

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**19EC420 ANALOG COMMUNICATION LAB MANUAL**

**REGULATION – 2024**

**VISION**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**ENGINEERING**

To develop, the department in to a State of Art, with Centre of Excellence in Electronics and Communication Engineering education, on par with global standards.

## MISSION

To provide education to students to enable them to compete internationally, produce creative solutions to the society's needs, and to make them conscious to the universal moral values, adherent to the professional ethical code, generate and disseminate knowledge and technologies essential to the local and global needs in the fields of Electronics and Communication Engineering.

## PROGRAMME OUTCOMES

### Engineering Graduates will be able to,

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## PROGRAMME EDUCATIONAL OBJECTIVES

1. To prepare students for successful careers in industry that meet the needs of Indian and multinational companies.
2. To develop the ability among students to synthesize data and technical concepts for application to product design.
3. To provide opportunity for students to work as part of teams on multidisciplinary projects.
4. To provide students with a sound foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems and to prepare them for graduate studies.
5. To promote student awareness of the life-long learning and to introduce them to

professional ethics and codes of professional practice.

## PROGRAM SPECIFIC OUTCOMES

* 1. An ability to apply creativity in design and development of electronic circuits, equipments, systems.
  2. An ability to apply existing hardware and software programming skills in real time.
  3. An ability to comprehend and apply the knowledge of wired and wireless systems in the electronics and communication applications.

## COURSE OUTCOMES:

|  |  |  |
| --- | --- | --- |
| CO1 | Examine AM communication systems and study their properties. | Apply |
| CO2 | Distinguish and implement FM and PM modulation techniques | Analyse |
| CO3 | Apply Core Principles and Applications of Stochastic Processes | Apply |
| CO4 | Illustrate Stochastic Processes in Linear System Environments | Analyse |
| CO5 | Employ AI algorithms in analog communication system design | Apply |

**MAPPING OF COs WITH POs AND PSOs:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **COURSE OUTCOMES** | **PROGRAMME OUTCOMES** | | | | | | | | | | | | **PROGRAMME**  **SPECIFIC OUTCOMES** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 | 2 | 3 | 2 | 3 | - | - | - | - | - | 1 | 2 | 2 | 3 | 2 |
| **CO2** | 3 | 2 | 3 | 2 | 3 | - | - | - | - | - | 1 | 2 | 2 | 3 | 3 |
| **CO3** | 2 | 2 | 2 | 2 | 2 | - | - | - | - | - | 2 | 2 | 2 | 2 | 2 |
| **CO4** | 2 | 2 | 2 | 2 | 3 | - | - | - | - | - | 2 | 2 | 2 | 2 | 2 |
| **CO5** | 3 | 2 | 3 | 2 | 3 | - | - | 1 | - | - | 2 | 2 | 2 | 3 | 3 |

## 19EC420 ANALOG COMMUNICATION L T P C

**3 0 2 4**

## LIST OF EXPERIMENTS

1. Generation and detection of AM using SCILAB
2. DSB-SC-AM modulator and demodulator using SCILAB
3. SSB-SC-AM modulation and demodulation using SCILAB.
4. Generation and detection of FM using SCILAB
5. Simulation of Mean, variance and Cross Correlation using SCILAB**.**
6. Simulation of Auto correlation and PSD using SCILAB
7. Generation and detection of AM Modulation and Demodulation using Python.
8. Generation and detection of FM Modulation and Demodulation using Python.
9. Generation and detection of PM Modulation and Demodulation using Python.
10. Evaluation of Maximum range of Radar using Radar range Equation in Python.
11. Evaluation of Time Division Multiplexing (Beyond Syllabus)

### LAB Requirements for a Batch of 30 students (3 students per experiment)

CROs/DSOs – 15 Nos, Function Generators – 15 Nos. SCILAB software package for simulation experiments PCs - 15 Nos

## EXP NO: 1 GENERATION AND DETECTION OF AM

**AIM:**

To generate and detect the amplitude modulation and demodulation u s i n g S C I L A B and to calculate modulation index of AM.

**EQUIPMENTS REQUIRED**

* Computer with i3 Processor
* SCI LAB

## THEORY:

Modulation can be defined as the process by which the characteristics of carrier wave are varied in accordance with the modulating wave (signal). Modulation is performed in a transmitter by a circuit called a modulator.

Need for modulation is as follows:

* + Avoid mixing of signals
  + Reduction in antenna height
  + long distance communication
  + Multiplexing
  + Improve the quality of reception
  + Ease of radiation.

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. The output waveform contains all the frequencies that make up the AM signal and is used to transport the information through the system. Therefore the shape of the modulated wave is called the AM envelope. With no modulating signal the output waveform is simply the carrier signal. Coefficient of modulation is a term used to describe the amount of amplitude change present in an AM waveform. There are three degrees of modulation available based on value of modulation index.

* 1. Under modulation : m<1, Em < Ec
  2. Critical modulation: m-1, Em = Ec
  3. Over modulation: m>1, Em > Ec

**Note:** Keep all the switch faults in off position

### Algorithm

1. **Define Parameters**

First, define the parameters for your signals:

* + Carrier frequency (fc)
  + Modulating signal frequency (fm)
  + Sampling frequency (Fs)
  + Duration of the signal (T)

1. Create a time vector based on the sampling frequency and duration.

### Create Modulating Signal

Define the modulating signal (message signal).

### Create Carrier Signal

Define the carrier signal.

### Perform Amplitude Modulation

Multiply the carrier signal by the modulating signal plus 1 (to ensure the modulation depth).

### Plot the Signals

Visualize the modulating, carrier, and modulated signals.

### Demodulate the AM Signal

To demodulate, you can use envelope detection. One way is to rectify the signal and then apply a low-pass filter.

### Plot the Demodulated Signal

Visualize the demodulated signal.

### Compare Signals

**Compare the original modulating signal with the demodulated signal. PROCEDURE**

* + Refer Algorithms and write code for the experiment.
  + Open SCILAB in System
  + Type your code in New Editor
  + Save the file
  + Execute the code
  + If any Error, correct it in code and execute again
  + Verify the generated waveform using Tabulation and Model Waveform

