a unique pattern of binary digits. Usually, each phase encodes an equal number of bits. Each pattern of bits forms the symbol that is represented by the particular phase. The demodulator, which is designed specifically for the symbol-set used by the modulator, determines the phase of the received signal and maps it back to the symbol it represents, thus recovering the original data.

In a coherent binary PSK system, the pair of signal $S_1(t)$ and S_2 (t) used to represent binary symbols 1 & 0 are defined by

$S_1 (t) = \sqrt{}2E_b/ T_b \, Cos\ 2\pi f_c t$

$S_2 (t) =\sqrt{}2E_b/T_b (2\pi f_c t+\pi) = - \sqrt{}\ 2E_b/T_b \, Cos\ 2\pi f_c t$ where $0 \leq t< T_b$ and

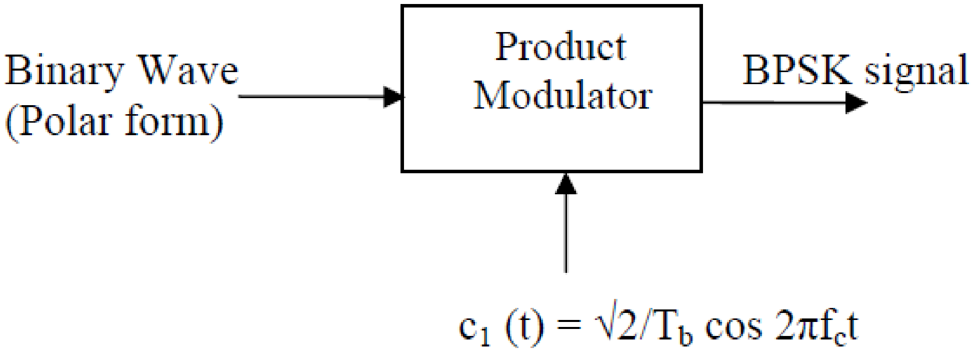
Eb = Transmitted signed energy for bit

The carrier frequency fc $=n/T_b$ for some fixed integer n.

Antipodal Signal:

The pair of sinusoidal waves that differ only in a relative phase shift of 180° are called antipodal signals.

BPSK Transmitter



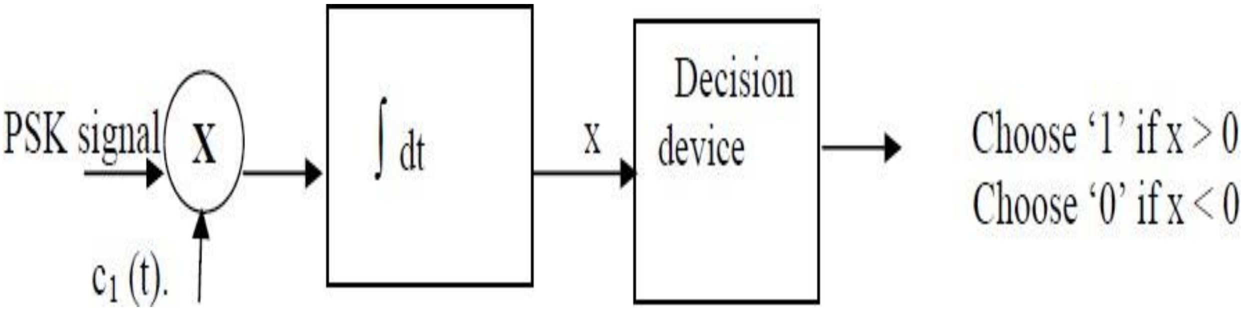
Program

```
% BPSK modulation
clc;
clear all;
close all;
%GENERATE CARRIER SIGNAL
Tb=1;
t=0:Tb/100:Tb;
fc=2;
c=sqrt(2/Tb)*sin(2*pi*fc*t);
%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb
for i=1:N
t=[t1:.01:t2]
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=-1*ones(1,length(t));
end
message(i,:)=m_s;
%product of carrier and message signal
bpsk_sig(i,:)=c.*m_s;
%Plot the message and BPSK modulated signal
subplot(5,1,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
title('message signal(POLAR form)');xlabel('t--->');ylabel('m(t)');
grid on; hold on;
subplot(5,1,4);plot(t,bpsk_sig(i,:));
title('BPSK signal');xlabel('t--->');ylabel('s(t)');
```

grid on; hold on;

The input binary symbols are represented in polar form with symbols 1 & 0 represented by constant amplitude levels √Eb& -√Eb. This binary wave is multiplied by a sinusoidal carrier in a product modulator. The result in a BSPK signal.

BSPK Receiver:



The received BPSK signal is applied to a correlator which is also supplied with a locally generated reference signal c_1 (t). The correlated o/p is compared with a threshold of zero volts. If x> 0, the receiver decides in favour of symbol 1. If x< 0, it decides in favour of symbol 0.

Algorithm

Initialization commands

BPSK modulation

1. Generate carrier signal.
2. Start FOR loop
3. Generate binary data, message signal in polar form
4. Generate PSK modulated signal.
5. Plot message signal and PSK modulated signal.
6. End FOR loop.
7. Plot the binary data and carrier.

BPSK demodulation

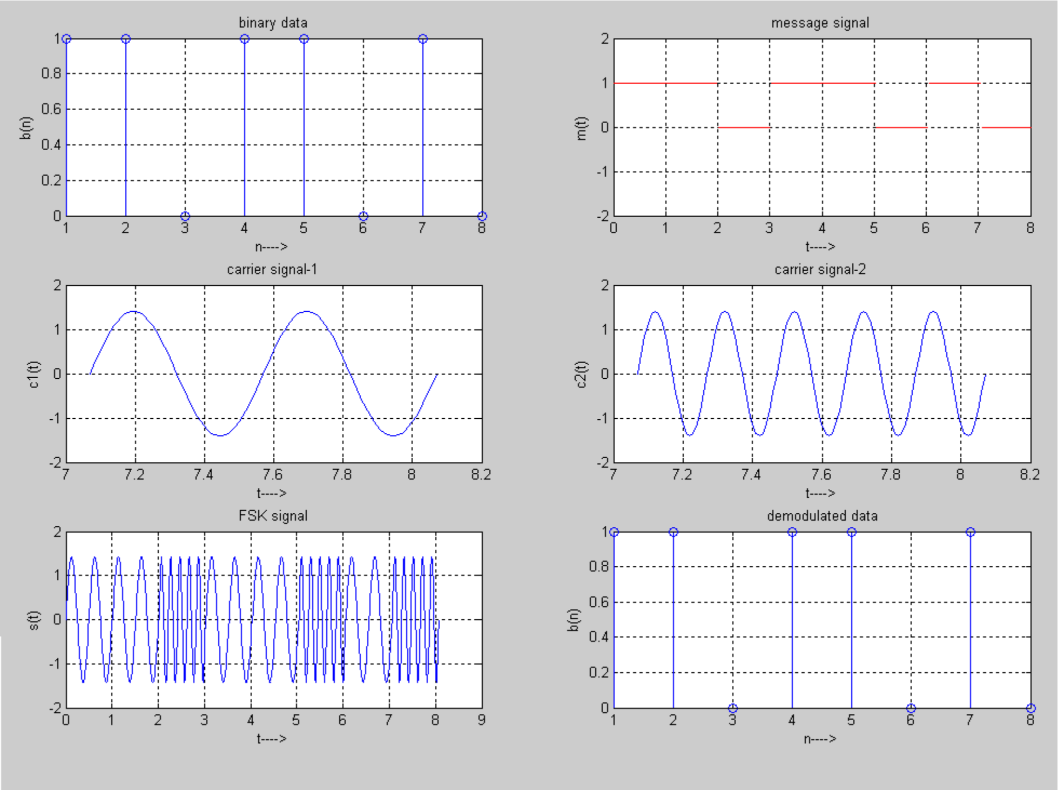
1. Start FOR loop
2. Perform correlation of PSK signal with carrier to get decision variable
3. Make decision to get demodulated binary data. If x>0, choose '1' else choose '0'
4. Plot the demodulated binary data.

```
t1=t1+1.01; t2=t2+1.01;
end
hold off
%plot the input binary data and carrier signal
subplot(5,1,1);stem(m);
title('binary data bits');xlabel('n--->');ylabel('b(n)');
grid on;
subplot(5,1,3);plot(t,c);
title('carrier signal');xlabel('t--->');ylabel('c(t)');
grid on;
% PSK Demodulation
t1=0;t2=Tb
for i=1:N
t=[t1:.01:t2]
%correlator
x=sum(c.*bpsk_sig(i,:));
%decision device
if x>0
demod(i)=1;
else
demod(i)=0;
end
t1=t1+1.01;
t2=t2+1.01;
end
%plot the demodulated data bits
subplot(5,1,5);stem(demod);
title('demodulated data');xlabel('n--->');ylabel('b(n)');
grid on;
```

Result

Thus the program for BPSK modulation and demodulation has been simulated in MATLAB and necessary graphs are plotted.

SIMULATION WAVEFORM



10. SIMULATION OF FSK GENERATION AND DETECTION SCHEME

Aim

To generate and demodulate frequency shift keyed (FSK) signal using MATLAB

Theory

Generation of FSK

Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier wave. 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.

In binary FSK system, symbol 1 & 0 are distinguished from each other by transmitting one of the two sinusoidal waves that differ in frequency by a fixed amount.

Si (t) = $\sqrt{2E/T_b} \cos 2\pi f_1 t$ $0 \leq t \leq T_b$

0 elsewhere

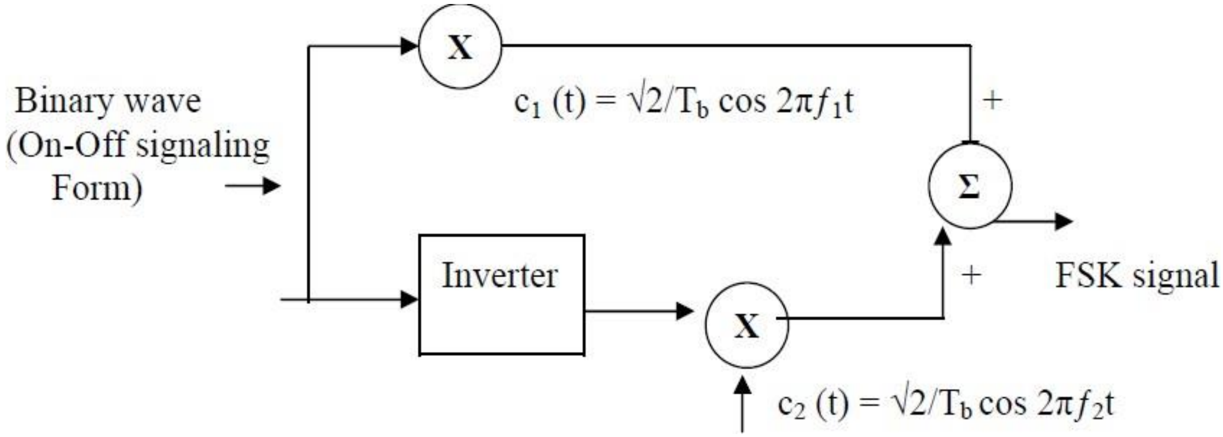
Where i=1, 2 &E_b=Transmitted energy/bit

Transmitted freq= $f_i = (nc+i)/T_b$, and n = constant (integer), T_b = bit interval

Symbol 1 is represented by S_1 (t)

Symbol 0 is represented by S_0 (t)

BFSK Transmitter


Binary wave
(On-Off signaling
Form)
X
c_1 (t) = $\sqrt{2/T_b} \cos 2\pi f_1 t$
+
Σ
Inverter
+
FSK signal
X
c_2 (t) = $\sqrt{2/T_b} \cos 2\pi f_2 t$

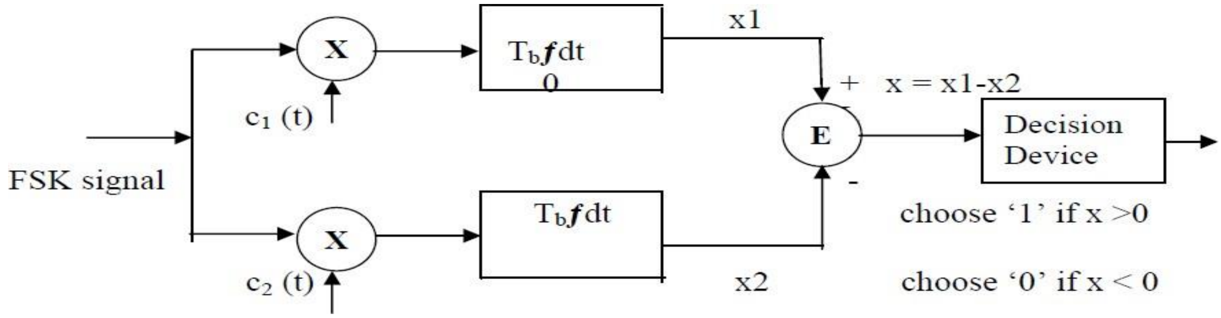
The input binary sequence is represented in its ON-OFF form, with symbol 1 represented by constant amplitude of $\sqrt{E_b}$with & symbol 0 represented by zero volts. By using inverter in the lower channel, we in effect make sure that when symbol 1is at the input, The two frequency f1& f2 are chosen to be equal integer multiples of the bit rate 1/T_b.By summing the upper & lower channel outputs, we get BFSK signal.

Program

```
% FSK Modulation
clc;
clear all;
close all;
%GENERATE CARRIER SIGNAL
Tb=1; fc1=2;fc2=5;
t=0:(Tb/100):Tb;
c1=sqrt(2/Tb)*sin(2*pi*fc1*t);
c2=sqrt(2/Tb)*sin(2*pi*fc2*t);
%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb
for i=1:N
t=[t1:(Tb/100):t2]
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
invm_s=zeros(1,length(t));
else
m(i)=0;
m_s=zeros(1,length(t));
invm_s=ones(1,length(t));
end
message(i,:)=m_s;
%Multiplier
fsk_sig1(i,:)=c1.*m_s;
fsk_sig2(i,:)=c2.*invm_s;
fsk=fsk_sig1+fsk_sig2;
%plotting the message signal and the modulated signal
subplot(3,2,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
```

title('message signal');xlabel('t --->');ylabel('m(t)');grid on;hold on;

BFSK Receiver



The receiver consists of two correlators with common inputs which are supplied with locally generated coherent reference signals c1(t) and c2 (t).

The correlator outputs are then subtracted one from the other, and the resulting difference x is compared with a threshold of zero volts. If x >0, the receiver decides in favour of symbol 1 and if x <0, the receiver decides in favour of symbol 0.

Algorithm

Initialization commands

FSK modulation

1. Generate two carriers signal.
2. Start FOR loop
3. Generate binary data, message signal and inverted message signal
4. Multiply carrier 1 with message signal and carrier 2 with inverted message signal
5. Perform addition to get the FSK modulated signal
6. Plot message signal and FSK modulated signal.
7. End FOR loop.
8. Plot the binary data and carriers.

FSK demodulation

1. Start FOR loop
2. Perform correlation of FSK modulated signal with carrier 1 and carrier 2 to get two decision variables x1 and x2.
3. Make decisionon x = x1-x2 to get demodulated binary data. If x>0, choose '1' else choose '0'.
4. Plot the demodulated binary data.

```
subplot(3,2,5);plot(t,fsk(i,:));
title('FSK signal');xlabel('t--- >');ylabel('s(t)');grid on;hold on;
t1=t1+(Tb+.01); t2=t2+(Tb+.01);
end
hold off
%Plotting binary data bits and carrier signal
subplot(3,2,1);stem(m);
title('binary data');xlabel('n --- >'); ylabel('b(n)');grid on;
subplot(3,2,3);plot(t,c1);
title('carrier signal-1');xlabel('t --- >');ylabel('c1(t)');grid on;
subplot(3,2,4);plot(t,c2);
title('carrier signal-2');xlabel('t --- >');ylabel('c2(t)');grid on;
```

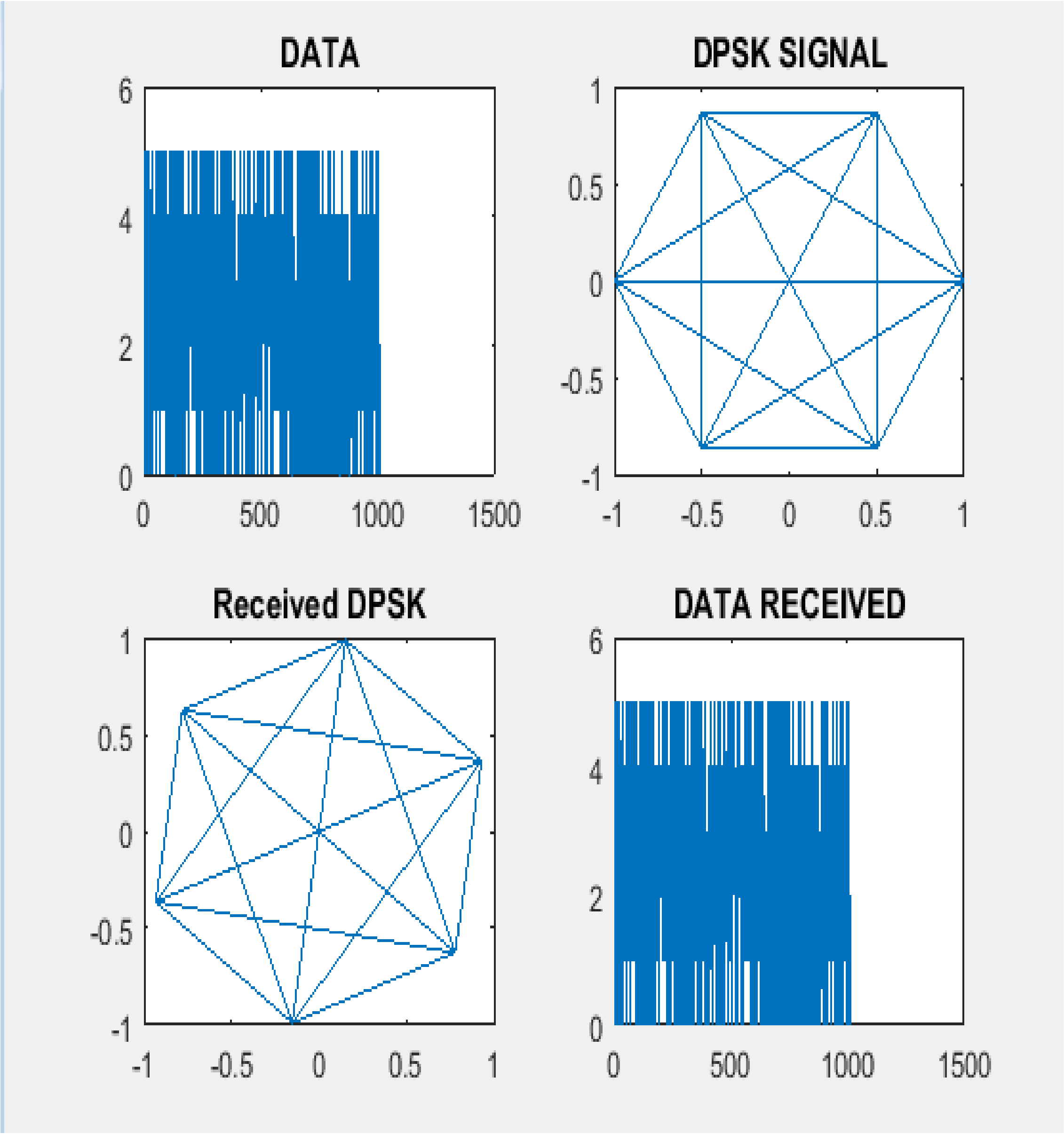
% **FSK Demodulation**

```
t1=0;t2=Tb
for i=1:N
t=[t1:(Tb/100):t2]
%correlator
x1=sum(c1.*fsk_sig1(i,:));
x2=sum(c2.*fsk_sig2(i,:));
x=x1-x2;
%decision device
if x>0
demod(i)=1;
else
demod(i)=0;
end
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end
%Plotting the demodulated data bits
subplot(3,2,6);stem(demod);
title(' demodulated data');xlabel('n--- >');ylabel('b(n)'); grid on;
```

Result

Thus the program for FSK modulation and demodulation has been simulated in MATLAB and necessary graphs are plotted.