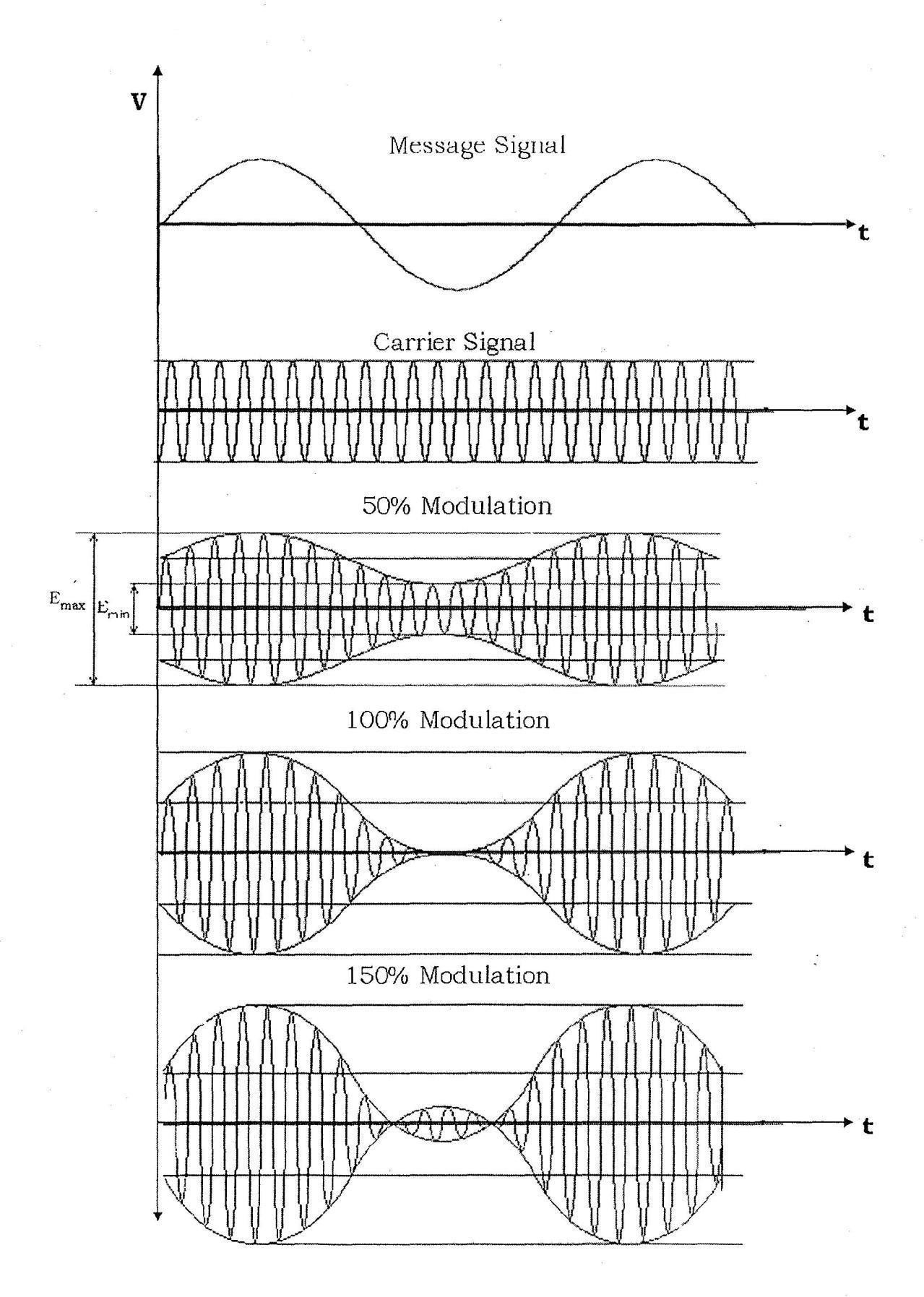
**TABULATION:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl. No. | SIGNAL |  | AMPLITUDE(v) | Time | FREQUENCY (hz) |
| 1 | Message Signal | Theory (am) |  |  |  |
| Practical |  |  |  |
| 2 | Carrier Signal | Theory (ac) |  |  |  |
| Practical |  |  |  |
| 3 | Modulated Signal | Practical | Emax =  Emin = |  |  |
| 4. | Demodulat ed Signal | Practical |  |  |  |

# Calculation

1. **ma (Theory) = am/ac =**
2. **ma(Practical) = (Emax-Emin)/(Emax+Emin) =**

## MODEL GRAPH



**RESULT:**

Thus the amplitude modulation and demodulation is experimentally done and the output is verified.

**Viva questions on Amplitude Modulation (AM) and Demodulation:**

### Basic Concepts

1. **What is Amplitude Modulation (AM)?**
   * **Answer**: Amplitude Modulation (AM) is a technique where the amplitude of a carrier signal is varied in proportion to the amplitude of the message signal while keeping the carrier frequency and phase constant. This technique is used to encode information in a carrier wave.

### Explain the difference between Amplitude Modulation (AM) and Frequency Modulation (FM).

* + **Answer**: In AM, the amplitude of the carrier signal varies according to the message signal, while in FM, it is the frequency of the carrier signal that varies according to the message signal. AM is simpler but more susceptible to noise compared to FM.

### Modulation Process

1. **How do you calculate the modulation index in an AM signal?**
   * **Answer**: The modulation index (m) in AM is calculated using the formula m=Am/Ac where Am is the peak amplitude of the message signal and Ac is the peak amplitude of the carrier signal.

### What are the components of an AM signal?

* + **Answer**: An AM signal consists of three components: the carrier signal, the upper sideband (USB), and the lower sideband (LSB). The carrier frequency is the central frequency, while the sidebands carry the actual information of the message signal.

### Demodulation Process

1. **Describe the basic principle of envelope detection used in AM demodulation.**
   * **Answer**: Envelope detection in AM demodulation involves extracting the varying envelope of the modulated signal. The envelope of an AM signal is directly proportional to the message signal, and it can be recovered using a rectifier followed by a low-pass filter to smooth out the signal.

### What is the role of a local oscillator in the process of AM demodulation?

* + **Answer**: In AM demodulation, the local oscillator in a demodulator generates a signal that is mixed with the incoming modulated signal to produce a difference frequency that can be easily filtered. This process helps in recovering the original baseband signal.

### Practical Considerations

1. **How does the presence of noise affect an AM signal, and what techniques can be used to mitigate it?**
   * **Answer**: Noise affects an AM signal by causing amplitude variations that can distort the demodulated signal. Techniques to mitigate noise include using band-pass filters to limit the bandwidth and employing noise reduction algorithms in the receiver.

### Why is the carrier signal necessary in AM modulation?

* + **Answer**: The carrier signal is necessary in AM modulation because it provides a fixed frequency reference for the modulation process. It also allows the modulated signal to be transmitted over the air or through a medium, making it possible to recover the message signal at the receiver end.

### What is the bandwidth of an AM signal?

* + Answer: The bandwidth of an AM signal is twice the maximum frequency of the message signal. If the message signal has a maximum frequency fm, the total bandwidth of the AM signal will be 2fm. This is because the signal consists of two sidebands (USB and LSB), each with a width equal to the maximum frequency of the message signal.

### How can you visualize the spectrum of an AM signal using Scilab?

**Answer**: In Scilab, you can visualize the spectrum of an AM signal using the Fast Fourier Transform (FFT). Compute the FFT of the modulated signal and plot its magnitude spectrum to observe the carrier and sidebands. This helps in analyzing the frequency components of the AM signal.

**EX NO: 2 DSB-SC-AM MODULATOR AND DEMODULATOR**

**AIM:**

To write a program to perform DSBSC modulation and demodulation using SCI LAB and study its spectral characteristics

**EQUIPMENTS REQUIRED**

* Computer with i3 Processor
* SCI LAB

**Note:** Keep all the switch faults in off position

**Algorithm:**

1. **Define Parameters:**
   * **Fs: Sampling frequency.**
   * **T: Duration of the signal.**
   * **Fc: Carrier frequency.**
   * **Fm: Frequency of the message signal.**
   * **Amplitude: Maximum amplitude of the message signal.**
2. **Generate Signals:**
   * **Message Signal: A sinusoidal signal that will be modulated.**
   * **Carrier Signal: A high-frequency sinusoidal signal used for modulation.**
3. **DSBSC Modulation:**
   * **Modulated Signal: Multiply the message signal by the carrier signal to produce the DSBSC signal.**
4. **DSBSC Demodulation:**
   * **Multiplication: Multiply the modulated signal by the carrier signal to get the product of the message signal with itself (i.e., the original message signal plus high-frequency components).**
   * **Low-pass Filtering: Apply a Butterworth low-pass filter to remove the high- frequency components and recover the original message signal.**
5. **Visualization:**

**Plot the message signal, carrier signal, DSBSC modulated signal, and the recovered signal after demodulation.**

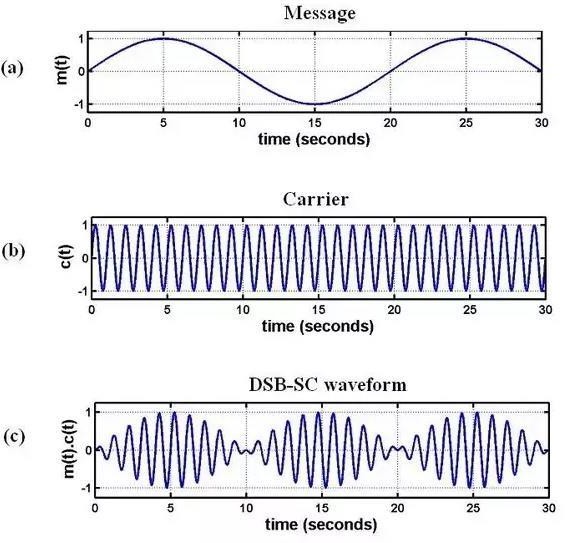
## PROCEDURE

* + Refer Algorithms and write code for the experiment.
  + Open SCILAB in System
  + Type your code in New Editor
  + Save the file
  + Execute the code
  + If any Error, correct it in code and execute again
  + Verify the generated waveform using Tabulation and Model Waveform

## TABULATION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | SIGNAL |  | AMPLITUDE(v) | FREQUENCY (hz) |
| 1 | Message Signal | Theory |  |  |
| Practical |  |  |
| 2 | Carrier Signal | Theory |  |  |
| Practical |  |  |
| 3 | Modulated Signal | Practical | Emax = Emin = |  |
| 4. | Demodulat ed Signal | Practical |  |  |

### Model Graph:



* ​

### Result:

Thus the DSB-SC-AM Modulation and Demodulation is generated.