SIMULATION WAVEFORM



11. SIMULATION OF DPSK, QPSK GENERATION SCHEMES

AIM:

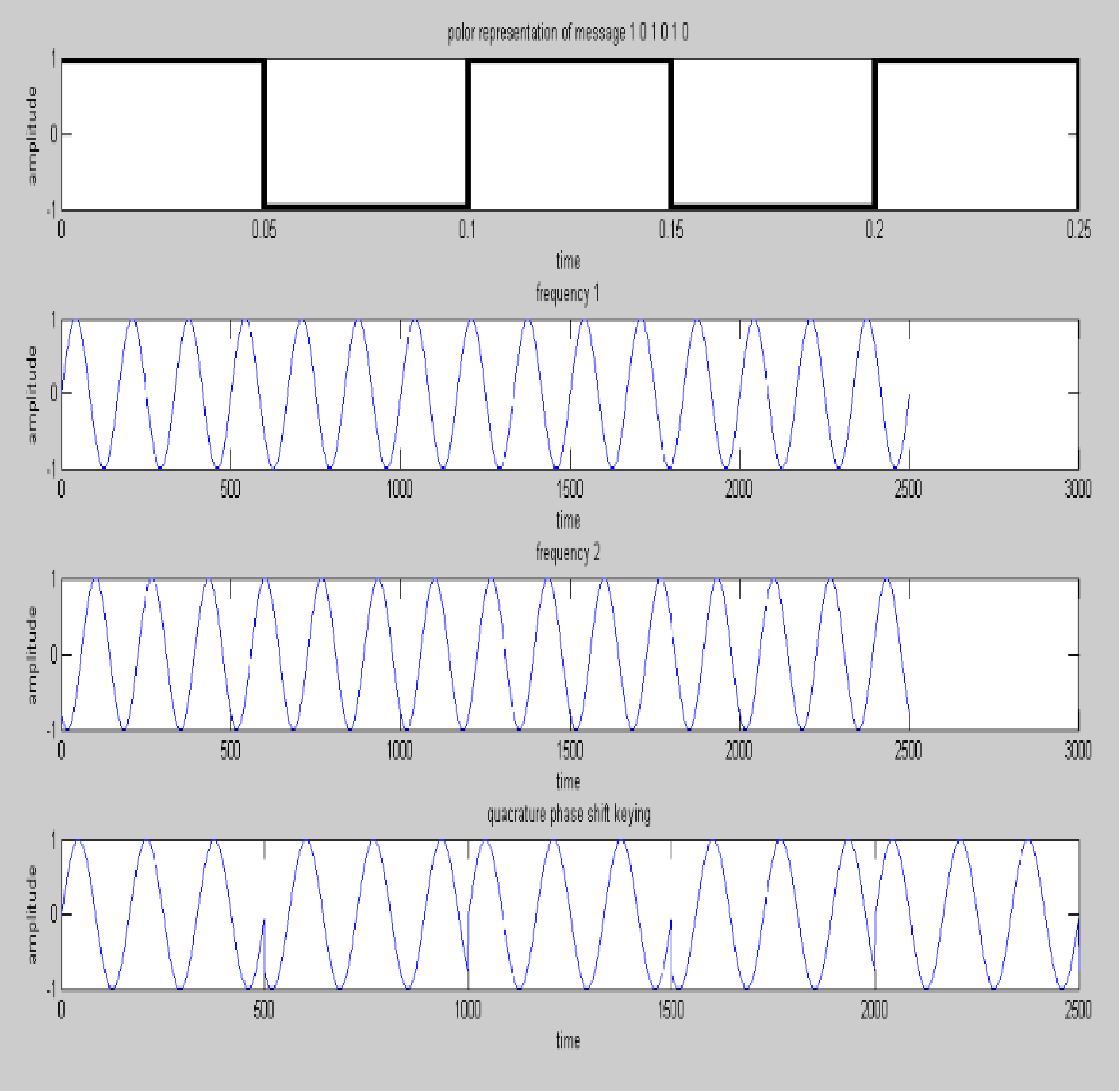
To simulate DPSK and QPSK Generation Schemes using MATLAB.

SOFTWARE REQUIRED: MATLAB

PROGRAM FOR DPSK GENERATION SCHEME:

```
clc;
clear all;
rng default
M = 6; % Alphabet size
dataIn = randi([0 M-1],1011,1); % Random message
txSig = dpskmod(dataIn,M); % Modulate
rxSig = txSig*exp(2i*pi*rand());
dataOut = dpskdemod(rxSig,M);
errs = symerr(dataIn,dataOut)
errs = symerr(dataIn(2:end),dataIn(2:end))
figure
subplot(2,2,1)
plot(dataIn)
title('DATA')
subplot(2,2,2)
plot(txSig)
title('DPSK SIGNAL')
subplot(2,2,3)
plot(rxSig)
title('Received DPSK')
subplot(2,2,4)
plot(dataOut)
title('DATA RECEIVED')
```

SIMULATION WAVEFORM



PROGRAM FOR QPSK GENERATION SCHEME:

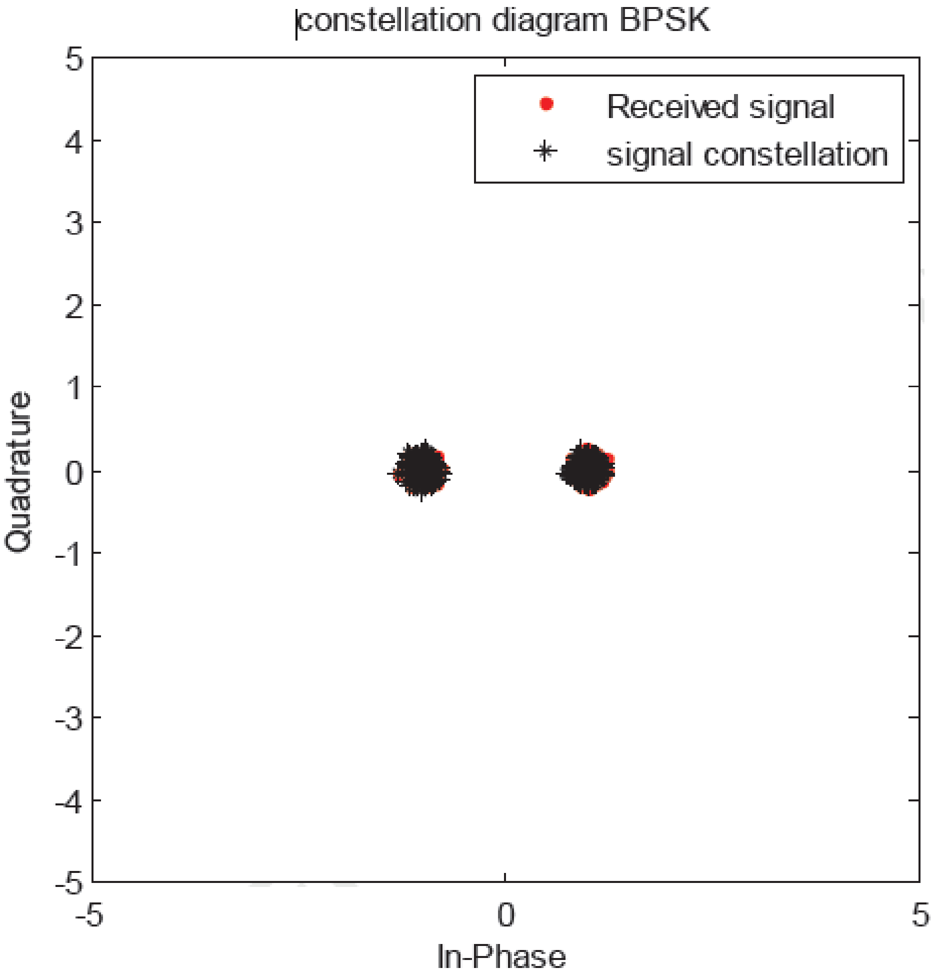
QPSK

```
clc; clear all;
t=0:0.0001:0.25;
m=square(2*pi*10*t);
c1=sin(2*pi*60*t);
c2=sin(2*pi*60*t+180);
for i=1:2500
    if(mod(i,1000))<500
      s(i)=c1(i);
    else
        s(i)=-c2(i);
    end
end
subplot(4,1,1);
plot(t,m,'k','linewidth',5);
title('polor representation of message 1 0 1 0 1 0');
xlabel('time'); ylabel('amplitude')
subplot(4,1,2); plot(c1);
title('frequency 1');
xlabel('time'); ylabel('amplitude');
subplot(4,1,3); plot(c2);
title('frequency 2');
xlabel('time'); ylabel('amplitude');
subplot(4,1,4); plot(s);
title('quadrature phase shift keying');
xlabel('time');
ylabel('amplitude');
```

RESULT:

Thus the DPSK and QPSK generation schemes were simulated using MATLAB.

BPSKCONSTELLATION:



12. OBSERVATION (SIMULATION) OF SIGNAL CONSTELLATIONS OF BPSK, QPSK AND QAM

Aim:

To plot the constellation diagram of digital modulation system BPSK, QPSK & QAM using MATLAB.

APPARATUS REQUIRED:

1. PC
2. MATLAB SOFTWARE

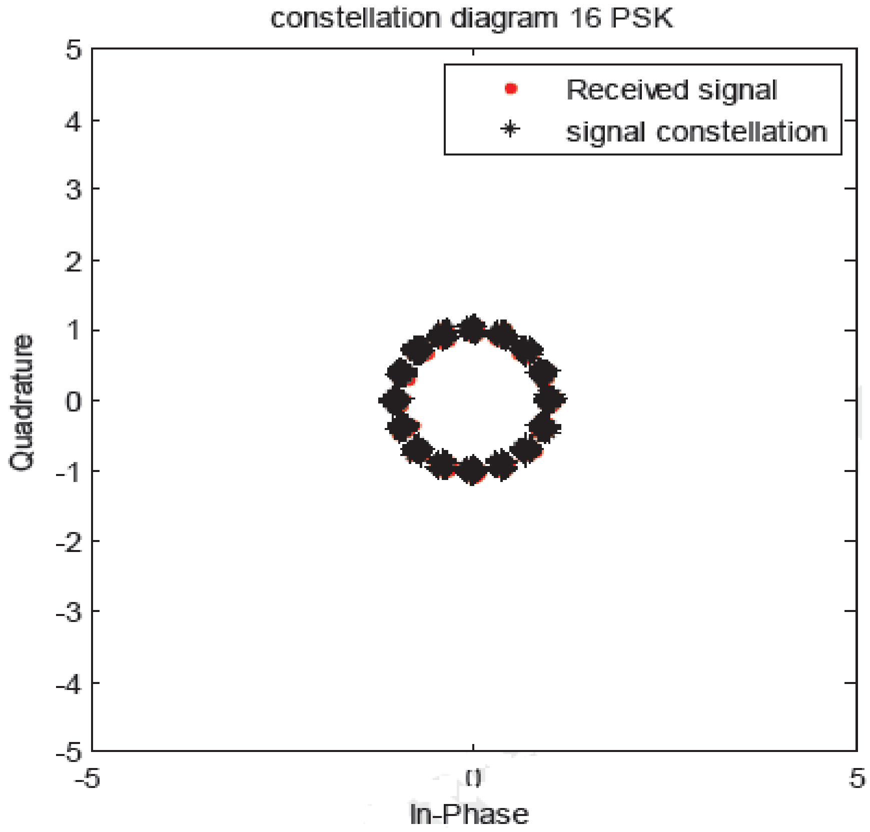
THEORY:

A constellation diagram is a representation of a signal modulated by an arbitrary digital modulation scheme. It displays the signal as a two dimensional scatter diagram in the complex plane at symbol sampling instants. It can also be viewed as the possible symbols that may be selected by a given modulation scheme as points in the complex plane.

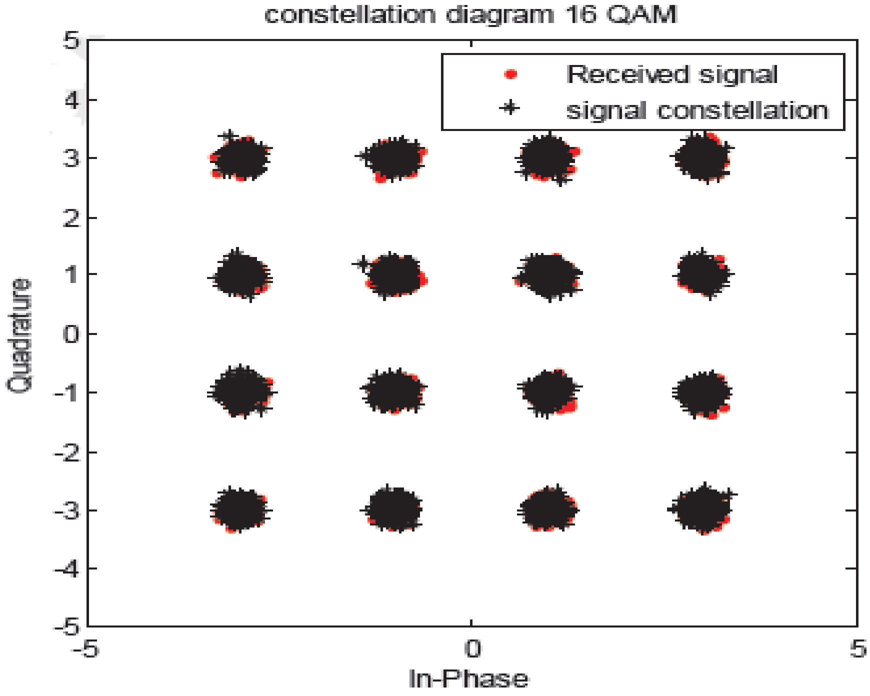
PROGRAM: BPSK

```
clc;
clear all;
close all;
M=2;
k=log2(M);
n=3*1e5;
nsamp=8;
X=randint(n,1);
xsym = bi2de(reshape(X,k,length(X)/k).','left-msb');
Y_psk= modulate(modem.pskmod(M),xsym);
Ytx_psk = Y_psk;
EbNo=30;
SNR=EbNo+10*log10(k)-10*log10(nsamp);
Ynoisy_psk = awgn(Ytx_psk,SNR,'measured');
Yrx_psk = Ynoisy_psk;
h1=scatterplot(Yrx_psk(1:nsamp*5e3),nsamp,0,'r.');
hold on;
scatterplot(Yrx_psk(1:5e3),1,0,'k*',h1);
title('constellation diagram BPSK');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
```

QPSKCONSTELLATION:



QAM CONSTELLATION:



Program for QPSK & QAM:

```
clc;
clear all;
close all;
M=16;
k=log2(M);
n=3*1e5;
nsamp=8;
X=randint(n,1);
xsym = bi2de(reshape(X,k,length(X)/k).','left-msb');
Y_qam= modulate(modem.qammod(M),xsym);
Y_qpsk= modulate(modem.pskmod(M),xsym);
Ytx_qam = Y_qam;
Ytx_qpsk = Y_qpsk;
EbNo=30;
SNR=EbNo+10*log10(k)-10*log10(nsamp);
Ynoisy_qam = awgn(Ytx_qam,SNR,'measured');
Ynoisy_qpsk = awgn(Ytx_qpsk,SNR,'measured');
Yrx_qam = Ynoisy_qam;
Yrx_qpsk = Ynoisy_qpsk;
h1=scatterplot(Yrx_qam(1:nsamp*5e3),nsamp,0,'r.');
hold on;
scatterplot(Yrx_qam(1:5e3),1,0,'k*',h1);
title('constellation diagram 16 QAM');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
h2=scatterplot(Yrx_qpsk(1:nsamp*5e3),nsamp,0,'r.');
hold on;
scatterplot(Yrx_qpsk(1:5e3),1,0,'k*',h2);
title('constellation diagram 16 PSK');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
hold off;
title('constellation diagram 16 PSK');
legend('Received signal' ,'signal constellation');
axis([-5 5 -5 5]);
holdoff;
```

RESULT:

Thus the constellation diagrams of digital modulation system BPSK, QPSK & QAM are simulated & plotted in MATLAB.

OUTPUT

```
Enter the Generator Matrix: [1 0 0 0 1 0 1; 0 1 0 0 1 1 1; 0 0 1 0 1 1 0; 0 0 0 1 0 1 1]
  g =

     1    0    0    0    1    0  1
     0    1    0    0    1    1  1
     0    0    1    0    1    1  0
     0    0    0    1    0    1  1

G =
The Order of Linear block Code for given Generator Matrix is:
n =7
k =4

The Possible Code words are:
  c =


     0    0    0    0    0    0  0
     0    0    0    1    0    1  1
     0    0    1    0    1    1  0
     0    0    1    1    1    0  1
     0    1    0    0    1    1  1
     0    1    0    1    1    0  0
     0    1    1    0    0    0  1
     0    1    1    1    0    1  0

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

13. SIMULATION OF LINEAR BLOCK CODING SCHEME

AIM:

To simulate and study the error control coding scheme of linear block code using MATLAB

ALGORITHM:

STEP 1: Give the generator matrix

STEP 2: Find the order of the linear block code for the given generator matrix

STEP 3: Obtain the possible code words

STEP 4: Find the minimum hamming distance

STEP 5: Give the received code word

STEP 6: Calculate the syndrome vector and compare it with transpose of hamming matrix.

STEP 7: Find the error bit position and display the corrected code word

MATLAB CODE:

```
% Input Generator Matrix
g=input('Enter The Generator Matrix: ')
disp ('G = ')
disp ('The Order of Linear block Code for given Generator Matrix is:')
[n,k] = size(transpose(g))
fori = 1:2^k
forj = k:-1:1
ifrem(i-1,2^(-j+k+1))>=2^(-j+k)
u(i,j)=1;
else
u(i,j)=0;
end
end
end
disp('The Possible Codewords are :')
c = rem(u*g,2)
disp('The Minimum Hamming Distance dmin for given Block Code is= ')
d_min = min(sum((c(2:2^k,:))'))
```

```
The Minimum Hamming Distance dmin for given Block Code is=

           d_min = 3

        Enter the Received Code Word:[1 0 0 0 1 0 0]
        r =
           1    0    0    0    1    0    0
        Hammimg Code
         ht =

             1    0   1
             1    1   1
             1    1   0
             0    1   1
             1    0   0
             0    1   0
             0    0   1
        Syndrome of a Given Code word is :
        s =
           0    0    1
        The Error is in bit:
           7
        The Corrected Code word is:
           1    0    0    0    1    0    1
```

```
 % Code Word
r         = input('Enter the Received Code Word:')
p          = [g(:,n-k+2:n)];

 h         = [transpose(p),eye(n-k)]; disp('HammimgCode')
 ht = transpose(h)
 disp('Syndrome of a Given Codeword is :')
                    s= rem(r*ht,2)
 fori = 1:1:size(ht)
 if(ht(i,1:3)==s)
 r(i) = 1-r(i);
 break;
 end
 end
 disp('The Error is in bit:')
 disp(i);
 disp('The Corrected Codeword is :')
 disp(r);
```

RESULT:

Thus the error control coding scheme of linear block code is studied using MATLAB.

OUTPUT

COMPUTATION OF CODE VECTORS FOR A CYCLIC CODE

```
Msg=
        1 0 0 1
        1 0 1 0
        1 0 1 1

Code =
        1 1 0 1 0 0 1
        0 1 1 1 0 1 0
        0 0 0 1 0 1 1
```

SYNDROME DECODING

```
Recd=
        1 0 1 1 1 1 0

Syndrome=7(decimal), 1 1 1(binary)

Parmat=
        1 0 0 1 0 1 1
        0 1 0 1 1 1 0
        0 0 1 0 1 1 1

Corrvect=
        0 0 0 0 0 1 0

Correctedcode=
        1 0 1 1 1 0 0
```

14. SIMULATION OF ERROR CONTROL USING CYCLICCODE

AIM:

a. To generate parity check matrix & generator matrix for a (7,4) Hamming code.

b. To generate parity check matrix given generator polynomial $g(x) = 1+x+x^3$.

c. To determine the code vectors.

d. To perform syndrome decoding

PROGRAM:

Generation of parity check matrix and generator matrix for a (7, 4) Hamming code.

```
[h,g,n,k] = hammgen(3);
```

Generation of parity check matrix for the generator polynomial $g(x) = 1+x+x^3$.