**Viva questions on Double Sideband Suppressed Carrier (DSBSC) modulation and demodulation**

### Basic Concepts

1. **What is Double Sideband Suppressed Carrier (DSBSC) modulation?**
   * **Answer**: DSBSC modulation is a type of amplitude modulation where the carrier signal is not transmitted, only the sidebands are. The modulated signal contains only the upper and lower sidebands, which carry the information of the message signal.

### How does DSBSC differ from standard AM modulation?

* + **Answer**: In standard AM modulation, the carrier signal is transmitted along with the sidebands, whereas in DSBSC modulation, the carrier is suppressed, and only the sidebands are transmitted. This makes DSBSC more power-efficient as it does not require the transmission of the carrier.

### Modulation Process

1. **What are the advantages of using DSBSC modulation?**
   * **Answer**: The primary advantages of DSBSC modulation are increased power efficiency (since no power is wasted on transmitting the carrier) and reduced bandwidth usage compared to standard AM modulation, though it still requires a coherent detector at the receiver.

### How do you generate a DSBSC signal mathematically?

* + **Answer**: A DSBSC signal is generated by multiplying the message signal m(t)m(t)m(t) with a cosine wave of carrier frequency fcf\_cfc, given by s(t)=m(t)⋅cos⁡(2⋅π⋅fc⋅t)s(t) = m(t) \cdot \cos(2 \cdot \pi \cdot f\_c \cdot t)s(t)=m(t)⋅cos(2⋅π⋅fc⋅t), where m(t)m(t)m(t) is the message signal, and fcf\_cfc is the carrier frequency.

### Demodulation Process

1. **What is the principle of coherent detection in DSBSC demodulation?**
   * **Answer**: Coherent detection involves mixing the received DSBSC signal with a locally generated carrier signal that has the same frequency and phase as the original carrier. This process regenerates the original message signal by producing a baseband signal that can be filtered to retrieve the message.

### Describe the role of a local oscillator in the DSBSC demodulation process.

* + **Answer**: The local oscillator generates a signal at the carrier frequency that is used to mix with the received DSBSC signal. The result of this mixing process is a baseband signal that contains the original message signal, allowing it to be recovered.

### Practical Considerations

1. **How does the absence of a carrier affect the demodulation of DSBSC signals?**
   * **Answer**: The absence of a carrier in DSBSC signals requires that the receiver uses a coherent detector to accurately recover the message signal. If the local oscillator is not perfectly synchronized in frequency and phase with the original carrier, demodulation can lead to errors and distortion.

### What are some common challenges associated with DSBSC modulation?

* + **Answer**: Challenges with DSBSC modulation include the need for precise frequency and phase synchronization in the demodulation process, susceptibility to noise, and the requirement for a coherent detector. These factors can complicate the implementation and increase system complexity.

### Scilab Implementation

1. **How can you simulate DSBSC modulation in Scilab?**
   * **Answer**: In Scilab, you can simulate DSBSC modulation by generating a message signal and a carrier signal, then multiplying them to produce the DSBSC signal.

### How do you perform coherent demodulation of a DSBSC signal using Scilab?

**Answer**: Coherent demodulation in Scilab involves mixing the received DSBSC signal with a locally generated carrier signal and then applying a low-pass filter to extract the message signal.

**EXPT: 3 SSB-SC-AM MODULATOR AND DEMODULATOR**

**AIM:**

To write a program to perform SSBSC modulation and demodulation using SCI LAB and study its spectral characteristics

**EQUIPMENTS REQUIRED**

* Computer with i3 Processor
* SCI LAB

**Note:** Keep all the switch faults in off position

### Algorithm

1. **Define Parameters:**

* Fs: Sampling frequency.
* T: Duration of the signal.
* Fc: Carrier frequency.
* Fm: Frequency of the message signal.
* Amplitude: Maximum amplitude of the message signal.

### Generate Signals:

* Message Signal: The baseband signal that will be modulated.
* Carrier Signal: A high-frequency signal used for modulation.
* Analytic Signal: Constructed using the Hilbert transform to get the in-phase and quadrature components.

### SSBSC Modulation:

* Modulated Signal: Create the SSBSC signal using the in-phase and quadrature components, modulated by the carrier.

### SSBSC Demodulation:

* Mixing: Multiply the SSBSC signal with the carrier to retrieve the message signal.
* Low-pass Filtering: Apply a low-pass filter to remove high-frequency components and recover the original message signal.

### Visualization:

Plot the message signal, carrier signal, SSBSC modulated signal, and the recovered signal after demodulation.

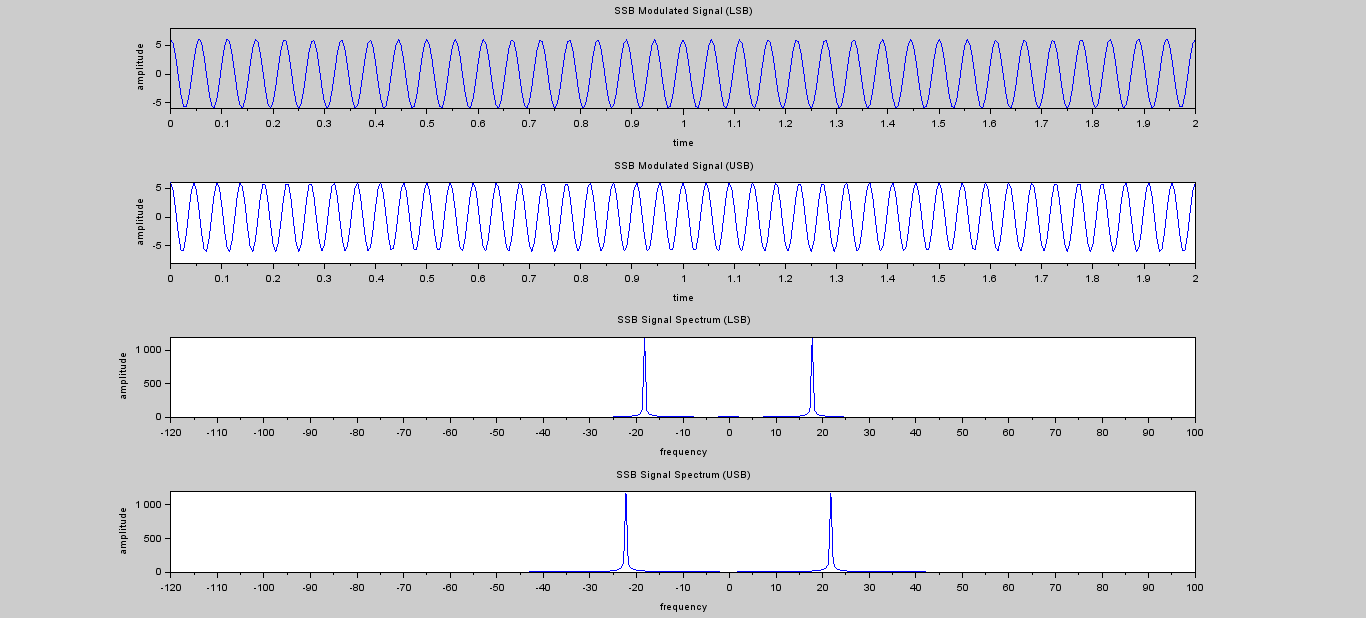
## PROCEDURE

* + Refer Algorithms and write code for the experiment.
  + Open SCILAB in System
  + Type your code in New Editor
  + Save the file
  + Execute the code
  + If any Error, correct it in code and execute again
  + Verify the generated waveform using Tabulation and Model Waveform

## TABULATION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | SIGNAL |  | AMPLITUDE(v) | FREQUENCY (hz) |
| 1 | Message Signal | Theory |  |  |
| Practical |  |  |
| 2 | Carrier Signal | Theory |  |  |
| Practical |  |  |
| 3 | Modulated Signal | Practical | Emax = Emin = |  |
| 4. | Demodulat ed Signal | Practical |  |  |

**Model Graph:**

****

## RESULT:

Thus, the SSB-SC-AM Modulation and Demodulation is experimentally done and the output is verified.