```
h1 = hammgen(3,[1011]);
```

Computation of code vectors for a cyclic code

```
clc;
close all;
n=7;
k=4;
msg=[1 0 0 1; 1 0 1 0; 1 0 1 1];
code = encode(msg,n,k,'cyclic');
msg
code
```

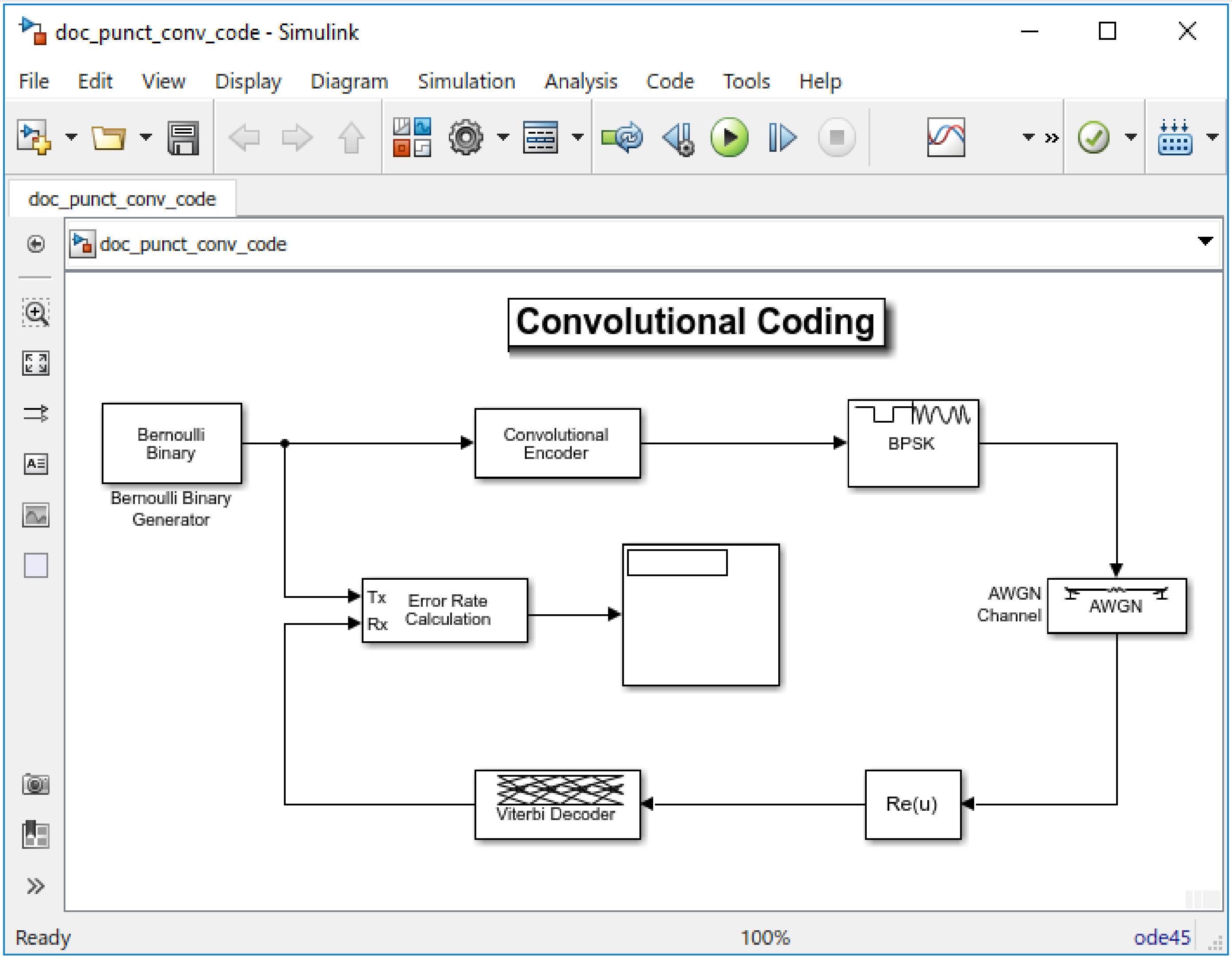
Syndrome decoding