**Viva questions on Single Sideband Suppressed Carrier (SSBSC) modulation and demodulation:**

### Basic Concepts

1. **What is Single Sideband Suppressed Carrier (SSBSC) modulation?**
   * **Answer**: SSBSC modulation is a form of amplitude modulation where only one sideband (either the upper or lower sideband) is transmitted, and the carrier is suppressed. This technique is used to reduce bandwidth and improve power efficiency.

### How does SSBSC differ from Double Sideband Amplitude Modulation (DSBAM)?

* + **Answer**: In DSBAM, both the upper and lower sidebands are transmitted along with the carrier. In contrast, SSBSC transmits only one sideband and suppresses the carrier and the other sideband, resulting in reduced bandwidth and power consumption.

### Modulation Process

1. **What are the advantages of SSBSC modulation over standard AM?**
   * **Answer**: The main advantages of SSBSC modulation are reduced bandwidth usage (since only one sideband is transmitted) and improved power efficiency (since no power is wasted on the carrier). This leads to better utilization of the available frequency spectrum.

### How do you generate an SSBSC signal mathematically?

* + **Answer**: An SSBSC signal is generated by filtering the DSBSC signal to retain only one of the sidebands. Mathematically, if m(t) is the message signal and cos⁡(2πfct) is the carrier signal, you can use a Hilbert transform to generate the analytic signal and then filter to obtain the SSBSC signal.

### Demodulation Process

1. **Describe the principle of synchronous detection in SSBSC demodulation.**
   * **Answer**: Synchronous detection involves mixing the received SSBSC signal with a locally generated carrier signal that matches the original carrier frequency and phase. This process retrieves the original message signal by generating a baseband signal from the modulated sideband.

### What is the role of a Hilbert transform in SSBSC modulation and demodulation?

* + **Answer**: The Hilbert transform is used to create an analytic signal from a real signal. In SSBSC modulation, it helps in generating a single sideband by shifting the phase of the message signal. In demodulation, it helps in reconstructing the original message signal from the SSBSC signal.

### Practical Considerations

1. **What are the main challenges in implementing SSBSC modulation and demodulation?**
   * **Answer**: Challenges include maintaining precise synchronization of the local oscillator in the receiver, ensuring accurate phase and frequency alignment, and managing signal distortion and noise. These factors can complicate the demodulation process and require careful design.

### How does the suppression of the carrier affect the recovery of the message signal?

* + **Answer**: The suppression of the carrier requires the use of coherent detection to recover the message signal accurately. The receiver must generate a carrier signal that matches the original in frequency and phase to properly demodulate the SSBSC signal.

### Scilab Implementation

1. **How can you simulate SSBSC modulation in Scilab?**
   * **Answer**: In Scilab, you can simulate SSBSC modulation by using the Hilbert transform to create an analytic signal and then filtering to isolate one sideband

### How do you perform coherent demodulation of an SSBSC signal using Scilab?

**Answer**: Coherent demodulation in Scilab involves mixing the SSBSC signal with a locally generated carrier signal and then using a low-pass filter to extract the message signal.

## EXP NO : 4 GENERATION AND DETECTION OF FM

**AIM:**

To write a program for Frequency Modulation and Demodulation using SCILAB and to observe and measure the frequency deviation and the modulation index of FM.

**EQUIPMENTS REQUIRED**

* Computer with i3 Processor

## SCI LAB

**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 DEVIATION *f* and MODULATION INDEX *m f* :

The frequency deviation *f* represents the maximum shift between the modulatedsignal

frequency, over and under the frequency of the carrier.

We define modulation index m *f* the ratio between *f a*nd the modulating frequency

m= *f / fm*

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.

### Algorithm

1. **Define Parameters:**

* Fs: Sampling frequency.
* T: Duration of the signal.
* Fc: Carrier frequency.
* Fm: Frequency of the modulating signal.
* Beta: Modulation index, which controls the extent of frequency deviation.

### Generate Signals:

* modulating\_signal: Sinusoidal signal used for modulation.
* carrier\_signal: The high-frequency carrier signal.
* modulated\_signal: FM modulated signal calculated by varying the carrier frequency according to the modulating signal.

### FM Modulation:

* Modulated\_signal is obtained by modulating the carrier signal with the modulating signal.

### FM Demodulation:

* Differentiation: Computes the derivative of the modulated signal to extract frequency variations.
* Envelope Detection: Takes the absolute value to retrieve the envelope of the signal.
* Low-pass Filtering: Applies a Butterworth low-pass filter to smooth the envelope and recover the original modulating signal.

### Visualization:

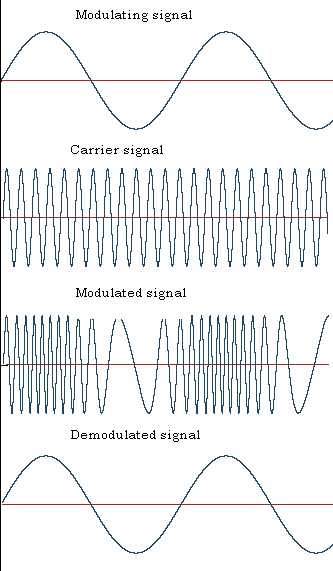
* Plots the modulating signal, carrier signal, FM modulated signal, and demodulated signal for analysis.

## PROCEDURE

* + Refer Algorithms and write code for the experiment.
  + Open SCILAB in System
  + Type your code in New Editor
  + Save the file
  + Execute the code
  + If any Error, correct it in code and execute again

**Verify the generated waveform using Tabulation and Model Waveform**

## MODEL GRAPH:

****

**TABULATION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | SIGNAL |  | AMPLITUDE(v) | FREQUENCY (hz) |
| 1 | Message Signal | Theory |  |  |
| Practical |  |  |
| 2 | Carrier Signal | Theory |  |  |
| Practical |  |  |
| 3 | Modulated Signal | Practical |  | fmax = fmin = |
| 4. | Demodulat ed Signal | Practical |  |  |

Calculation

Frequency Deviation Practical = Modulation Index Practical = Modulation Index Theoretical =

## RESULT:

Thus the frequency modulation and demodulation is successfully done and the output is experimentally verified.

**Viva questions for an experiment on Frequency Modulation (FM) and Demodulation:**

### Basic Concepts

1. **What is Frequency Modulation (FM)?**
   * **Answer**: Frequency Modulation (FM) is a modulation technique where the frequency of the carrier signal is varied in proportion to the amplitude of the message signal. This technique encodes the information into the frequency variations of the carrier wave.

### How does FM differ from Amplitude Modulation (AM)?

* + **Answer**: In FM, the carrier frequency varies according to the message signal, whereas in AM, the carrier amplitude varies. FM provides better noise immunity and a larger bandwidth compared to AM.

### Modulation Process

1. **What is the modulation index in FM, and how is it calculated?**
   * **Answer**: The modulation index (β) in FM is the ratio of the frequency deviation of the carrier signal to the frequency of the message signal. It is calculated as β=Δffm\beta =

\frac{\Delta f}{f\_m}β=fmΔf, where Δf\Delta fΔf is the peak frequency deviation and fmf\_mfm is the frequency of the message signal.

### How do you mathematically represent an FM signal?

* + **Answer**: An FM signal can be mathematically represented as s(t)=Ac⋅cos⁡(2πfct+2πΔffm∫0tm(τ) dτ)s(t) = A\_c \cdot \cos \left(2 \pi f\_c t + 2 \pi

\frac{\Delta f}{f\_m} \int\_{0}^{t} m(\tau) \, d\tau \right)s(t)=Ac⋅cos(2πfct+2πfmΔf∫0t m(τ)dτ), where AcA\_cAc is the carrier amplitude, fcf\_cfc is the carrier frequency, Δf\Delta fΔf is the peak frequency deviation, fmf\_mfm is the frequency of the message signal, and m(t)m(t)m(t) is the message signal.

### Demodulation Process

1. **What is the principle of frequency demodulation?**
   * **Answer**: Frequency demodulation involves recovering the original message signal from the modulated signal by detecting variations in the frequency of the received FM signal. Techniques such as the phase-locked loop (PLL) or differentiating and filtering can be used for demodulation.

### Explain the role of a Phase-Locked Loop (PLL) in FM demodulation.

* + **Answer**: A Phase-Locked Loop (PLL) is used in FM demodulation to synchronize the phase of a local oscillator with the phase of the received FM signal. By tracking the phase difference, the PLL extracts the frequency variations corresponding to the original message signal.

### Practical Considerations

1. **What are the advantages of FM over AM in terms of noise immunity?**
   * **Answer**: FM is more resistant to noise than AM because noise typically affects amplitude, whereas FM encodes information in frequency variations. Noise impacts the amplitude of the FM signal, which does not affect the frequency information, thus providing better signal quality and clarity.

### How does frequency deviation affect the bandwidth of an FM signal?

* + **Answer**: The bandwidth of an FM signal is proportional to the modulation index. As the frequency deviation increases, the bandwidth of the FM signal also increases. According to Carson's rule, the bandwidth BBB is approximately 2(Δf+fm)2(\Delta f + f\_m)2(Δf+fm), where Δf\Delta fΔf is the peak frequency deviation and fmf\_mfm is the highest frequency in the message signal.

### Scilab Implementation

1. **How can you simulate FM modulation in Scilab?**
   * **Answer**: To simulate FM modulation in Scilab, generate a carrier signal and modulate its frequency according to the message signal.

### How do you perform FM demodulation using a Phase-Locked Loop (PLL) in Scilab?

* + **Answer**: In Scilab, you can implement FM demodulation using a PLL by creating a PLL model that locks onto the frequency of the received FM signal and extracts the message signal.

## EXPT.NO.5 SIMULATION OF MEAN AND VARIANCE USING SCILAB

**AIM:**

To write a program for mean, variance and cross correlation in SCILAB and verify the output.

**EQUIPMENTS Needed**

* Computer with i3 Processor
* SCI LAB

### Algorithm

1. **Define the Function**: Specify the function you want to simulate. For example, f(x)=sin⁡(x)f(x) = \sin(x)f(x)=sin(x) or any other function.
2. **Generate Sample Points**: Decide on the range and the number of sample points. Generate these sample points within the desired range.
3. **Evaluate the Function**: Compute the function values at each of these sample points.
4. **Compute Mean, Variance and Cross Correlation:** Use Scilab's functions to calculate the mean and variance of the computed function values.
5. **Display Results**: Output the computed mean variance and **Cross Correlation PROCEDURE**
   * Refer Algorithms and write code for the experiment.
   * Open SCILAB in System
   * Type your code in New Editor
   * Save the file
   * Execute the code
   * If any Error, correct it in code and execute again
   * Verify the generated results

## PROGRAM

clear; clc; clear;

//Mean Value function X=f(x),

z=3\*(1-x)^2,//Marginal Probability Density Function X=x\*z

endfunction a=0;

b=1;

EX=intg(a,b,f);//Mean value of X function Y=c(y)

z=3\*(1-y)^2,//Marginal Probability Density Function Y=y\*z

endfunction EY=intg(a,b,c);//Mean value of Y disp(EX,"i)Mean of X =") disp(EY," Mean of Y =")

### Variance

function X=g(x),

z=3\*(1-x)^2,//Marginal Probability Density Function X=x^2\*z

endfunction a=0;

b=1;

EX2=intg(a,b,g); function Y=h(y)

z=3\*(1-y)^2,//Marginal Probability Density Function Y=y^2\*z

endfunction EY2=intg(a,b,h);

vX2=EX2-(EX)^2;//Variance of X vY2=EY2-(EY)^2;//Variance of Y disp(vX2,"ii)Variance of X"); disp(vY2," Variance of Y");

### Cross Correlation

x= input("type in the reference sequence="); y= input("type in the second sequence="); n1=max(size(y))-1;

n2=max(size(x))-1;

r=corr(x,y,n1);

plot2d3('gnn',r);

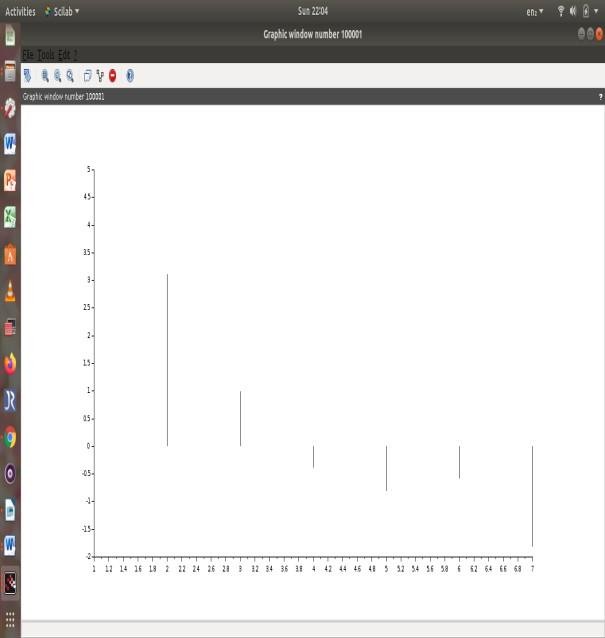
## OUTPUT

1. **Mean of X = 0.25 Mean of Y = 0.25**
2. **Variance of X 0.0375 Variance of Y 0.0375**

**Cross Correlation**

**Type in the reference sequence = [1 2 3 4 5 6 7 8]**

**Type in the second sequence = [2 1 3 5 6 3 5 9]**



## RESULT:

Thus the mean , variance and cross correlation are executed in Scilab and output is verified.

**Viva questions on mean and variance of a function using Scilab**

* 1. **What is the mean of a function's output, and how is it calculated in Scilab?**

**Answer:** The mean of a function's output is the average value of the function over a specified range or domain. To calculate it in Scilab, you first compute the function's values over a range of inputs and then calculate the mean of these values using the mean() function.

* 1. **How is the variance of a function's output defined, and how can it be computed in Scilab? Answer:** The variance of a function's output measures the spread of the function's values around the mean. It is computed as the average of the squared differences between each function value and the mean. In Scilab, variance can be computed using the variance() function:
  2. **Explain how you would simulate the mean and variance of the function f(x)=exf(x) = e^xf(x)=ex over the interval [0, 5] using Scilab.**

**Answer:** To simulate the mean and variance of f(x)=exf(x) = e^xf(x)=ex over the interval [0, 5], follow these steps:

* + - Define the range of xxx values.
    - Compute the function values f(x)f(x)f(x).
    - Calculate the mean and variance of the function values.
  1. **How would you estimate the mean and variance of a noisy function f(x)=sin⁡(x)+noisef(x) =**

**\sin(x) + \text{noise}f(x)=sin(x)+noise where noise is normally distributed?**

**Answer:** To estimate the mean and variance of a noisy function, you need to:

* + - Generate the noise, which is typically added to the function output.
    - Compute the function values with noise.
    - Calculate the mean and variance of these noisy function values

### What is the significance of Monte Carlo simulation in estimating the mean and variance of a function, and how can it be implemented in Scilab?

**Answer:** Monte Carlo simulation is used to estimate the mean and variance by repeatedly sampling from a probabilistic model. It involves generating multiple random samples and computing statistical metrics from these samples. In Scilab, you can implement it by generating random samples of inputs, evaluating the function, and then calculating the mean and variance of the results:

### How would you use Scilab to simulate and compare the mean and variance of two functions, such as sin⁡(x)\sin(x)sin(x) and cos⁡(x)\cos(x)cos(x), over the same interval?