```
clc;
close all;
q=3;
n=2^q-1;
k=n-q;
parmat = hammgen(q); % produce parity-check matrix
trt = syndtable(parmat); % produce decoding table
recd = [1 0 1 1 1 1 0 ] %received vector
syndrome = rem(recd * parmat',2);
syndrome_de = bi2de(syndrome, 'left-msb'); %convert to decimal
disp(['Syndrome = ',num2str(syndrome_de),.....
' (decimal), ',num2str(syndrome),' (binary) ']);
corrvect = trt(1+syndrome_de, :);%correction vector
correctedcode= rem(corrvect+recd,2);
parmat
corrvect
correctedcode
```

RESULT:

Thus encoding and decoding of block codes are performed using MATLAB.

CONVOLUTIONAL CODING



15. SIMULATION OF CONVOLUTIONAL CODING SCHEME

AIM

To simulate convolutional coding scheme using MATLAB Simulink tool.

APPARATUS REQUIRED:

PC with MATLAB Software

ALGORITHM:

Simulate the link by following these steps:

1. Generate binary data.
2. Encode the data with a rate 2/3 convolutional code.
3. Modulate the encoded data.
4. Pass the signal through an AWGN channel.
5. Demodulate the received signal.
6. Decode the demodulated signal by using a Viterbi decoder.
7. Collect the error statistics.

THEORY:

Generating Random Data

The Bernoulli Binary Generator block produces the information source for this simulation. The block generates a frame of three random bits at each sample time. The **Samples per frame** parameter determines the number of rows of the output frame.

Convolutional Encoding with Puncturing

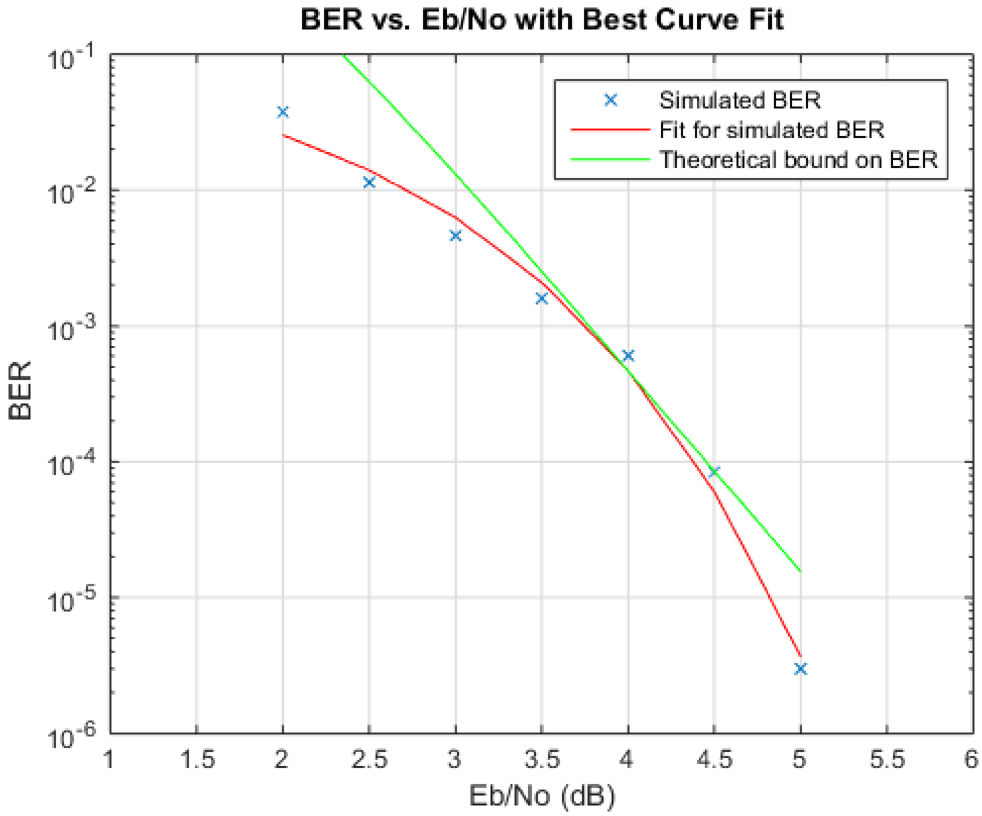
The Convolutional Encoder block encodes the data from the Bernoulli Binary Generator. This example uses the same code as described in Soft-Decision Decoding.

The puncture pattern is specified by the **Puncture vector** parameter in the mask. The puncture vector is a binary column vector. A 1 indicates that the bit in the corresponding position of the input vector is sent to the output vector, while a 0 indicates that the bit is removed.

For example, to create a rate 3/4 code from the rate 1/2, constraint length 7 convolutional code, the optimal puncture vector is [1 1 0 1 1 0].' (where the .' after the vector indicates the transpose). Bits in positions 1, 2, 4, and 5 are transmitted, while bits in positions 3 and 6 are removed. Now, for every 3 bits of input, the punctured code generates 4 bits of output (as opposed to the 6 bits produced before puncturing). This makes the rate 3/4.

In this example, the output from the Bernoulli Binary Generator is a column vector of length 3. Because the rate 1/2 Convolutional Encoder doubles the length of each vector, the length of the puncture vector must divide 6.

BIT ERROR RATE (BER)



Transmitting Data

The AWGN Channel block simulates transmission over a noisy channel. The parameters for the block are set in the mask as follows:

- The **Mode** parameter for this block is set to Signal to noise ratio (Es/No).
- The **Es/No** parameter is set to 2 dB. This value typically is changed from one simulation run to the next.
- The preceding modulation block generates unit power signals so the **Input signal power** is set to 1 Watt.
- The **Symbol period** is set to 0.75 seconds because the code has rate 3/4.

Demodulating

In this simulation, the Viterbi Decoder block is set to accept unquantized inputs. As a result, the simulation passes the channel output through a Simulink® Complex to Real-Image block that extracts the real part of the complex samples.

Viterbi Decoding of Punctured Codes

The Viterbi Decoder block is configured to decode the same rate 1/2 code specified in the Convolutional Encoder block.

In this example, the decision type is set to Unquantized. For codes without puncturing, you would normally set the **Traceback depth** for this code to a value close to 40. However, for decoding punctured codes, a higher value is required to give the decoder enough data to resolve the ambiguities introduced by the punctures.Since the punctured bits are not transmitted, there is no information to indicate their values. As a result they are ignored in the decoding process.

The **Puncture vector** parameter indicates the locations of the punctures or the bits to ignore in the decoding process. Each 1 in the puncture vector indicates a transmitted bit while each 0 indicates a puncture or the bit to ignore in the input to the decoder.In general, the two **Puncture vector** parameters in the Convolutional Encoder and Viterbi Decoder must be the same.

Calculating the Error Rate

The Error Rate Calculation block compares the decoded bits to the original source bits. The output of the Error Rate Calculation block is a three-element vector containing the calculated bit error rate (BER), the number of errors observed, and the number of bits processed.

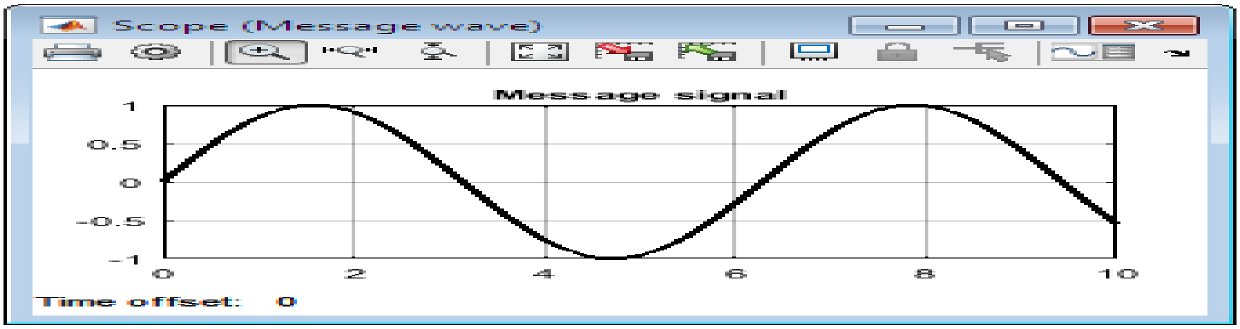
In the mask for this block, the **Receive delay** parameter is set to 96, because the **Traceback depth** value of 96 in the Viterbi Decoder block creates a delay of 96. If there were other blocks in the model that created delays, the **Receive delay** would equal the sum of all the delays.

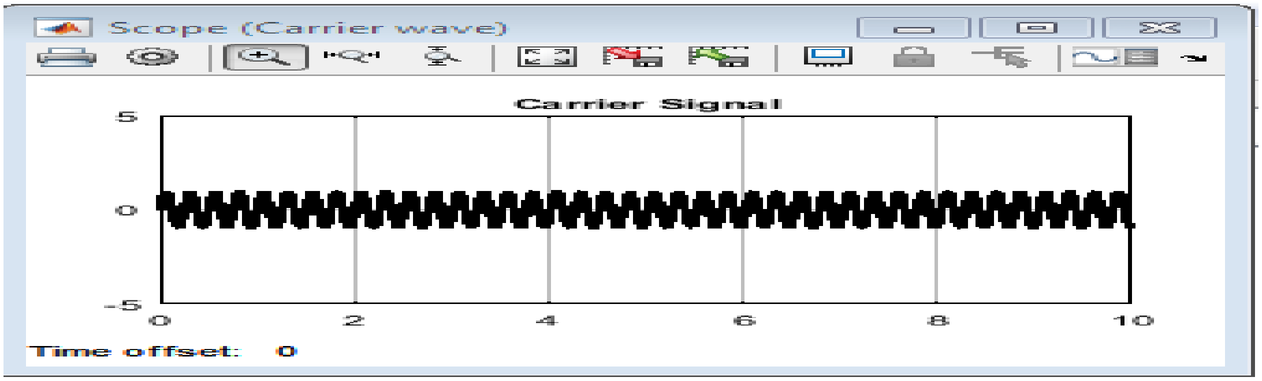
BER simulations typically run until a minimum number of errors have occurred, or until the simulation processes a maximum number of bits. The Error Rate Calculation block uses its **Stop simulation** mode to set these limits and to control the duration of the simulation.

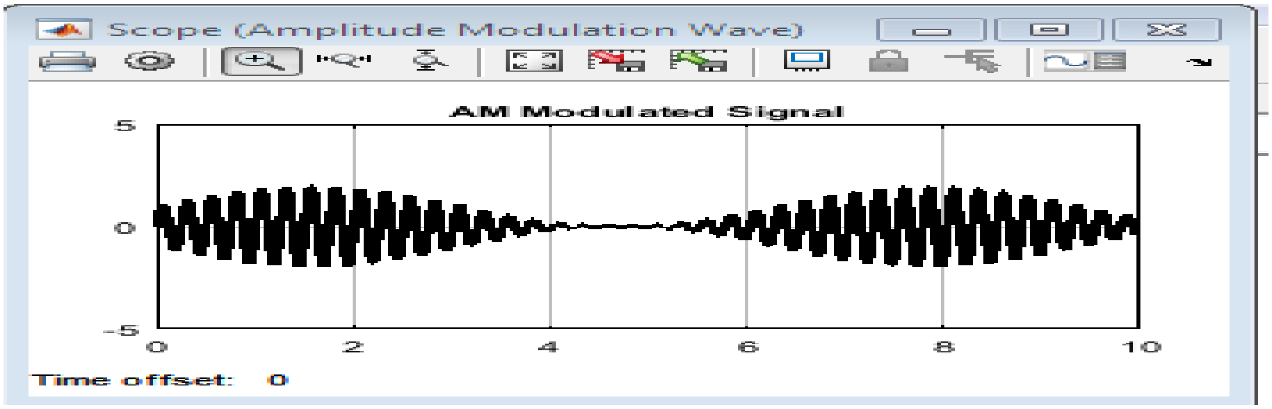
RESULT:

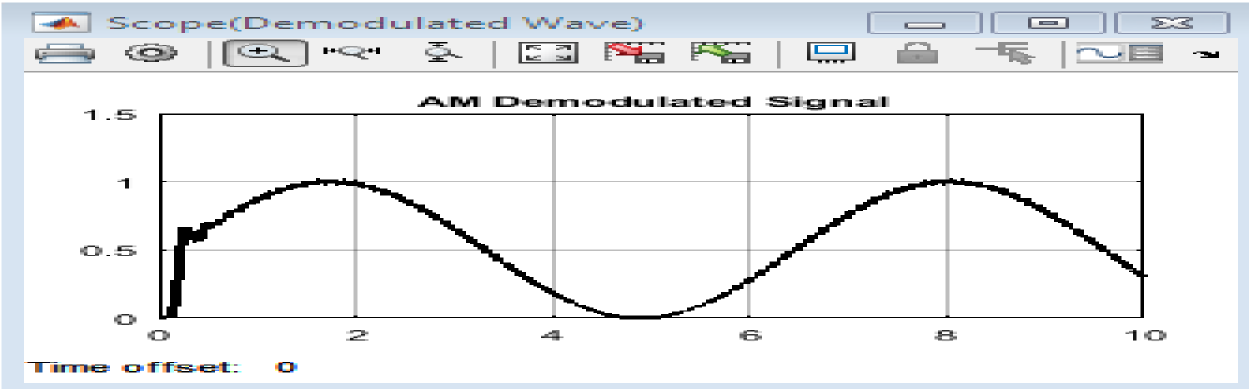
Thus the convolution coding scheme was simulated using MATLAB Simulink tool.

OUTPUT (AM)


Scope (Message wave)
Message signal
1
0.5
0
-0.5
-1
0
2
4
6
8
10
Time offset: 0


Scope (Carrier wave)
Carrier Signal
5
0
-5
0
2
4
6
8
10
Time offset: 0


Scope (Amplitude Modulation Wave)
AM Modulated Signal
5
0
-5
0
2
4
6
8
10
Time offset: 0


Scope(Demodulated Wave)
AM Demodulated Signal
1.5
1
0.5
0
0
2
4
6
8
10
Time offset: 0

16.COMMUNICATION LINK SIMULATIONS

AIM:

To generate modulation and demodulation of AM, ASK, DM and PAM using Simulink.

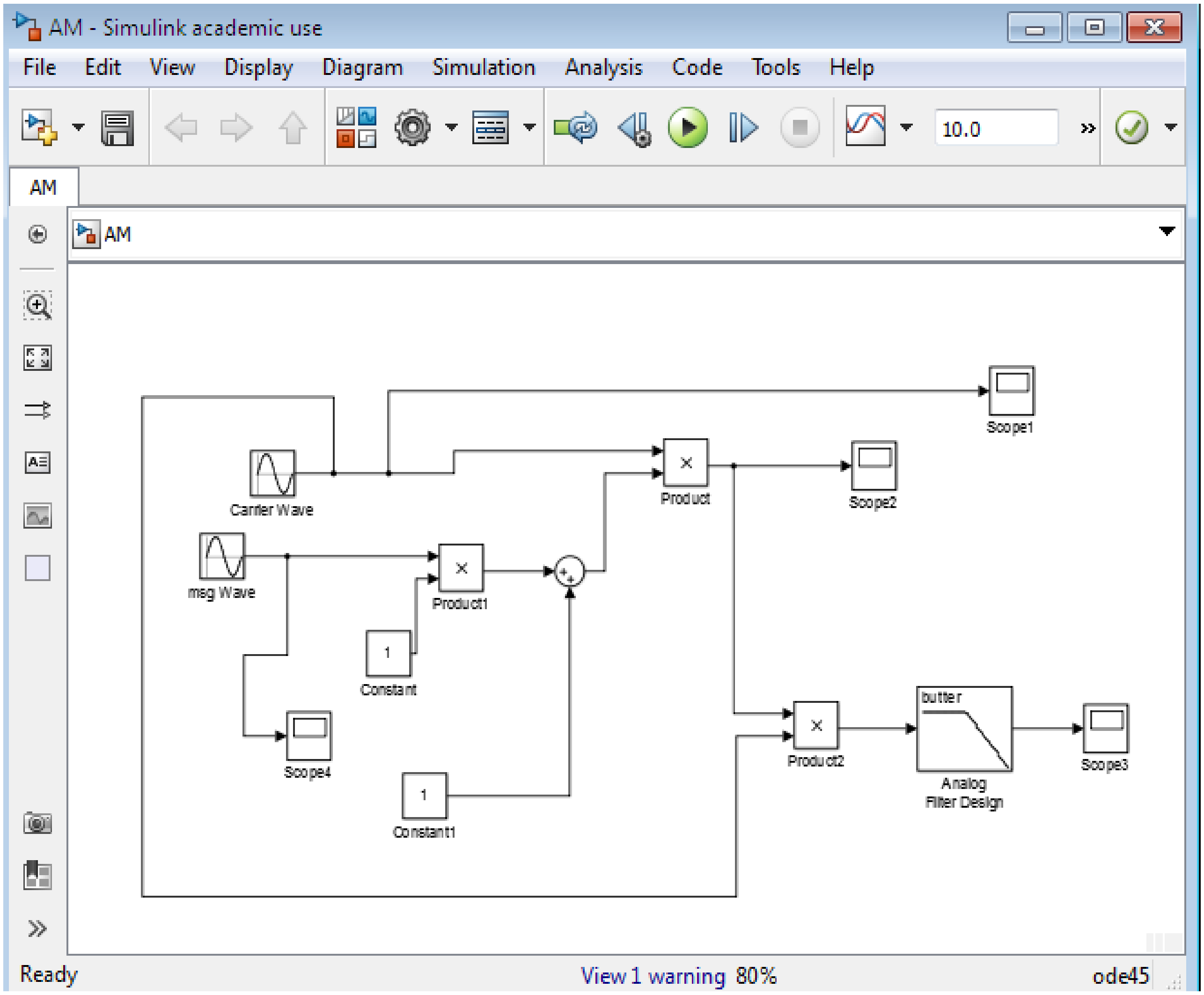
APPARATUS REQUIRED:

PC with MATLAB Software with Simulink tool

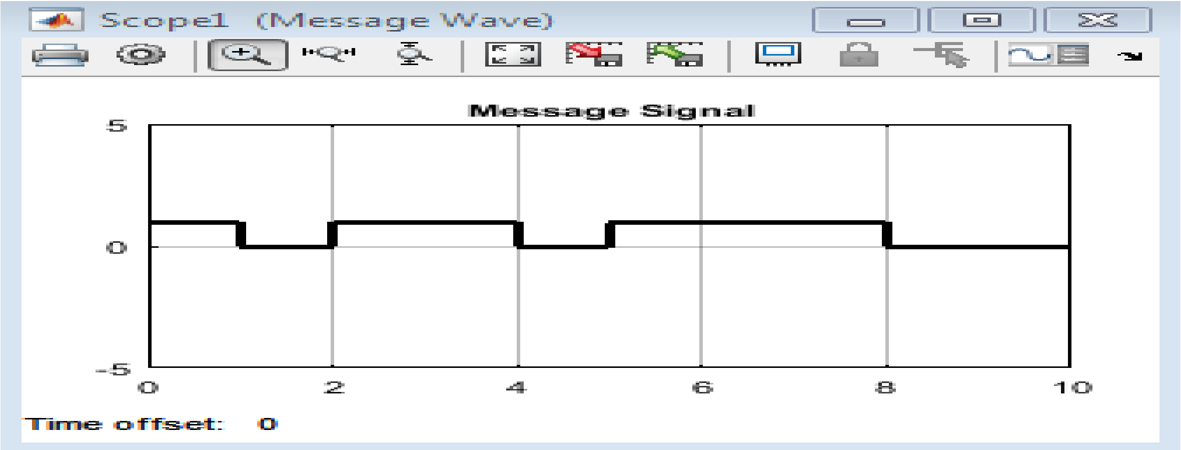
PROCEDURE:

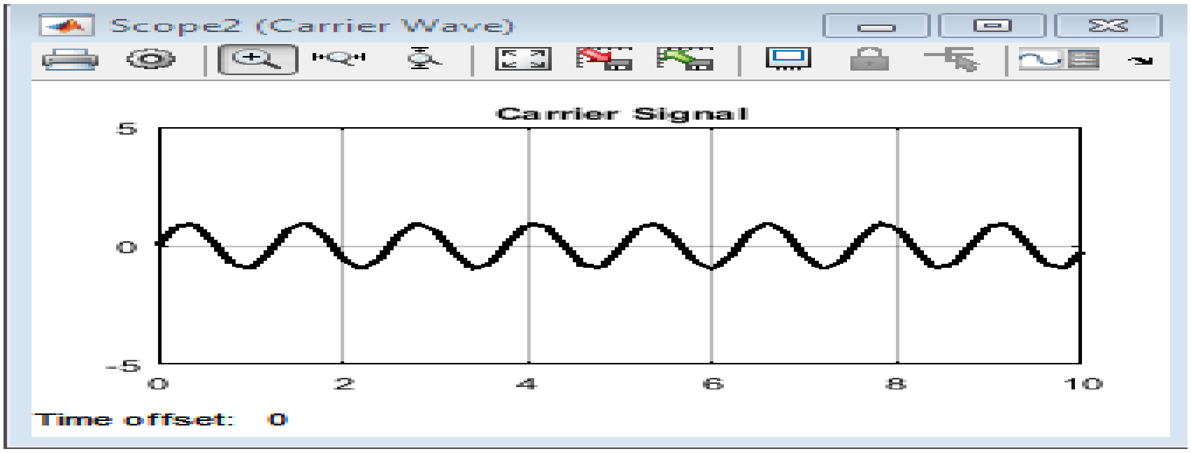
1. Start simulink section
2. Select file → New Model in the simulink library to construct a new model
3. Go to simulink library select appropriate module and add to model
4. Connect all the inserted models
5. Set the simulation parameters
6. Run the simulation and observe and save all the plots and values.

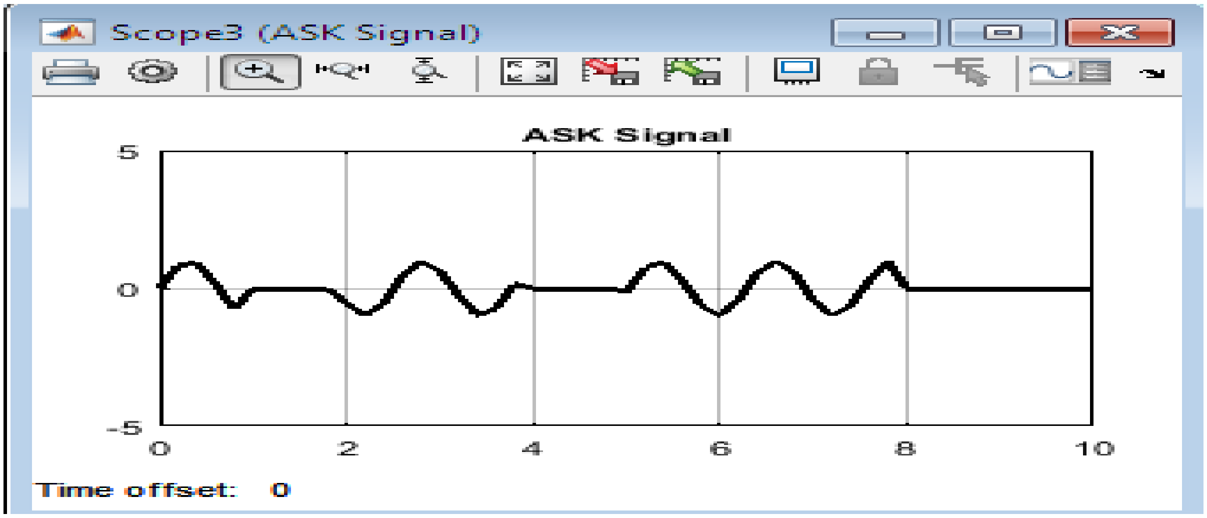
AM MODULE



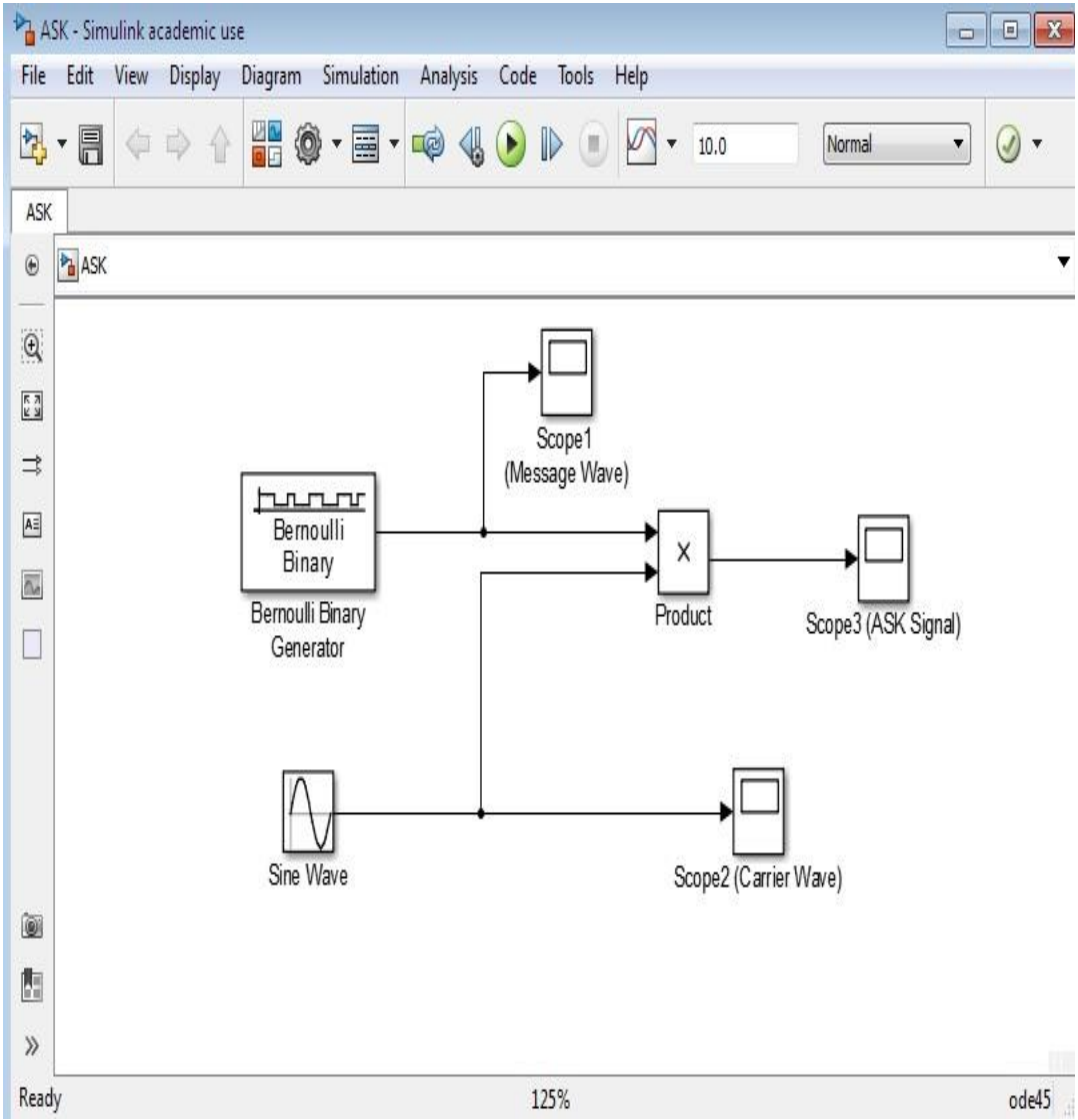
OUTPUT


Scope1 (Message Wave)
Message Signal
5
0
-5
0
2
4
6
8
10
Time offset: 0

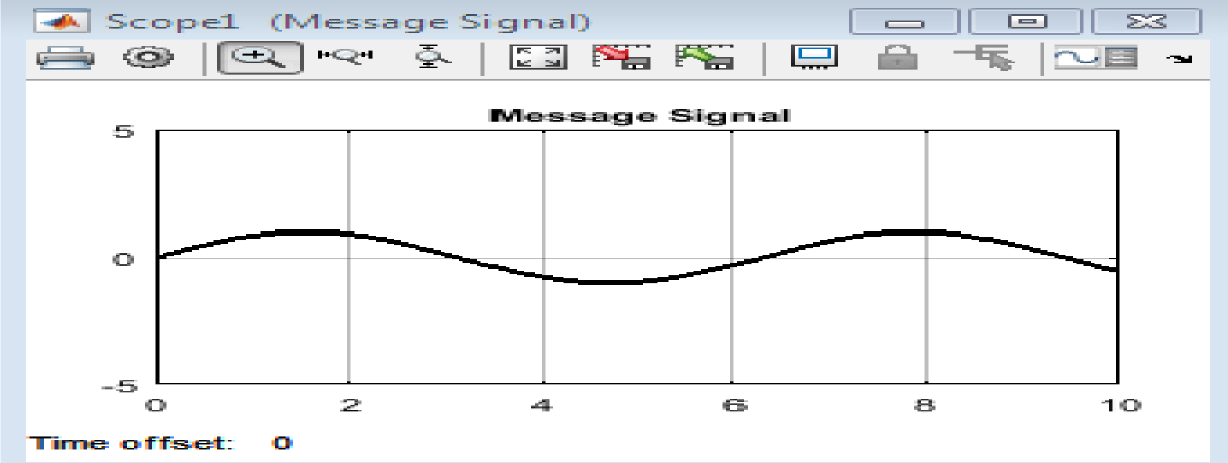

Scope2 (Carrier Wave)
Carrier Signal
5
0
-5
0
2
4
6
8
10
Time offset: 0

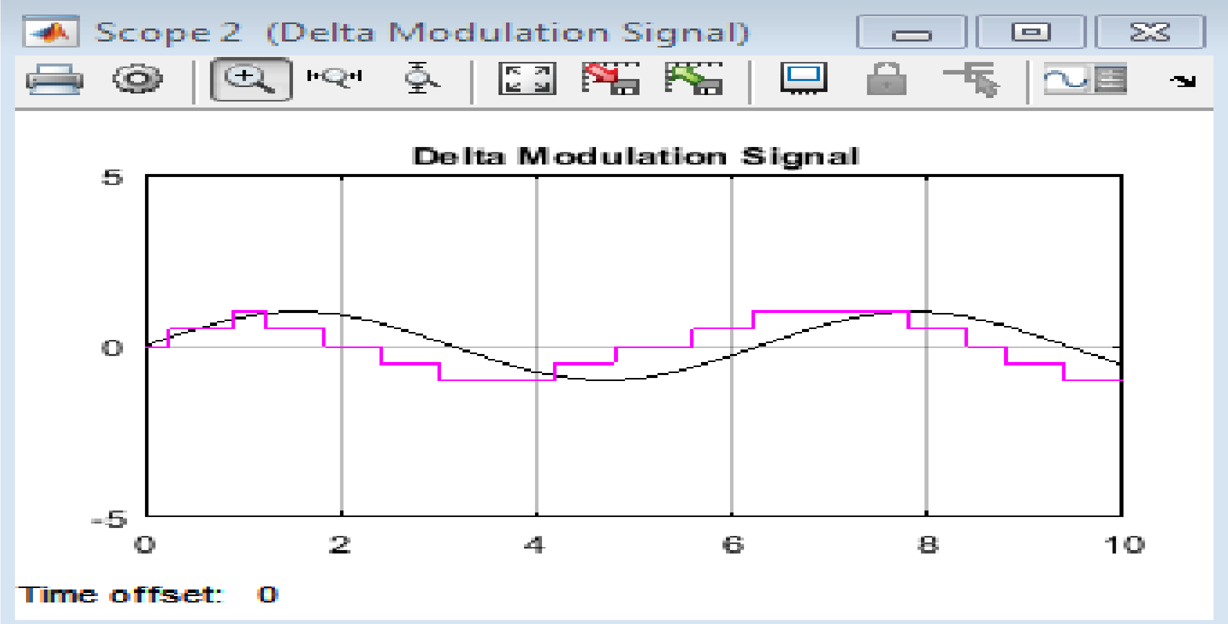

Scope3 (ASK Signal)
ASK Signal
5
0
-5
0
2
4
6
8
10
Time offset: 0

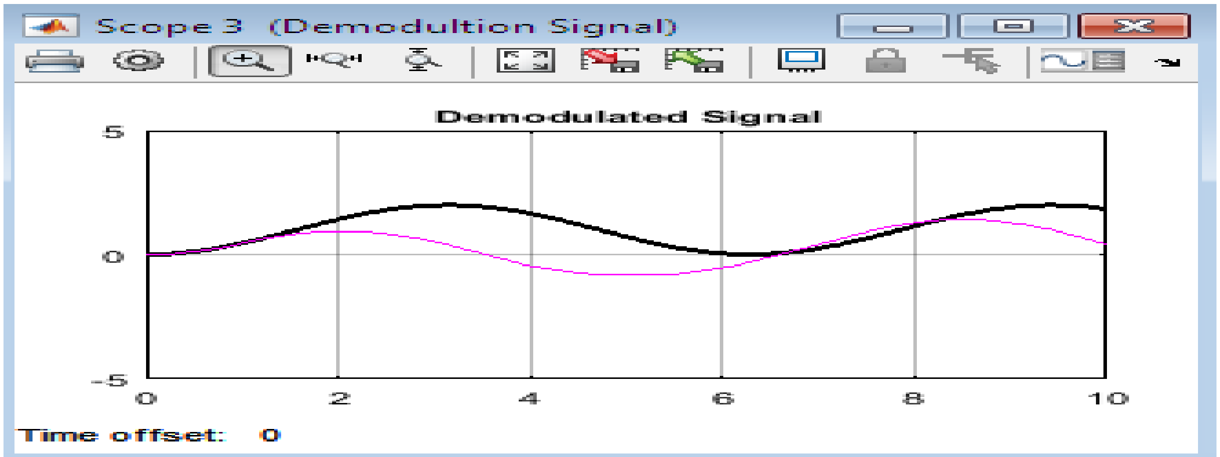
ASK MODULE


ASK - Simulink academic use
File
Edit
View
Display
Diagram
Simulation
Analysis
Code
Tools
Help
10.0
Normal
ASK
ASK
Scope1
(Message Wave)
Bernoulli
Binary
Bernoulli Binary
Generator
Product
Scope3 (ASK Signal)
Sine Wave
Scope2 (Carrier Wave)
Ready
125%
ode45

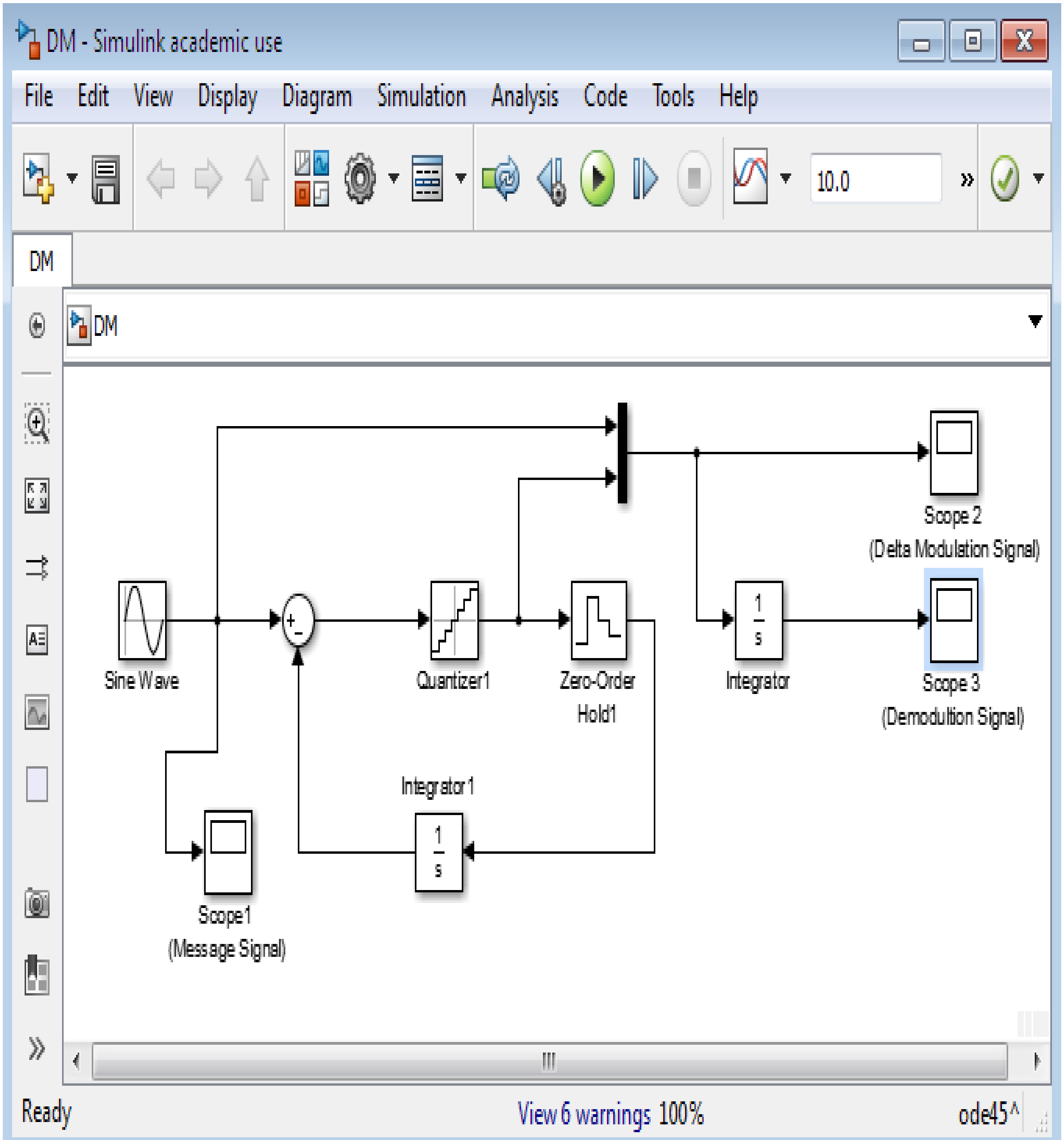
SIMULATION OUTPUT



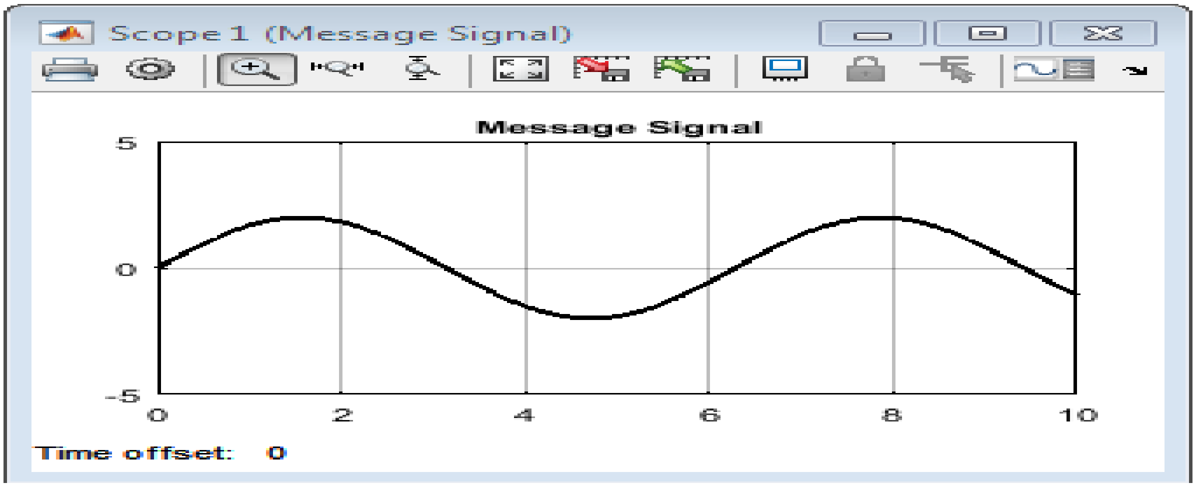


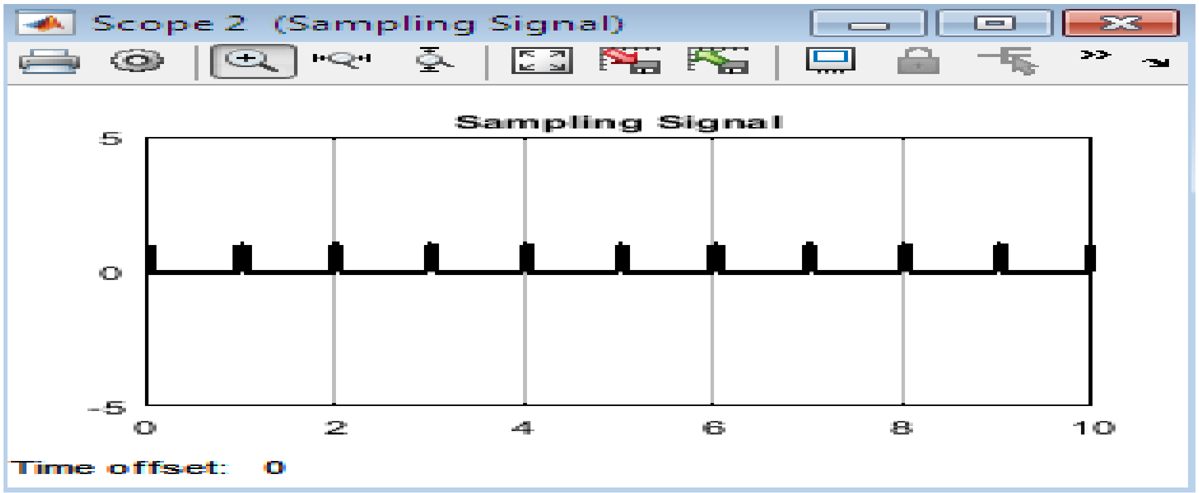


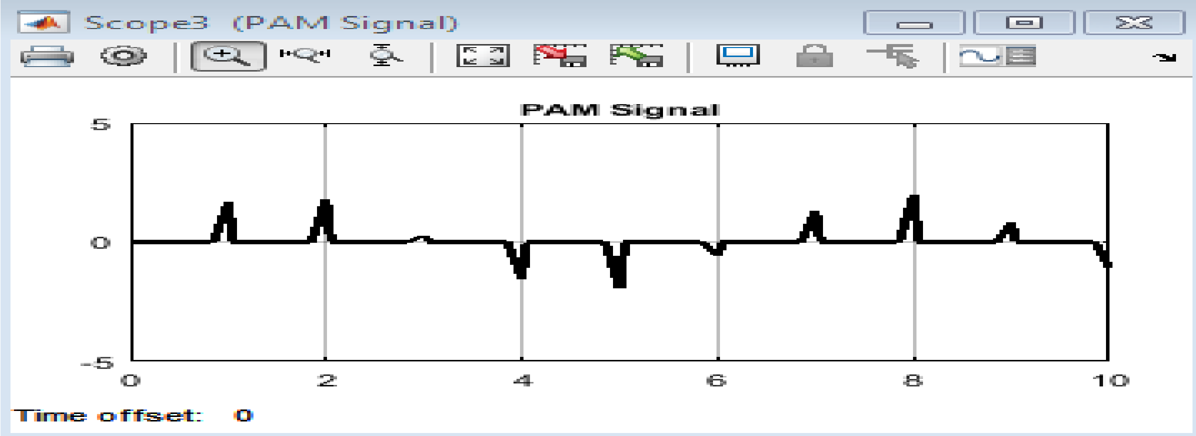
DM MODULE


DM - Simulink academic use
File
Edit
View
Display
Diagram
Simulation
Analysis
Code
Tools
Help
10.0
DM
DM
Sine Wave
+
-
Quantizer1
Zero-Order Hold1
Integrator1
1
s
Integrator
1
s
Scope1
(Message Signal)
Scope 2
(Delta Modulation Signal)
Scope 3
(Demodultion Signal)
Ready
View 6 warnings
100%
ode45

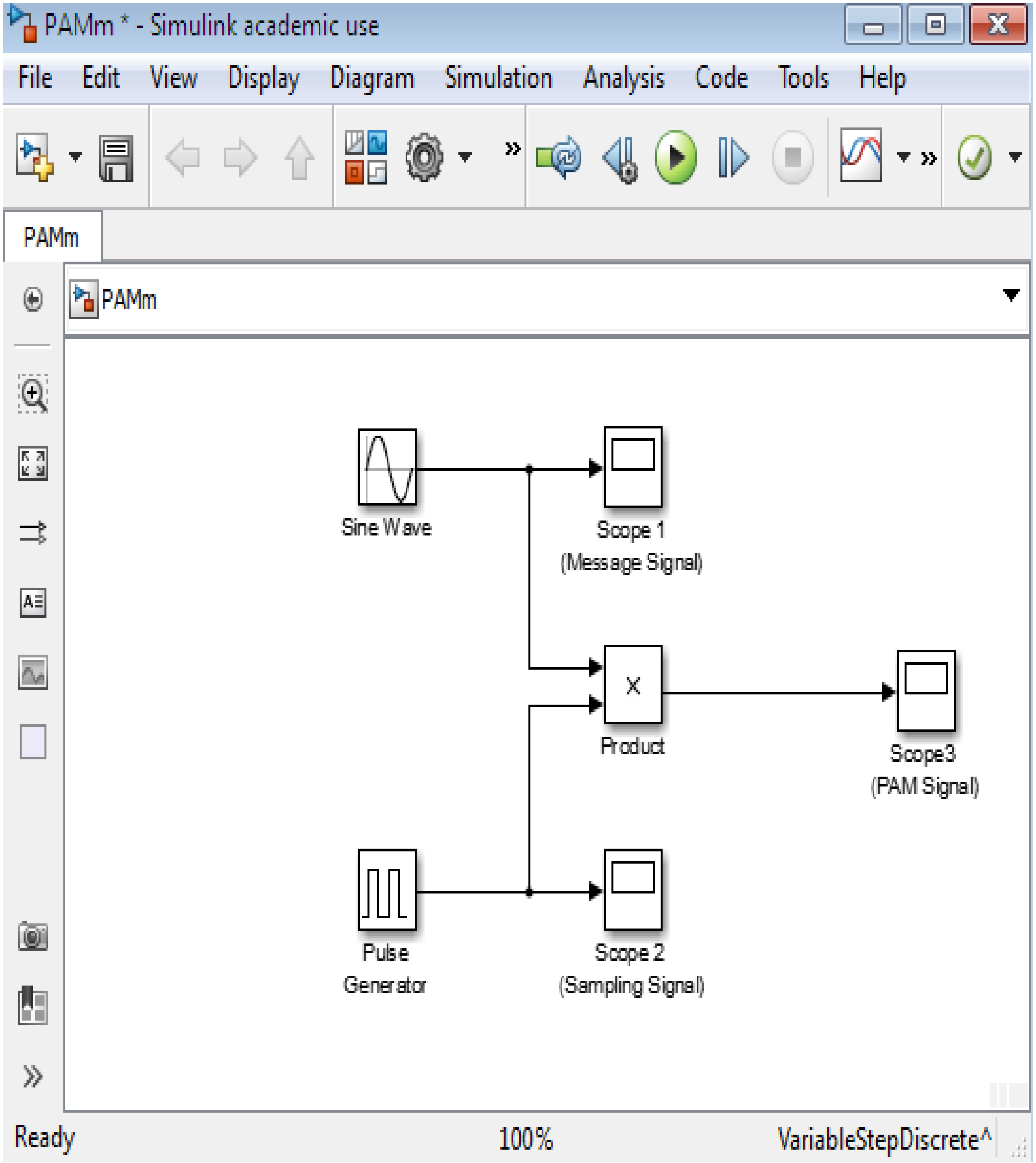
SIMULATION OUTPUT







PAM MODULE

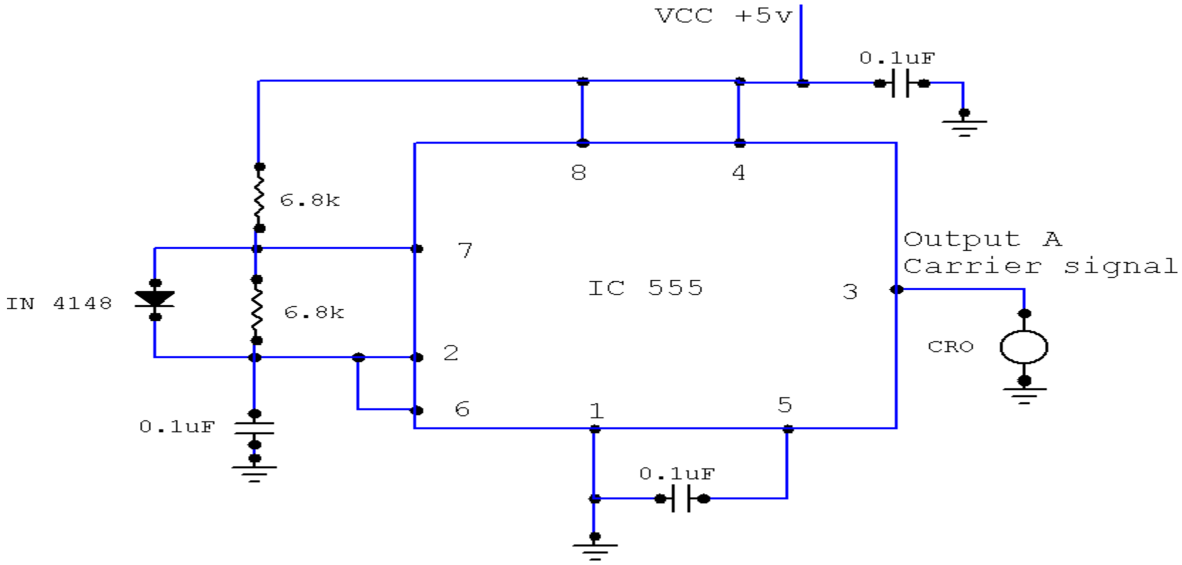


RESULT:

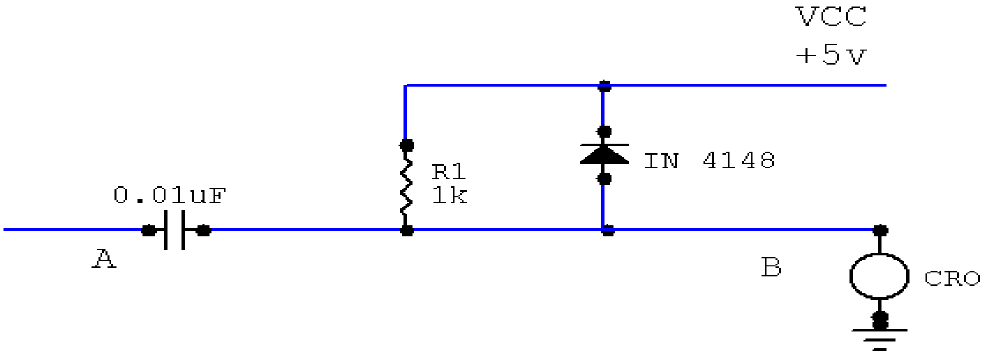
Thus the basic principle of AM, ASK, DM and PAM were studied using simulink.

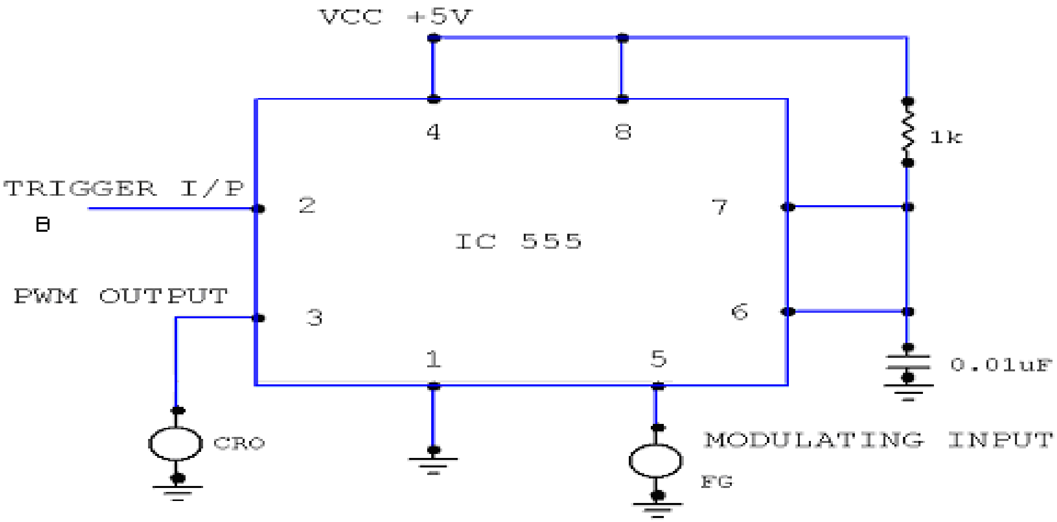
CONTENT BEYOND THE SYLLABUS

CIRCUIT DIAGRAM:



TRIGGER CIRCUIT





PULSE WIDTH MODULATION

AIM:

To generate the Pulse Width Modulated signal using 555 timer.

COMPONENTS REQUIRED:

S.NO	Name of the Component	Range	Quantity
1	IC 555	-	
2	Diode	IN4001	
3	Capacitors	0.1µF, 0.01 µF	3,2
4	Resistors	6.8K, 10K, 1.8K	2,1,1
5	Function Generator	1MHz	1
6	CRO	20MHz	1
7	Bread board	-	1
8	Regulated power supply	0-15 V	1

DESIGN:

Assume carrier frequency *fc* = 750 Hz.

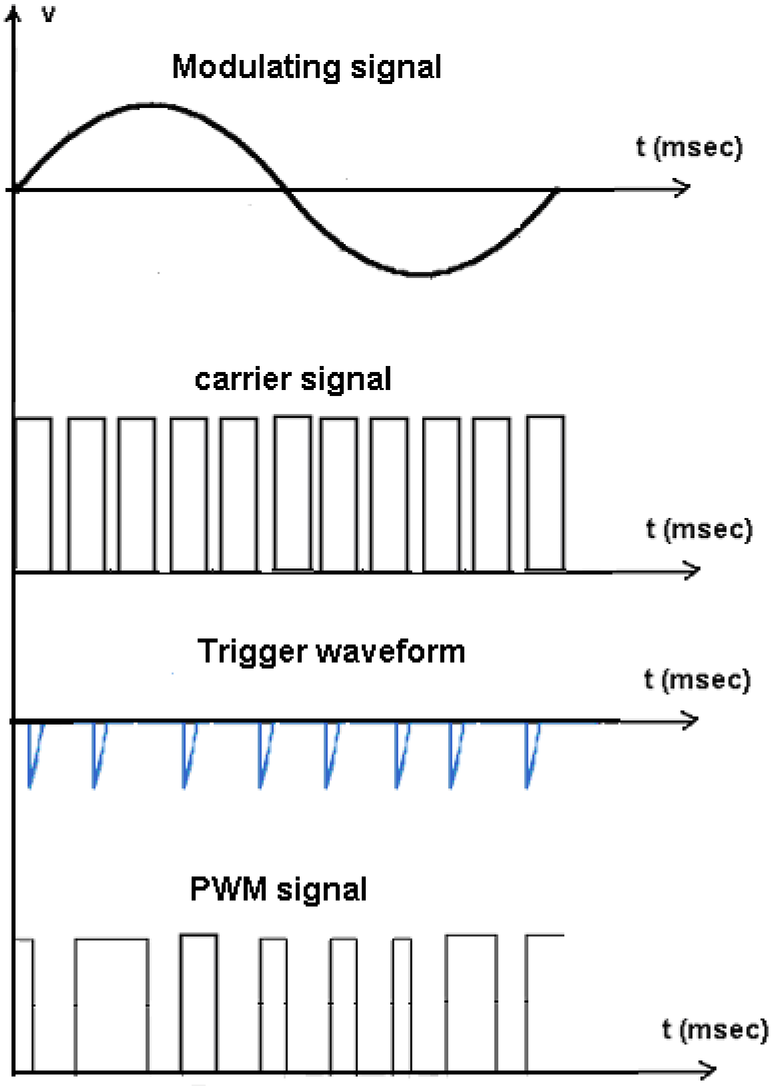
The operating frequency of IC 555 timer is given by $f_0 = \frac{1.45}{(RA+2\ RB)C}$

Let C = 0.1 µ*f, Let* RA=*RB* , Substituting we get, R_A =6.4K ;R_A =6.8

THEORY:

Pulse Time Modulation is also known as Pulse Width Modulation or Pulse Length Modulation. In PWM, the samples of the message signal are used to vary the duration of the individual pulses. Width may be varied by varying the time of occurrence of leading edge, the trailing edge or both edges of the pulse in accordance with modulating wave. It is also called Pulse Duration Modulation. Pulse width modulation is a one in which each pulse has fixed amplitude but width of the pulses is made proportional to amplitude of the modulating signal at that instant.

MODEL GRAPH:



TABULATION:

Parameter	Amplitude in Volts	Time period in ms
Modulating Signal		
Carrier signal		**Ton =** **Toff =**
Trigger input		**Positive spike =** **Negative spike =**

Modulated signal		**Time period of each pulse**

Pulse width increase when signal amplitude increases in positive direction and decreases when signal amplitude increases in negative direction. Pulses of PWM is of varying pulse width and hence of varying power component. So transmitter should be powerful enough to handle the power of maximum pulse width. But average power transmitted is only half is peak power. The main advantage of PWM is system will work even if the synchronization between the transmitter and receiver fails. The emitter coupled monostable multivibrator is an excellent voltage to time converter. Since its capacitor charges if the voltage is varied in accordance with the signal voltage, a series of rectangular pulses will be obtained with varying width as required.

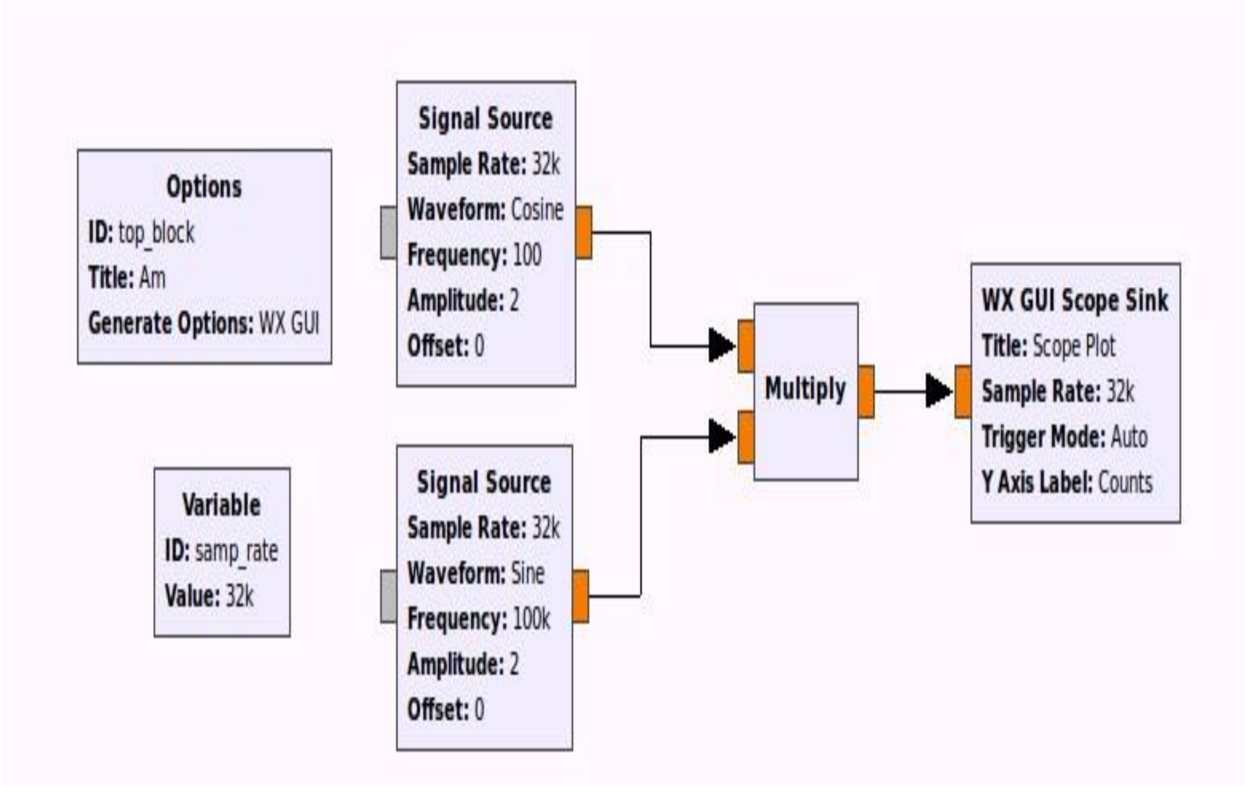
PROCEDURE:

1. Rig up the circuit as shown in the circuit diagram.
2. Note down the amplitude (V_m) and time period of the modulating signal.
3. Observe the output at „A‟ (carrier signal) and measure the amplitude (V_c) and time period.
4. Observe the spike output at „B‟ and measure the amplitude and time period.
5. Apply the modulating signal input and trigger input and observe the PWM output.
6. Note down the amplitude and time period of all the signals and plot them on the graph.

RESULT:

Thus the Pulse Width Modulated signal is generated using IC 555 timer and its Waveforms are plotted.

BLOCK DIAGRAM:



Options
ID: top_block
Title: Am
Generate Options: WX GUI

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 32k
Trigger Mode: Auto
Y Axis Label: Counts

AMPLITUDE MODULATOR USING SOFTWARE DEFINED RADIO

AIM:

To construct an Amplitude modulator and demodulator and using SDR TRAINER KIT.

EQUIPMENTS REQUIRED:

SDR Trainer Kit -1

SMA Connector-1

USB device -1

THEORY:

INTRODUCTION TO SDR KIT

What is Software defined radio (SDR)?

Software defined radio is defined as an environment where Hardware and Software are different parts allows user to implement the operating functions of hardware through a modifiable software. Complete design produces a radio which can receive and transmit widely different radio protocols (sometimes referred to as waveforms) based solely on the software used.

Where SDR can be used?

Due to its wide RF range it covers a wide range of applications including high frequency communications, FM and TV broadcast, cellular, Wi-Fi, ISM, and lot more. Starting from simple experiments, it makes you grow in experience and complexity up to being able to deal with competence and master the fundamental elements which makes the Software Based Radio.

FEATURES

- RFCoverage from70MHz – 6 GHz RF
- GNU Radio and open BTS support through the open source USRP Hardware Driver
- USB 3.0 High speed interface (Compatible with USB 2.0)
- Flexible rate 12 bit ADC/DAC
- 1TX, 1 RX, Half or Full Duplex
- Xilinx Spartan 6 XC6SLX75 FPGA
- Up to 56 MHz of real-time bandwidth

Power

- DC Input: 6V

SOFTWARE DESCRIPTION

What is GNU Radio?

GNU Radio is a ***software library***, which can be used to develop complete applications for radio engineering and signal processing.

Introduction

GNU Radio is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment.

GNU Radio is licensed under the GNU General Public License (GPL) version 3. All of the code is copyright of the Free Software Foundation. While all the applications are implemented using python language while critical signal processing path is done using C++ language.

PROCEDURE TO WORK ON GNU Radio Companion:

GNU Radio Companion (GRC) is a graphical user interface that allows you to build GNU Radio

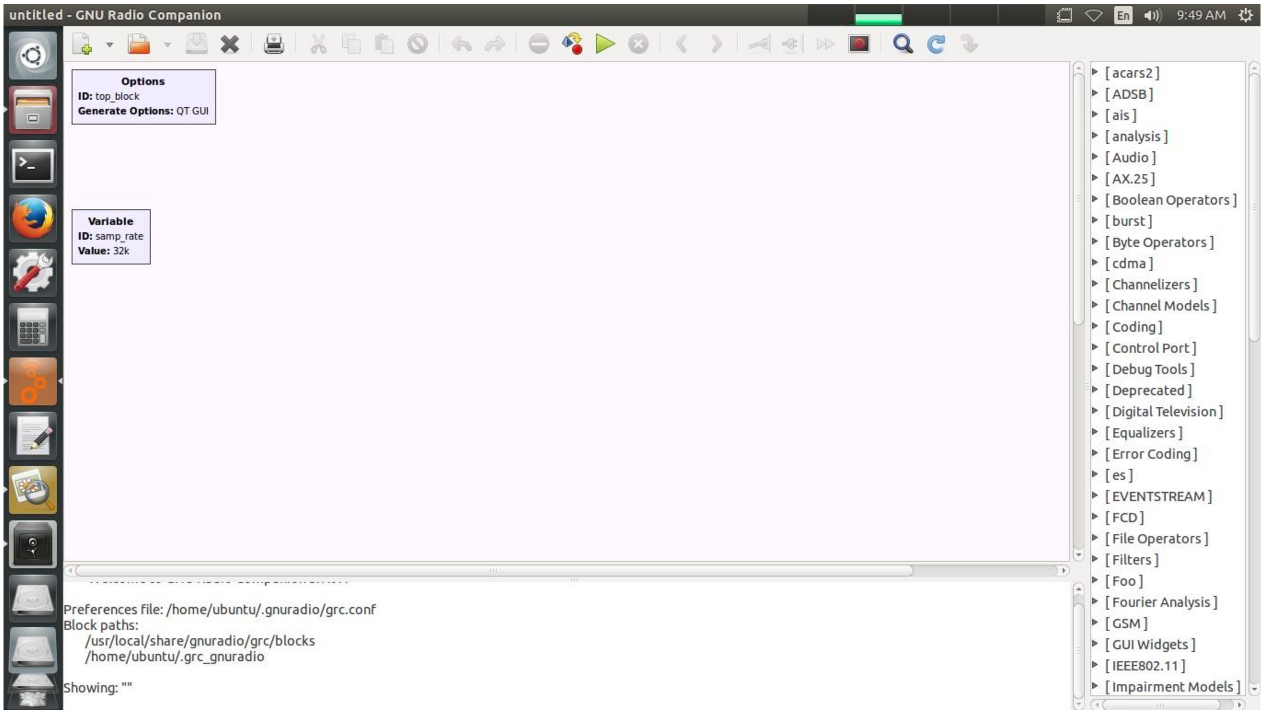
flow graphs. It is an excellent way to learn the basics of GNU Radio. This is the first in a series of tutorials that will introduce you to the use of GRC.

AMPLITUDE MODULATION

Amplitude modulation is the process of changing the amplitude of a relatively high frequency carrier signal in proportion with the instantaneous value of the modulating signal.

Procedure for Amplitude modulation:

STEP 1: Click on GRC.


untitled - GNU Radio Companion

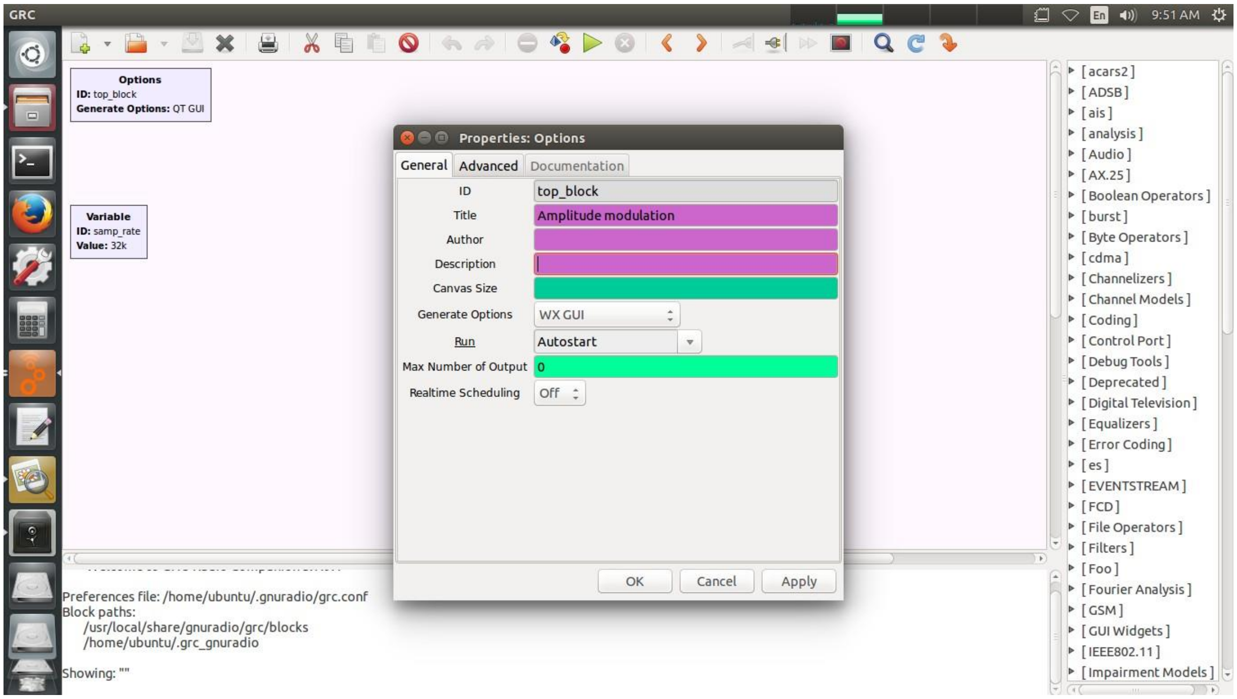
Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

```
Preferences file: /home/ubuntu/.gnuradio/grc.conf
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""
```

STEP 2: Click on options and name the title and change generate options as WX GUI.


GRC

Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

Properties: Options

General | Advanced | Documentation

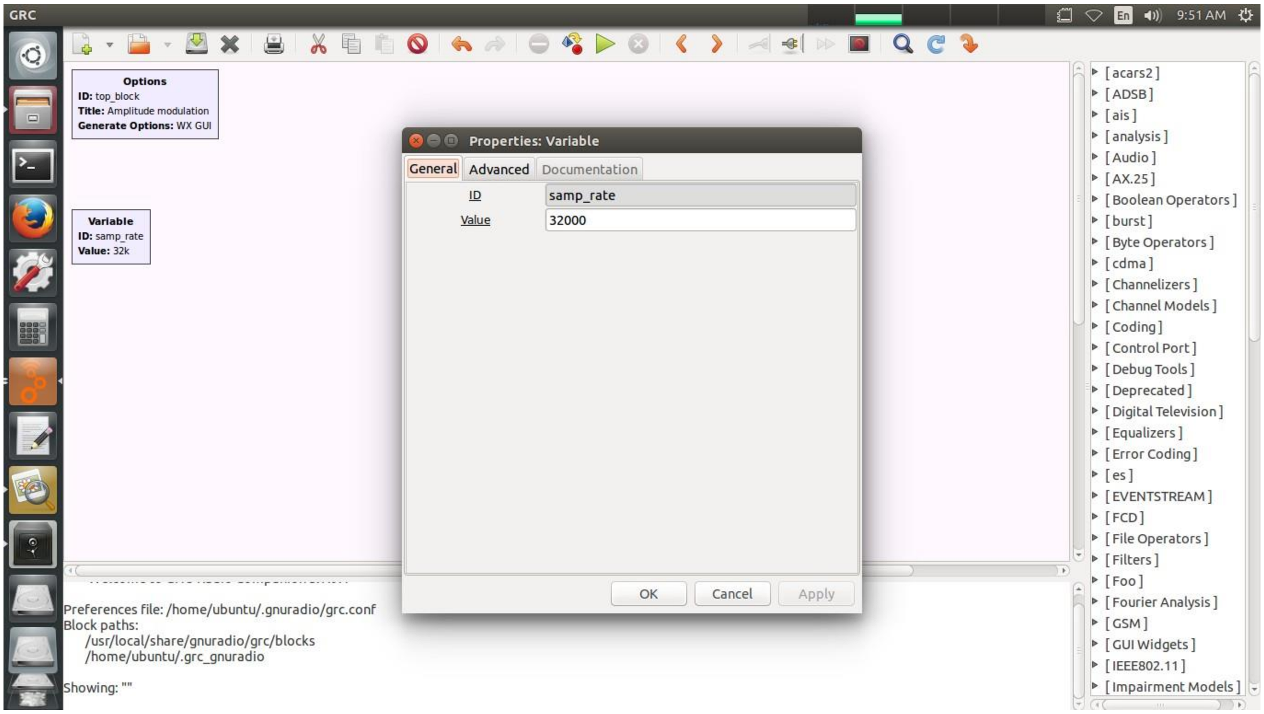
ID: top_block
Title: Amplitude modulation
Author:
Description:
Canvas Size:
Generate Options: WX GUI
Run: Autostart
Max Number of Output: 0
Realtime Scheduling: Off

OK | Cancel | Apply