**Answer:** To compare the mean and variance of two functions, simulate both functions over the same interval, compute their mean and variance, and then compare the results:

### What are the common pitfalls in simulating mean and variance of a function, and how can you avoid them using Scilab?

**Answer:** Common pitfalls include:

* + - **Insufficient Sample Size**: Leads to unreliable estimates. Ensure a large enough sample size.
    - **Numerical Precision**: Rounding errors can affect results. Use Scilab's high-precision functions and be cautious with floating-point arithmetic.
    - **Incorrect Function Evaluation**: Ensure that the function is correctly defined and evaluated. Verify the function implementation before analysis.

### What is the mean of a dataset, and how is it calculated in Scilab?

**Answer:** The mean of a dataset is the average of all the data points. It is calculated by summing all the values and dividing by the number of values. In Scilab, the mean can be calculated using the mean() function.

### How is variance defined, and how can it be calculated in Scilab?

**Answer:** Variance measures the dispersion of data points around the mean. It is the average of the squared differences between each data point and the mean. In Scilab, variance can be calculated using the variance() function

### Explain the difference between population variance and sample variance. How can you compute both in Scilab?

**Answer:** Population variance is calculated when the dataset includes the entire population, while sample variance is calculated from a sample of the population. Population variance divides by NNN (the number of observations), while sample variance divides by N−1N-1N−1 to account for degrees of freedom. In Scilab:

* + - **Population Variance**: variance(data, "population")
    - **Sample Variance**: variance(data, "sample")

## EXPT.NO.6 SIMULATION OF AUTOCORRELATION AND PSD USING SCILAB AIM:

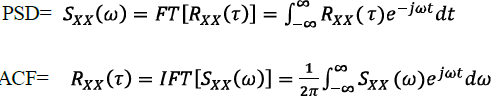
Write a program for Autocorrelation and PSD of signals in SCILAB and verify Wiener-Khinchin relation.

**EQUIPMENTS Needed**

* + - * Computer with i3 Processor
      * SCI LAB

**THEORY:**

The Wiener-Khinchin theorem states that the power spectral density of a wide sense stationary random process is the Fourier transform of the corresponding autocorrelation function.



**Algorithm**

1. **Load or Define the Signal**: Input your time-domain signal.
2. **Compute Autocorrelation**: Calculate the autocorrelation function of the signal.
3. **Compute Power Spectral Density (PSD)**: Estimate the PSD of the signal, either directly using a method like Welch’s periodogram or by using the Fourier transform of the autocorrelation.
4. **Plot Results**: Visualize the autocorrelation function and PSD.

## PROCEDURE

* + Refer Algorithms and write code for the experiment.
  + Open SCILAB in System
  + Type your code in New Editor
  + Save the file
  + Execute the code
  + If any Error, correct it in code and execute again
  + Verify the generated waveform using Tabulation and Model Waveform

**PROGRAM:**

clc

clear all; t=0:0.01:2\*pi;

x=sin(2\*t); subplot(3,2,1); plot(x); au=xcorr(x,x);

Subplot (3,2,2); plot (au); v=fft(au); subplot(3,2,3);

plot(abs(v)); fw=fft(x); subplot(3,2,4); plot(fw); fw2=(abs(fw)).^2;

subplot(3,2,5); plot(fw2);

## OUTPUT:

**RESULT:**

Thus the Autocorrelation and PSD are executed in Scilab and output is verified.

Viva questions:

1. What is autocorrelation?

Answer: Autocorrelation is a mathematical tool used to measure the similarity between a signal and a delayed version of itself over different time intervals. It helps in identifying repeating patterns or periodic signals within the data.

1. What is Power Spectral Density (PSD)?

Answer: Power Spectral Density (PSD) is a measure of the power distribution of a signal as a function of frequency. It indicates how the power of a signal is distributed across different frequency components.

1. How is autocorrelation useful in signal processing?

Answer: Autocorrelation is useful in signal processing for identifying periodicities, detecting repeating patterns, and estimating the signal's fundamental frequency. It is also used in noise reduction and system identification.

1. What is the relationship between autocorrelation and PSD?

Answer: The PSD of a signal is the Fourier transform of its autocorrelation function. This relationship, known as the Wiener-Khinchin theorem, implies that the frequency components of a signal can be analyzed by examining its autocorrelation function.

1. Why is Scilab used for simulating autocorrelation and PSD?

Answer: Scilab is used for simulating autocorrelation and PSD because it is a powerful open-source software with built-in functions for signal processing. It allows for easy computation and visualization of complex mathematical operations.

1. What Scilab function can be used to calculate the autocorrelation of a signal?

Answer: The Scilab function corr() can be used to compute the autocorrelation of a signal.

1. How can you compute the PSD of a signal in Scilab?

Answer: The PSD of a signal can be computed in Scilab using the fft() function to obtain the Fourier transform and then squaring the magnitude of the result. The spec() function can also be used for direct PSD computation.

1. What is the significance of the zero lag value in autocorrelation?

Answer: The zero lag value in autocorrelation represents the total power or energy of the signal. It is the maximum value of the autocorrelation function, indicating the similarity of the signal with itself at zero delay.

1. What is meant by the term "lag" in the context of autocorrelation?

Answer: In the context of autocorrelation, "lag" refers to the time shift or delay applied to the signal. The autocorrelation function is computed for different lags to assess how the signal correlates with itself at varying time delays.

1. How does windowing affect the computation of PSD?

Answer: Windowing affects the computation of PSD by reducing spectral leakage. When a signal is windowed before computing its Fourier transform, it reduces the discontinuities at the boundaries, which can cause distortions in the frequency domain representation. Different window functions, like Hamming or Hanning, can be applied to minimize this effect.

**EXPT.NO.7 Amplitude Modulation and Demodulation using NumPy and Matplotlib**

Aim

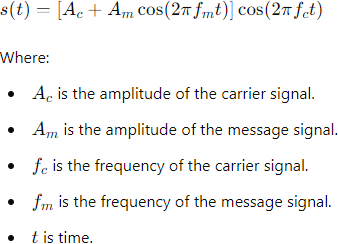
To implement and analyze amplitude modulation (AM) using Python's NumPy and Matplotlib libraries. Apparatus Required

**Software:** Python with NumPy and Matplotlib libraries

**Hardware:** Personal Computer

Theory

Amplitude Modulation (AM) is a technique used in electronic communication, primarily for transmitting information via a radio carrier wave. In AM, the amplitude of the carrier wave is varied in proportion to that of the message signal. The general form of an AM signal is:



Algorithm

1. **Initialize Parameters**: Set the values for carrier frequency, message frequency, and sampling frequency.
2. **Generate Time Axis**: Create a time vector for the signal duration.
3. **Generate Message Signal**: Define the message signal as a cosine wave.
4. **Generate Carrier Signal**: Define the carrier signal as a cosine wave.
5. **Modulate Signal**: Apply the AM formula to obtain the modulated signal.
6. **Plot the Signals**: Use Matplotlib to plot the message signal, carrier signal, and modulated signal.

Result

The message signal, carrier signal, and amplitude modulated (AM) signal will be displayed in separate plots. Thus AM is implemented using numPy and Matplotlib.

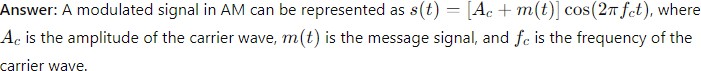
**Viva questions:**

1. What is amplitude modulation (AM)?

**Answer:** Amplitude Modulation (AM) is a modulation technique in which the amplitude of a carrier wave is varied in proportion to the instantaneous amplitude of the message signal. This technique is widely used in radio broadcasting.

1. What is a carrier wave in AM?

**Answer:** A carrier wave is a high-frequency sinusoidal signal that is modulated with the message signal in amplitude modulation. It serves as the base signal that carries the information.

1. How do you represent a modulated signal mathematically in AM?
2. What are the key components required to implement AM using NumPy and Matplotlib?

**Answer:** The key components required are:

* + NumPy for generating the message and carrier signals and performing mathematical operations.
  + Matplotlib for plotting the waveforms of the message signal, carrier signal, and modulated signal.