```
Preferences file: /home/ubuntu/.gnuradio/grc.conf
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""
```

STEP3: Click variable and change ID and value.



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

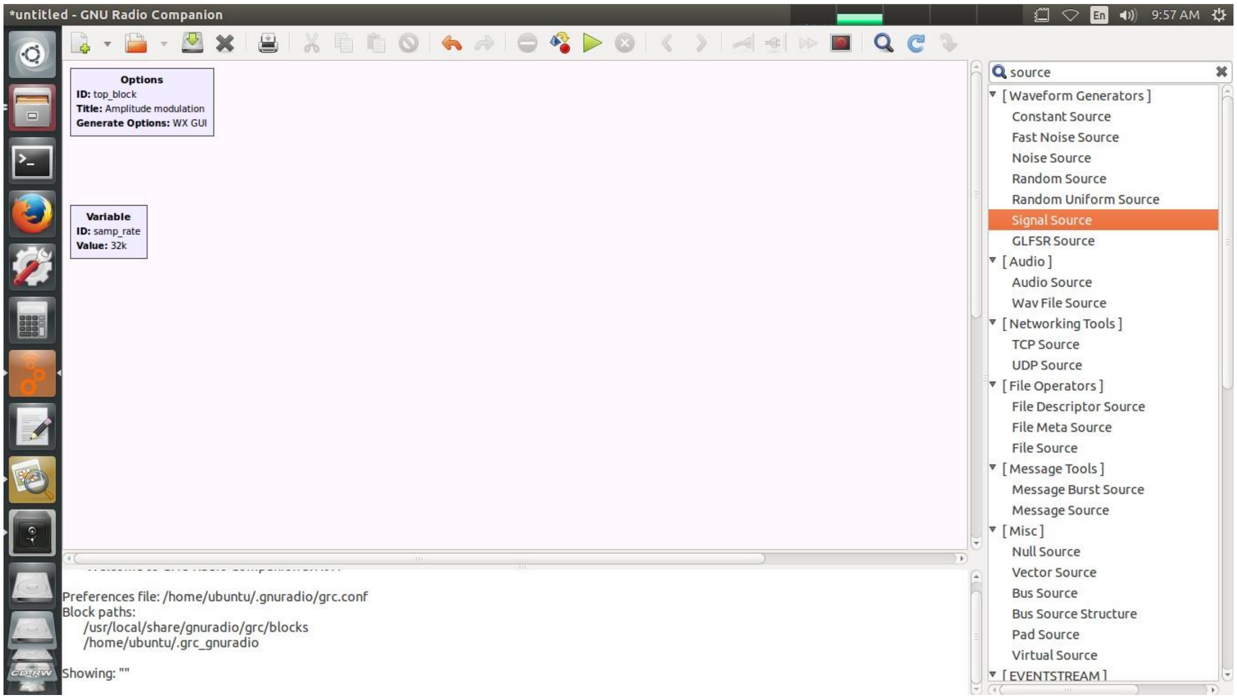
Variable
ID: samp_rate
Value: 32k

ID: samp_rate
Value: 32000