1. How do you generate a carrier wave using NumPy?

**Answer:** A carrier wave can be generated using NumPy by creating a time array and applying the cosine function:

import numpy as np

t = np.linspace(0, 1, 1000) # Time array fc = 10 # Carrier frequency

carrier = np.cos(2 \* np.pi \* fc \* t)

1. What is the modulation index in AM, and how is it calculated?

**Answer:** The modulation index (m) in AM is a measure of the extent of modulation and is defined as the ratio of the amplitude of the message signal (A\_m) to the amplitude of the carrier wave (A\_c). It is calculated as m=AmAcm = \frac{A\_m}{A\_c}m=AcAm.

1. What is over-modulation in AM, and why is it undesirable?

**Answer:** Over-modulation occurs when the modulation index m>1m > 1m>1. This means the message signal's amplitude exceeds the carrier amplitude, causing distortion and making it difficult to demodulate the signal accurately.

1. How can you visualize an AM waveform using Matplotlib?

**Answer:** An AM waveform can be visualized using Matplotlib by plotting the carrier signal, message signal, and modulated signal on the same plot or separate plots. Here’s a basic example:

import matplotlib.pyplot as plt plt.plot(t, modulated\_signal) plt.title('AM Waveform') plt.xlabel('Time') plt.ylabel('Amplitude') plt.show()

1. What is the purpose of using NumPy's np.cos() and np.sin() functions in AM simulation?

**Answer:** The functions np.cos() and np.sin() are used to generate sinusoidal waveforms for the carrier and message signals. These functions are essential in creating the periodic signals needed for AM simulation.

1. How can you adjust the modulation index in a NumPy implementation of AM?

**Answer:** The modulation index can be adjusted by changing the amplitude of the message signal or the carrier wave. In NumPy, this can be done by scaling the message signal array:

message\_amplitude = 2

carrier\_amplitude = 1

message\_signal = message\_amplitude \* np.cos(2 \* np.pi \* fm \* t) carrier\_signal = carrier\_amplitude \* np.cos(2 \* np.pi \* fc \* t)

modulated\_signal = (carrier\_amplitude + message\_signal) \* np.cos(2 \* np.pi \* fc \* t)

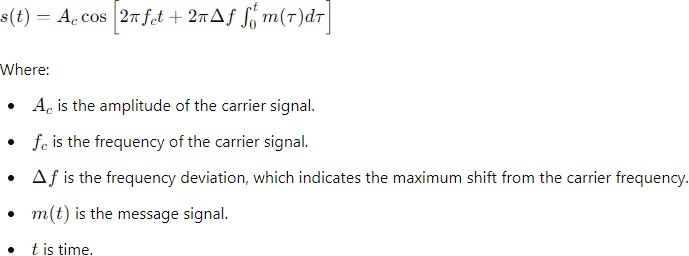
**EXPT.NO.8 Frequency Modulation and Demodulation using NumPy and Matplotlib**

Aim

To implement and analyze frequency modulation (FM) using Python's NumPy and Matplotlib libraries. Apparatus Required

1. **Software:** Python with NumPy and Matplotlib libraries
2. **Hardware:** Personal Computer Theory

Frequency Modulation (FM) is a method of transmitting information over a carrier wave by varying its frequency in accordance with the amplitude of the input signal (message signal). The frequency of the carrier wave is varied according to the instantaneous amplitude of the message signal. The general form of an FM signal is:



Algorithm

1. **Initialize Parameters**: Set the values for carrier frequency, message frequency, sampling frequency, and frequency deviation.
2. **Generate Time Axis**: Create a time vector for the signal duration.
3. **Generate Message Signal**: Define the message signal as a cosine wave.
4. **Compute the Integral of the Message Signal**: Calculate the integral of the message signal over time.
5. **Generate FM Signal**: Apply the FM modulation formula to obtain the modulated signal.
6. **Plot the Signals**: Use Matplotlib to plot the message signal, carrier signal, and modulated signal.

Result

The message signal, carrier signal, and frequency modulated (FM) signal will be displayed in separate plots. The modulated signal will show frequency variations corresponding to the amplitude of the message signal.

Viva Questions:

1. What is a carrier frequency in the context of FM?

Answer: The carrier frequency in FM is the frequency of the unmodulated carrier wave, which is varied in accordance with the amplitude of the input message signal.

1. What role does the modulation index play in FM?

Answer: In FM, the modulation index determines the extent of frequency deviation of the carrier signal from its center frequency due to the message signal. It is defined as the ratio of the frequency deviation to the frequency of the message signal.

1. What is frequency deviation in FM?

Answer: Frequency deviation refers to the maximum change in frequency from the carrier frequency due to the modulation by the message signal. It is a key parameter that affects the bandwidth and quality of the FM signal.

1. How is the instantaneous frequency of an FM signal defined?

Answer: The instantaneous frequency of an FM signal is the derivative of the phase of the signal with respect to time. It represents the frequency at any given moment and varies according to the modulating message signal.

1. What is the bandwidth of an FM signal, and how is it estimated?

Answer: The bandwidth of an FM signal is the range of frequencies occupied by the signal. It can be estimated using Carson's Rule, which states that the bandwidth is approximately equal to twice the sum of the maximum frequency deviation and the highest frequency of the modulating signal.

1. Why is the Nyquist sampling theorem important in digital FM implementation?

Answer: The Nyquist sampling theorem is important in digital FM implementation because it dictates the minimum sampling rate needed to accurately sample the signal without aliasing. For FM, the sampling rate should be at least twice the highest frequency present in the signal.

1. What is phase deviation in FM?

Answer: Phase deviation refers to the change in the phase of the carrier wave due to the modulating signal. In FM, the instantaneous phase deviation is proportional to the integral of the modulating signal.

1. How can NumPy be used to generate an FM signal?

Answer: NumPy can be used to generate an FM signal by creating a time array and computing the instantaneous phase of the carrier using the cumulative sum of the message signal. This phase is then used to modulate the carrier frequency:

import numpy as np

fc = 100 # Carrier frequency

kf = 10 # Frequency sensitivity

message = np.sin(2 \* np.pi \* 2 \* t) # Message signal instantaneous\_phase = 2 \* np.pi \* fc \* t + kf \* np.cumsum(message) fm\_signal = np.cos(instantaneous\_phase)

1. What is the significance of the baseband signal in FM?

**Answer:** The baseband signal in FM refers to the original message or information signal that modulates the carrier frequency. The characteristics of the baseband signal, such as amplitude and frequency content, directly influence the FM signal's characteristics.

1. How is Matplotlib used to visualize an FM signal?

**Answer:** Matplotlib is used to plot the time-domain representation of the FM signal, showing how the frequency of the carrier varies with time. It can also be used to plot the spectrogram or frequency spectrum of the FM signal to analyze its frequency components:

import matplotlib.pyplot as plt plt.plot(t, fm\_signal) plt.title('FM Signal') plt.xlabel('Time') plt.ylabel('Amplitude') plt.show()

### EXPT.NO.9 Phase Modulation using NumPy and Matplotlib

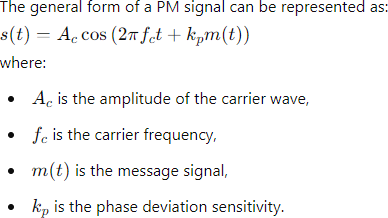
Aim

To implement and analyze phase modulation (PM) using Python's NumPy and Matplotlib libraries. Apparatus Required

1. **Software:** Python with NumPy and Matplotlib libraries
2. **Hardware:** Personal Computer Theory

Phase Modulation (PM) is a technique where the phase of the carrier wave is varied in proportion to the instantaneous amplitude of the input signal (message signal). Unlike frequency modulation, where the frequency is varied, in phase modulation, the phase angle of the carrier wave changes with the amplitude of the message signal.

The general form of a PM signal can be represented as:



Algorithm

### Initialize Parameters:

* + Set values for carrier amplitude (AcA\_cAc), carrier frequency (fcf\_cfc), message frequency (fmf\_mfm), sampling frequency, and phase deviation sensitivity (kpk\_pkp).

### Generate Time Axis:

* + Create a time vector for the signal duration based on the sampling frequency.