```
Preferences file: /home/ubuntu/.gnuradio/grc.conf
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""
```

STEP 4: Press control+F and search for signal source.



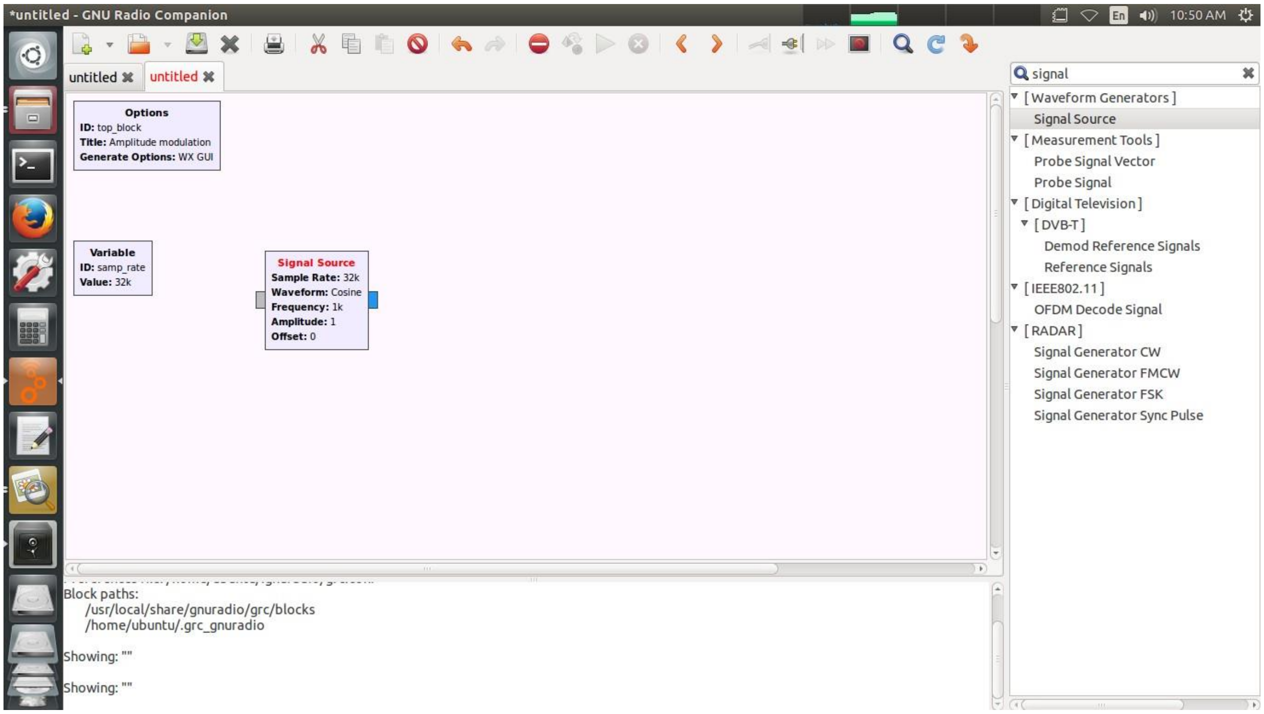
Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

```
Preferences file: /home/ubuntu/.gnuradio/grc.conf
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""
```

STEP 5: Place the signal source for message signal as amplitude modulation.



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

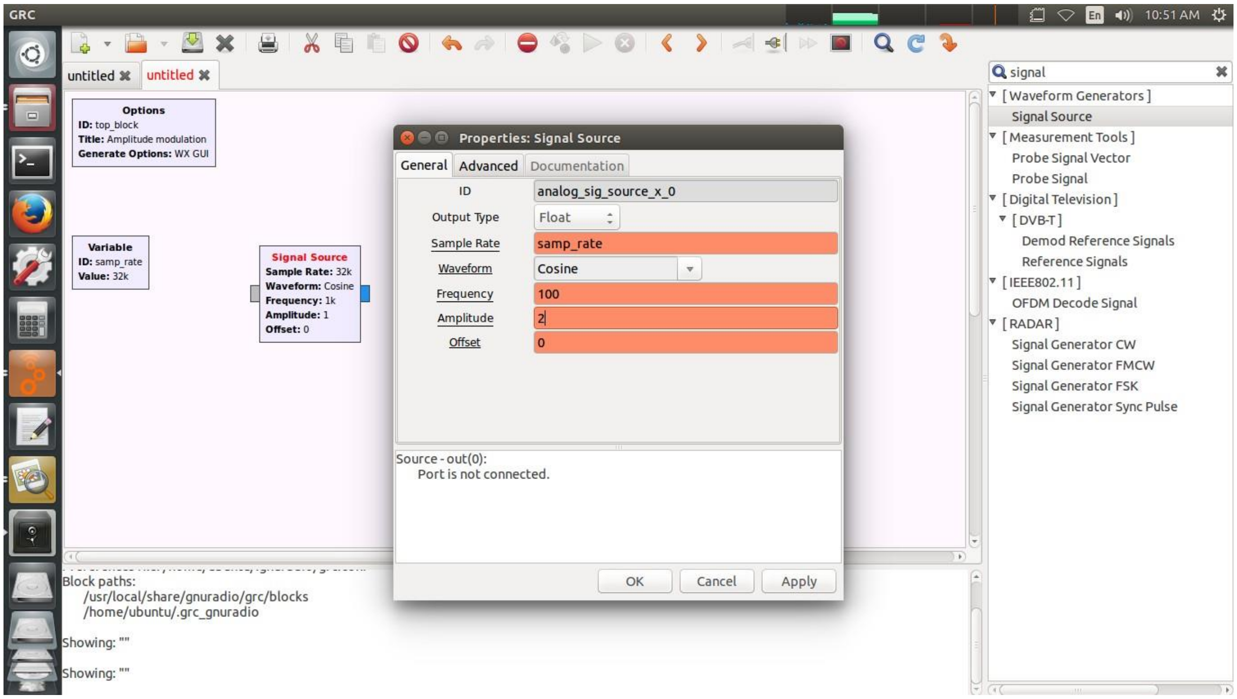
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 6: Change the properties in signal source as (i) Output Type: Float
(ii)Waveform: Cosine
(iii) Frequency: 100
(iv) Amplitude: 2



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

ID	analog_sig_source_x_0
Output Type	Float
Sample Rate	samp_rate
Waveform	Cosine
Frequency	100
Amplitude	2
Offset	0

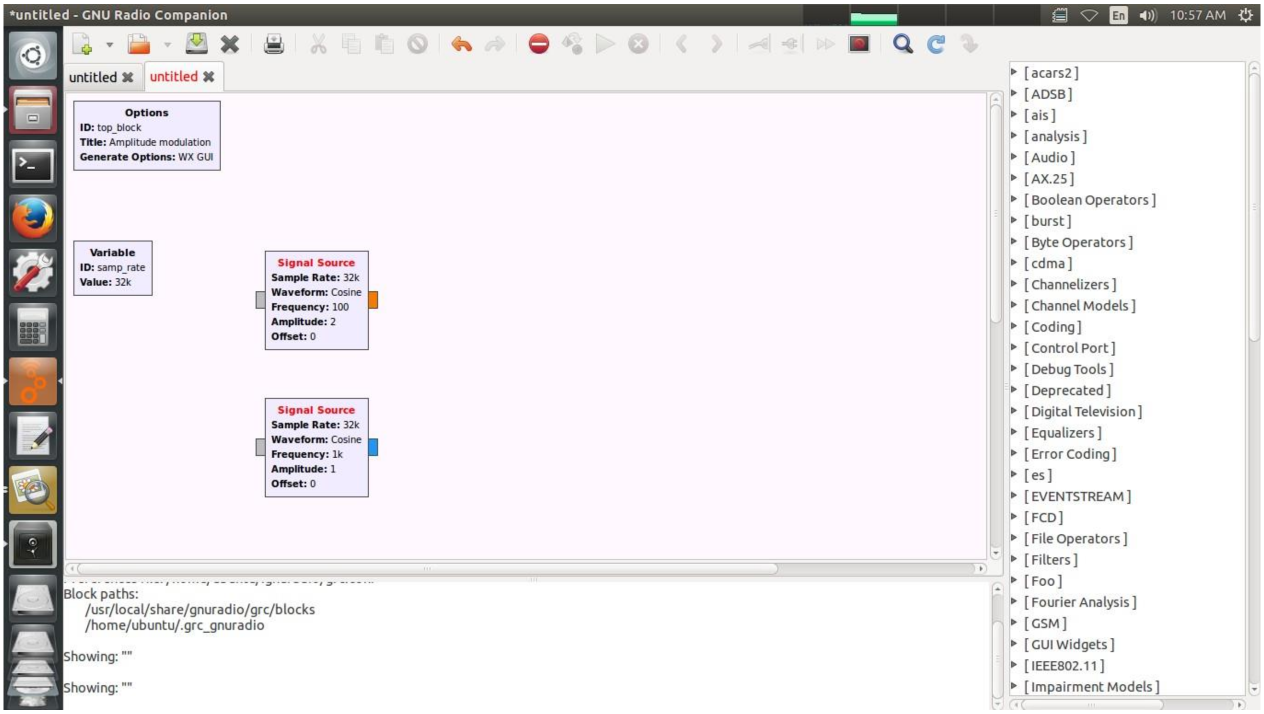
```
Source - out(0):
    Port is not connected.
```

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 7: Place another signal source for carrier signal in amplitude modulation**.**



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

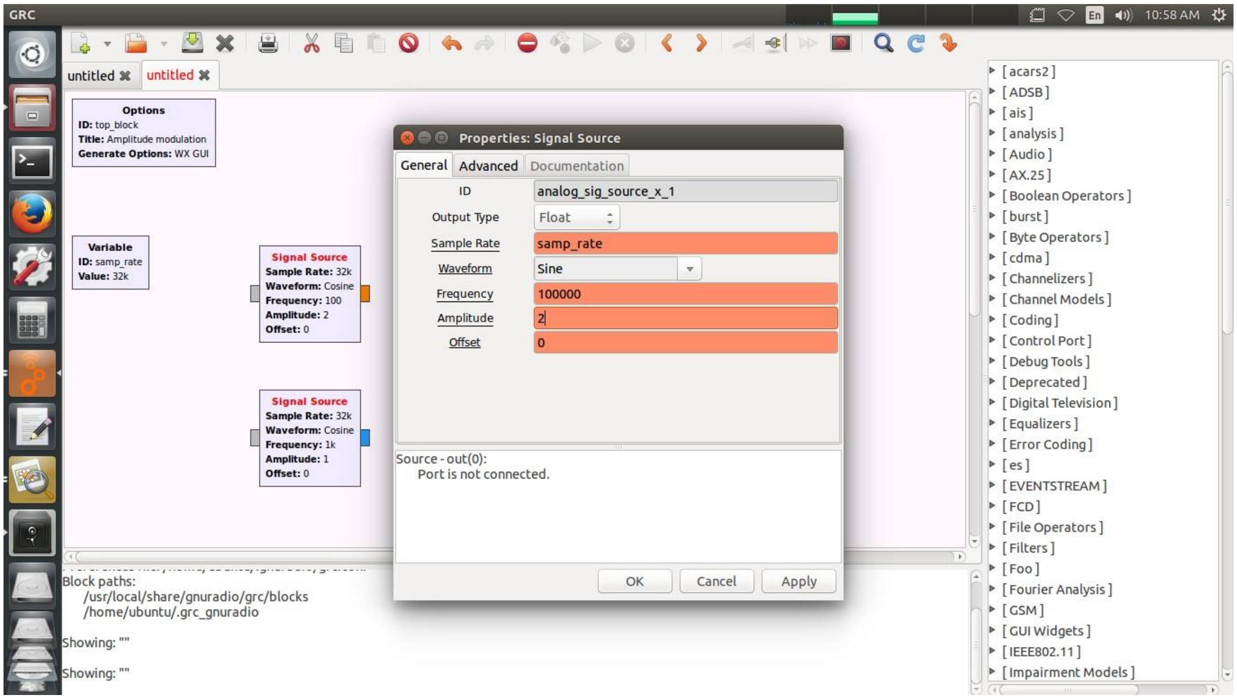
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 8: Change properties in signal source as (i) Output type: float
(ii) Waveform: Sine
(iii) Frequency: 100K
(iv) Amplitude: 2



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

ID	analog_sig_source_x_1
Output Type	Float
Sample Rate	samp_rate
Waveform	Sine
Frequency	100000
Amplitude	2
Offset	0

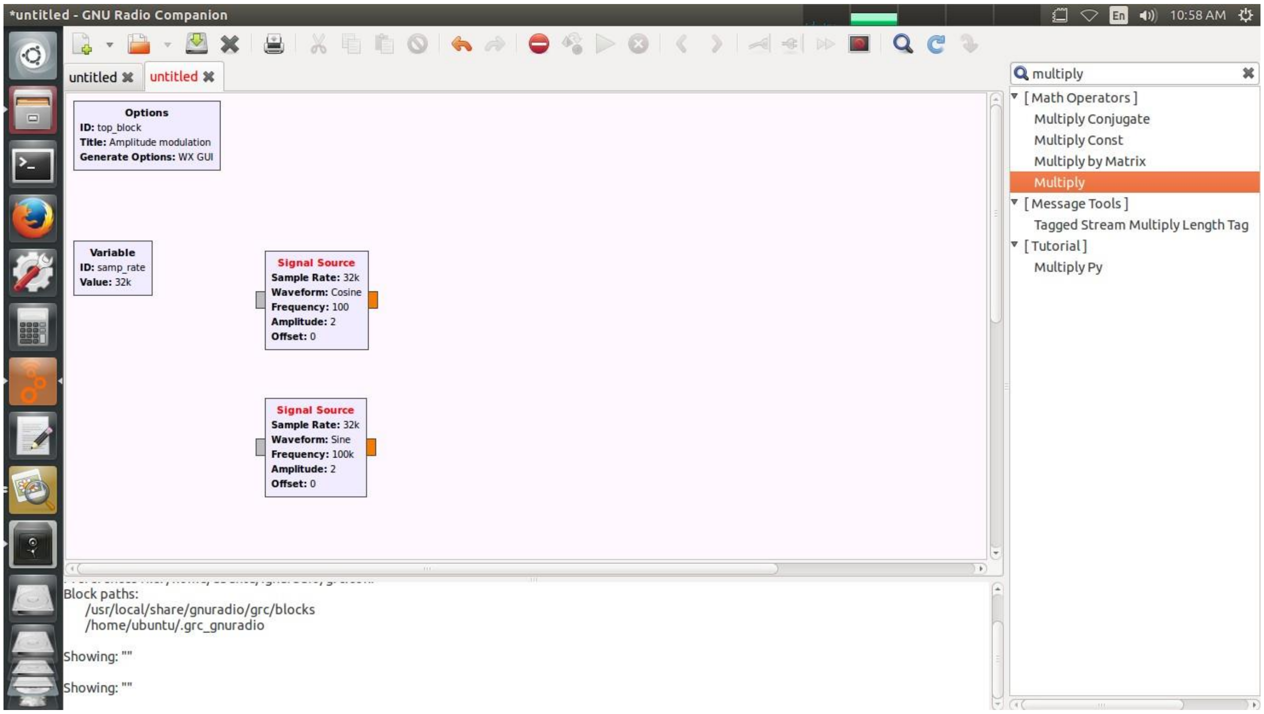
```
Source - out(0):
    Port is not connected.
```

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 9: Press control + F and search Multiply block**.**



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

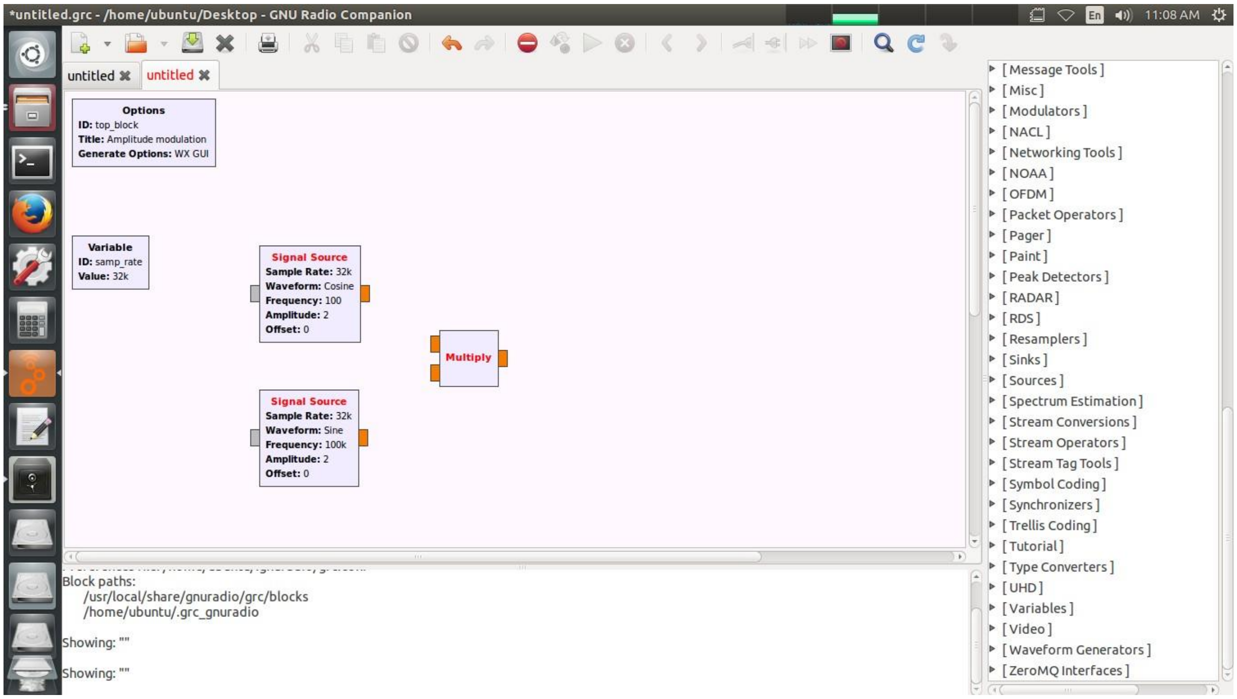
Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 10: Place Multiply for multiply the message signal and carrier signal.



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

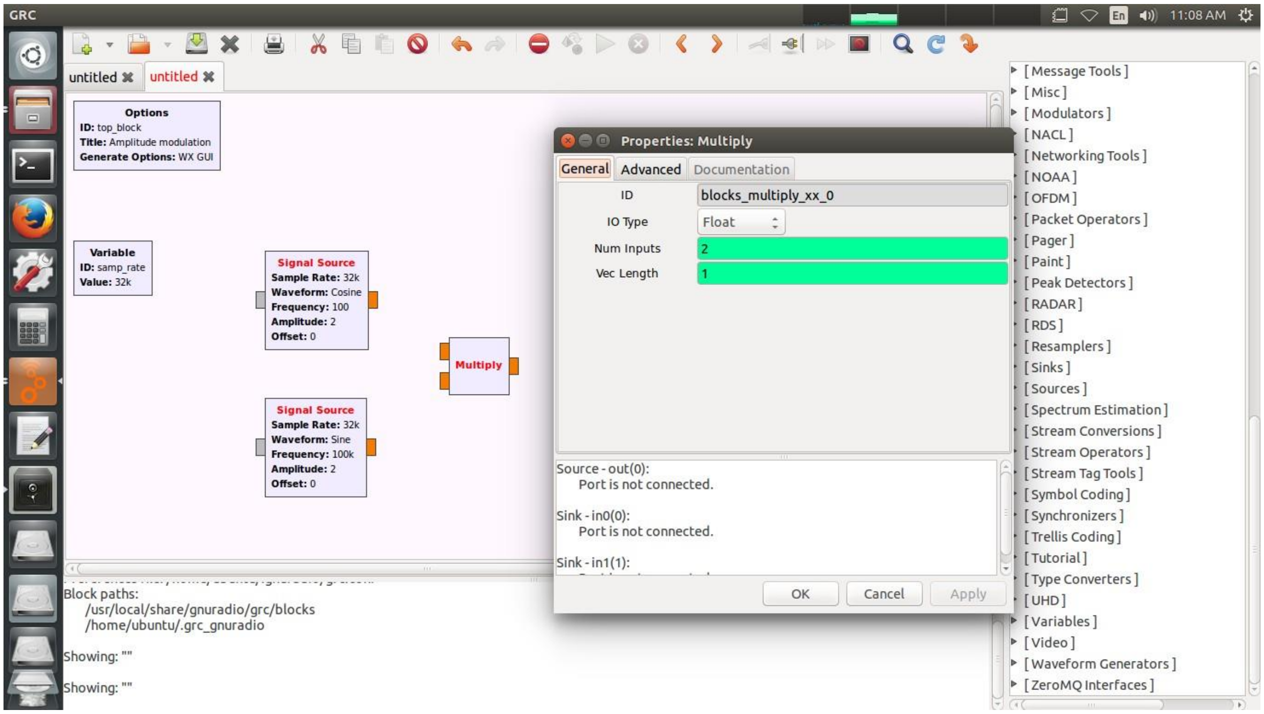
Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 11: Change properties in multiply as (i) ID Type: Float
(ii)Num Inputs: 2



Properties: Multiply

ID: blocks_multiply_xx_0
IO Type: Float
Num Inputs: 2
Vec Length: 1

Source - out(0):
Port is not connected.

Sink - in0(0):
Port is not connected.

Sink - in1(1):

Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

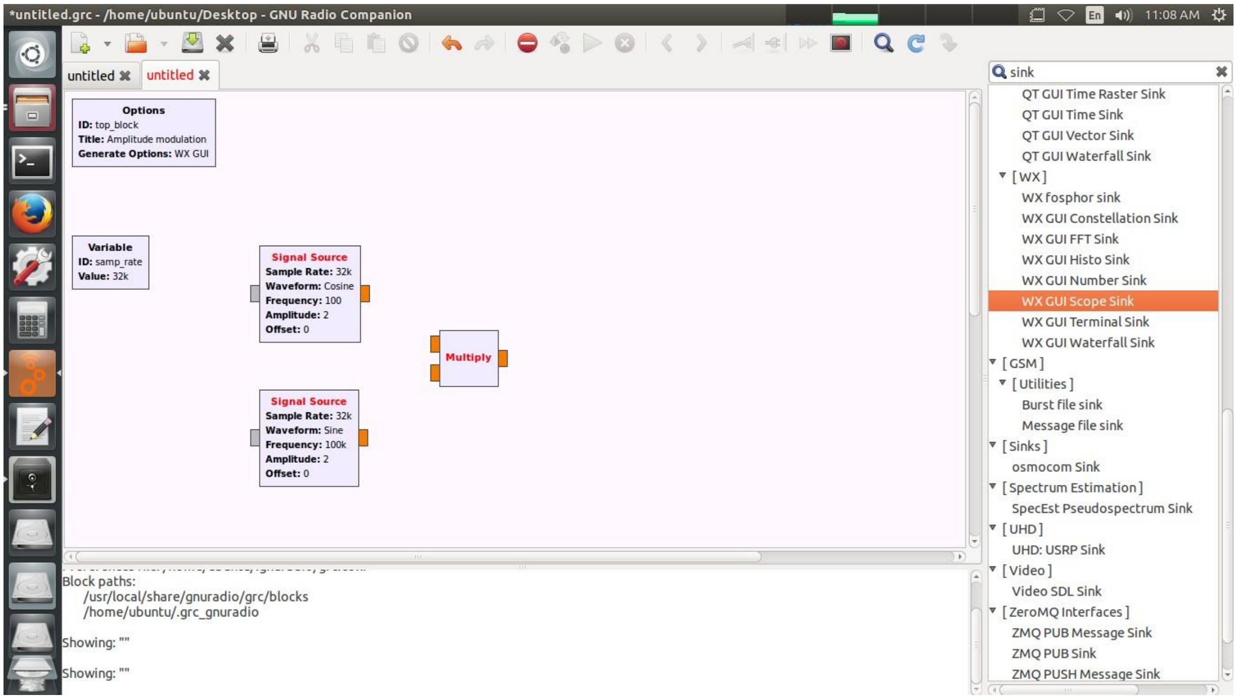
Multiply

Block paths:
/usr/local/share/gnuradio/grc/blocks
/home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""

STEP 12: Press Control + F and search for scope sink and Place WX GUI Scope sink.



*untitled.grc - /home/ubuntu/Desktop - GNU Radio Companion

Search: sink

Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

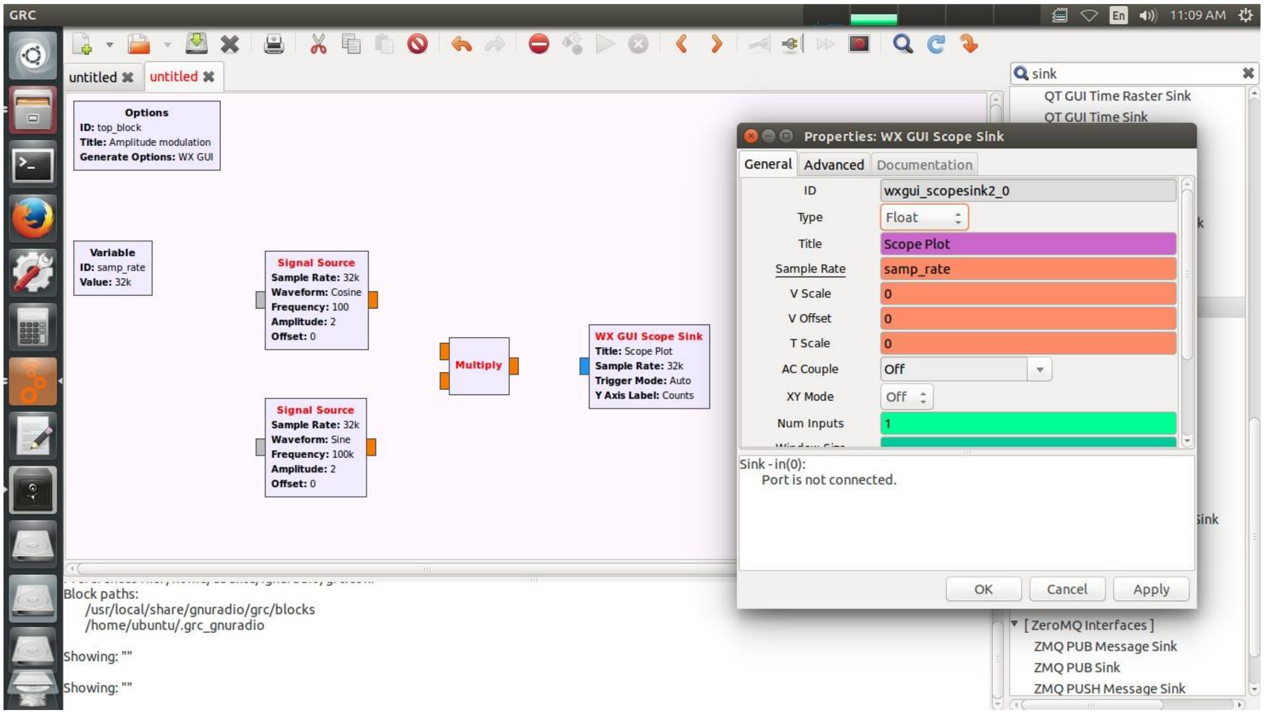
Multiply

Block paths:
/usr/local/share/gnuradio/grc/blocks
/home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""

STEP 13: Change the properties in Wx GUI scope sink as (i) Type: Float (ii)Num input: 2



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

Multiply

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 32k
Trigger Mode: Auto
Y Axis Label: Counts

ID	wxgui_scopesink2_0
Type	Float
Title	Scope Plot
Sample Rate	samp_rate
V Scale	0
V Offset	0
T Scale	0
AC Couple	Off
XY Mode	Off
Num Inputs	1

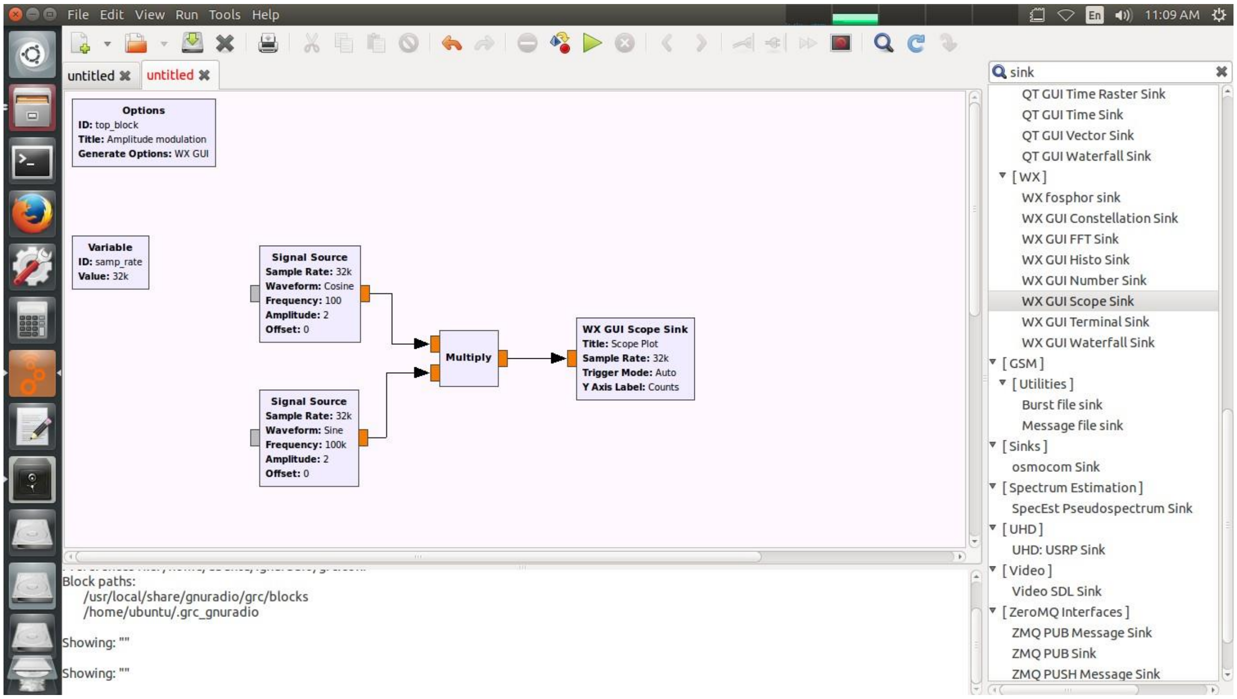
```
Sink - in(0):
    Port is not connected.
```

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 14: Connect wires of signal source to multiply and connect to WX GUI scope sink.



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 100
Amplitude: 2
Offset: 0

Signal Source
Sample Rate: 32k
Waveform: Sine
Frequency: 100k
Amplitude: 2
Offset: 0

Multiply

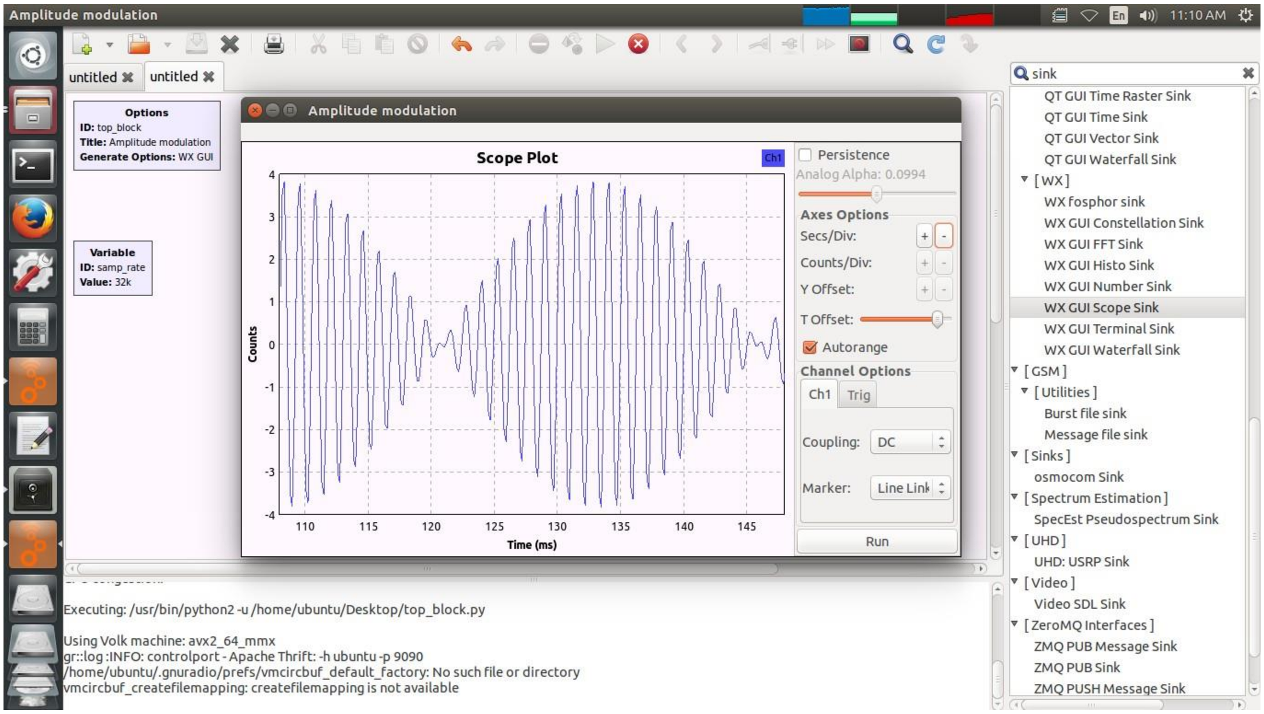
WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 32k
Trigger Mode: Auto
Y Axis Label: Counts

```
Block paths:
    /usr/local/share/gnuradio/grc/blocks
    /home/ubuntu/.grc_gnuradio

Showing: ""

Showing: ""
```

STEP 15: Click Run and stop.



Options
ID: top_block
Title: Amplitude modulation
Generate Options: WX GUI

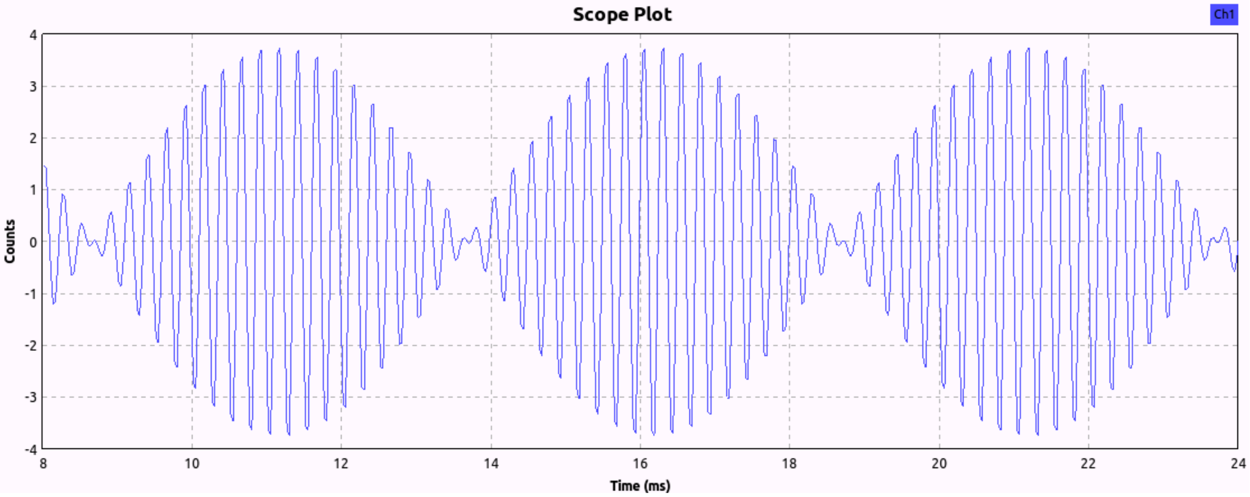
Variable
ID: samp_rate
Value: 32k

□ Persistence
Analog Alpha: 0.0994
☑ Autorange
Coupling: DC
Marker: Line Link

```
Executing: /usr/bin/python2 -u /home/ubuntu/Desktop/top_block.py

Using Volk machine: avx2_64_mmx
gr::log :INFO: controlport - Apache Thrift: -h ubuntu -p 9090
/home/ubuntu/.gnuradio/prefs/vmcircbuf_default_factory: No such file or directory
vmcircbuf_createfilemapping: createfilemapping is not available
```

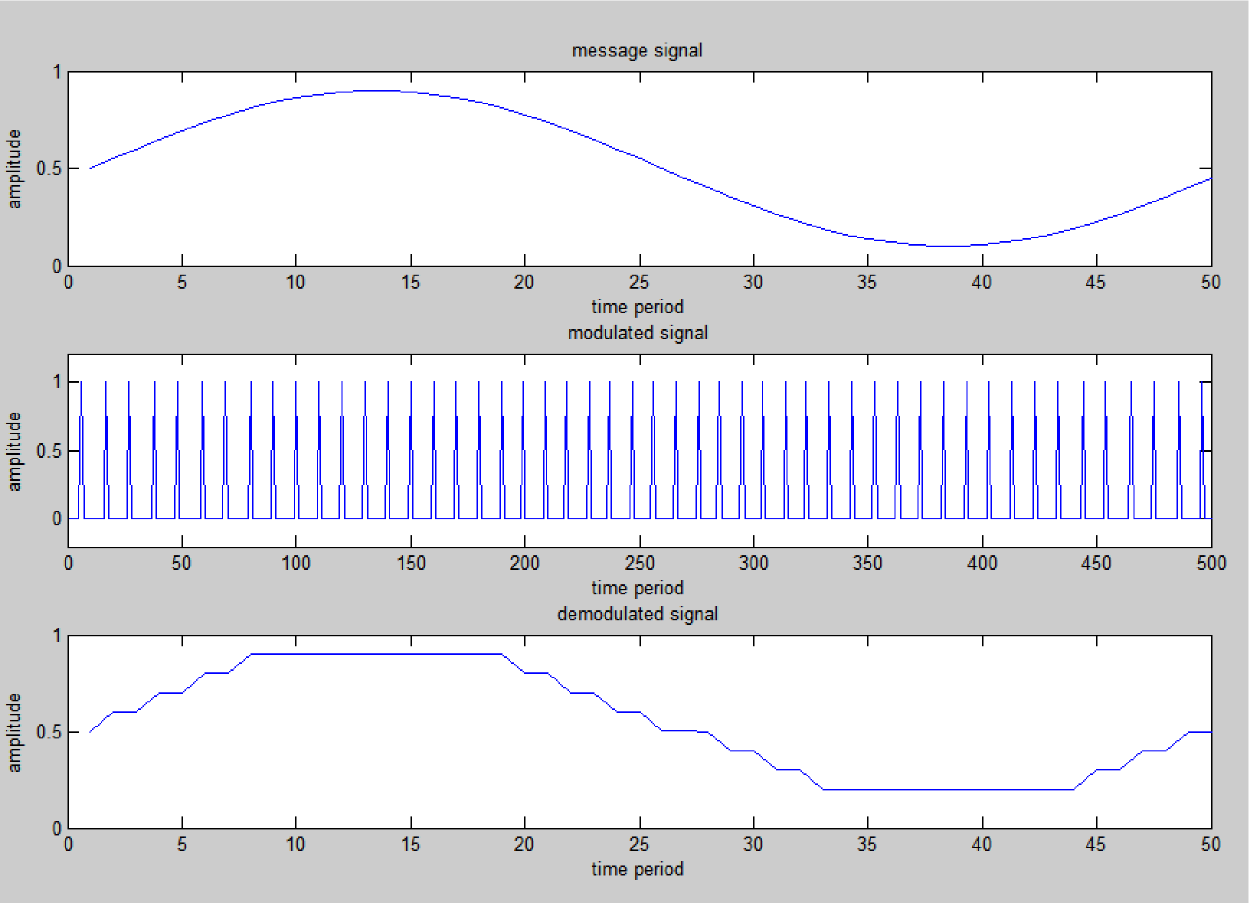
OUTPUT WAVEFORM:



RESULT:

Thus the Amplitude modulation was studied using SDR kit.

OUTPUT WAVEFORM:



PULSE POSITION MODULATION

AIM:

To write and simulate in the MATLAB codes for Pulse position modulation.

APPARATUS REQUIRED:

PC Installed with MATLAB

PROGRAM:

```
clc;
clear all;
close all;
fc=1000;
fs=10000;
fm=200;
t=0:1/fs:(2/fm-1/fs);
mt=0.4*sin(2*pi*fm*t)+0.5;
st=modulate(mt,fc,fs,'PPM');
dt=demod(st,fc,fs,'PPM');
figure
subplot(3,1,1);
plot(mt);
title('message signal');
xlabel('time period');
ylabel('amplitude');
axis([0 50 0 1])
subplot(3,1,2);
plot(st);
title('modulated signal');
xlabel('time period');
ylabel('amplitude');
axis([0 500 -0.2 1.2])
subplot(3,1,3);
plot(dt);
title('demodulated signal');
xlabel('time period');
ylabel('amplitude');
axis([0 50 0 1])
```

RESULT:

Thus the MATLAB code for PPM was written & output is verified.