### Generate Message Signal:

* + Define the message signal as a cosine wave.

### Generate PM Signal:

* + Apply the PM modulation formula to obtain the modulated signal.

### Plot the Signals:

* + Use Matplotlib to plot the message signal, carrier signal, and phase-modulated signal.

Result

The message signal, carrier signal, and phase-modulated (PM) signal will be displayed in separate plots. The modulated signal will show phase variations corresponding to the amplitude of the message signal.

### Viva Questions:

1. What is a phasor, and how is it used in signal processing?

Answer: A phasor is a complex number representation of a sinusoidal function, expressing both the amplitude and phase of the waveform. It is used in signal processing to simplify the analysis of sinusoidal signals, especially when dealing with oscillations and rotations in the complex plane.

1. How does sampling rate affect the representation of a signal in digital signal processing?

Answer: The sampling rate, or sampling frequency, determines how often the continuous signal is sampled to create a discrete signal. According to the Nyquist theorem, the sampling rate must be at least twice the highest frequency present in the signal to avoid aliasing and accurately represent the signal in the digital domain.

1. What is the role of the np.angle() function in NumPy?

Answer: The np.angle() function in NumPy returns the phase angle (in radians) of the complex argument. It is useful for extracting the phase information from a complex number or a complex-valued signal.

1. Explain the concept of a complex envelope in modulation.

Answer: The complex envelope of a modulated signal is a representation that separates the amplitude and phase variations from the carrier frequency. It consists of the in-phase (I) and quadrature (Q) components, allowing for a more efficient analysis and processing of modulated signals, particularly in phase and frequency modulation.

1. What is the difference between phase modulation (PM) and frequency modulation (FM)?

Answer: In phase modulation (PM), the phase of the carrier signal is varied in proportion to the instantaneous amplitude of the message signal. In frequency modulation (FM), the frequency of the carrier signal is varied according to the message signal. While PM directly varies the phase, FM indirectly varies the phase through changes in frequency.

1. How do you visualize the phase of a signal over time using Matplotlib?

Answer: To visualize the phase of a signal over time using Matplotlib, you can plot the phase angle extracted from the signal. For a complex-valued signal, use the np.angle() function to get the phase and then plot it against time:

import matplotlib.pyplot as plt phase = np.angle(signal) plt.plot(time, phase)

plt.title('Phase of Signal Over Time')

plt.xlabel('Time') plt.ylabel('Phase (radians)') plt.show()

1. What is the significance of the term 'baseband' in signal processing?

Answer: The term 'baseband' refers to the original frequency range of a signal before modulation. It typically includes frequencies close to zero. Baseband signals are modulated onto a higher frequency carrier for transmission and are demodulated back to baseband at the receiver.

1. How can you implement a phase shift in a signal using NumPy?

Answer: A phase shift in a signal can be implemented using NumPy by adding a constant phase angle to the argument of the cosine or sine function. For example:

phase\_shift = np.pi / 4 # Phase shift in radians

shifted\_signal = np.cos(2 \* np.pi \* frequency \* t + phase\_shift)

1. What is the purpose of the np.unwrap() function in NumPy?

**Answer:** The np.unwrap() function in NumPy is used to correct phase angles by adding multiples of 2π when absolute jumps between consecutive angles are greater than π. This is useful for preventing discontinuities in phase plots that can occur due to the inherent periodicity of the phase angle.

1. Explain the importance of filtering in signal processing.

**Answer:** Filtering is crucial in signal processing for removing unwanted components from a signal, such as noise or interference, and for extracting specific frequency bands. Filters can be designed to pass certain frequencies (passbands) and attenuate others (stopbands), thereby improving the signal quality and making the data more interpretable.

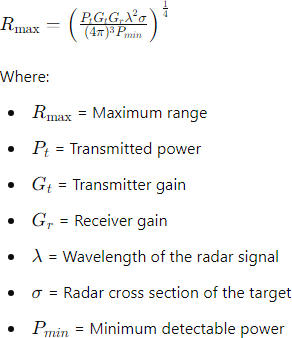
**EXP NO:10 EVALUATION OF RADAR RANGE USING PYTHON**

### Aim:

To calculate the maximum range of a radar system using the Radar Range Equation and verify the results through Python programming.

### Theory:

The Radar Range Equation is a fundamental formula used in radar system design to determine the maximum range at which a radar can detect a target. It is given by:



# Procedure

1. **Set Up the Python Environment**: Ensure that Python is installed on your system. You can use Anaconda for managing Python packages and environments, or any other Python IDE of your choice.
2. **Import Necessary Libraries**: Import the math library in Python.
3. **Define the Radar Range Equation Function**: Create a function to calculate the maximum range using the Radar Range Equation.
4. **Input Parameters for the Radar System**: Define the input parameters such as transmitted power, transmitter gain, receiver gain, radar frequency, radar cross section, and minimum detectable power.
5. **Calculate the Maximum Range**: Use the function to calculate the maximum range of the radar.
6. **Execute the Program**: Run the Python script to calculate and display the maximum range of the radar.

### Result:

Thus, the maximum range of a radar system using the Radar Range Equation is verified through a Python program.

### Viva Questions:

1. What is the importance of the antenna gain (G) in the radar range equation?

Answer: Antenna gain (G) represents the ability of the antenna to focus energy in a particular direction. Higher antenna gains result in more concentrated energy transmission and reception, which increases the radar's ability to detect targets at longer ranges.

1. What is radar cross-section (RCS)?

**Answer:** Radar Cross-Section (RCS) is a measure of how much power is scattered back towards the radar by a target. It is an important parameter in determining how detectable an object is by radar and is measured in square meters.

1. How is the wavelength (λ\lambdaλ) of the radar signal related to the frequency?

**Answer:** The wavelength (λ\lambdaλ) of the radar signal is inversely proportional to its frequency (f) and can be calculated using the formula: λ=cf\lambda = \frac{c}{f}λ=fc where ccc is the speed of light in a vacuum (≈3×108\approx 3 \times 10^8≈3×108 m/s).

1. What is the significance of the minimum detectable signal (SminS\_{\text{min}}Smin) in radar systems?

**Answer:** The minimum detectable signal (SminS\_{\text{min}}Smin) represents the smallest signal power level that the radar receiver can reliably detect. It depends on the sensitivity of the radar receiver and the noise level. A lower SminS\_{\text{min}}Smin allows the radar to detect weaker signals, thus increasing the maximum range.

1. Why are system losses (L) considered in the radar range equation?

**Answer:** System losses (L) account for the inefficiencies and losses in the radar system, such as those due to hardware components, signal attenuation, and atmospheric conditions. Including L in the radar range equation provides a more accurate estimation of the maximum range by accounting for these real-world factors.

1. What Python libraries can be used to evaluate the radar range?

**Answer:** Python libraries such as NumPy can be used for numerical calculations, SciPy for scientific computations, and Matplotlib for plotting results. These libraries are commonly used to implement and visualize the radar range equation.

1. How does increasing the transmitted power (Pt) affect the radar range?

Answer: Increasing the transmitted power (Pt) increases the maximum detectable range of the radar. This is because higher transmitted power results in a stronger return signal, making it easier for the radar to detect targets at greater distances.

1. What role does the speed of light play in radar systems?

**Answer:** The speed of light (c) is crucial in radar systems as it determines the wavelength of the transmitted signal when the frequency is known. It is also used to calculate the time taken for the radar signal to travel to the target and back, which is essential for determining the distance to the target.

1. How can atmospheric conditions affect radar range?

**Answer:** Atmospheric conditions such as humidity, temperature, and atmospheric pressure can affect the propagation of radar waves. They can cause signal attenuation, refraction, and absorption, which can reduce the effective range of the radar by weakening the transmitted and received signals.

1. What is the importance of the pulse repetition frequency (PRF) in radar systems?

**Answer:** The Pulse Repetition Frequency (PRF) is the rate at which successive pulses are transmitted by the radar. It is important because it determines the maximum unambiguous range of the radar and affects the radar's ability to distinguish between multiple targets. A higher PRF allows for faster updates but reduces the maximum unambiguous range.

**EXP NO:11 EVALUATION OF TIME DIVISION MULTIPLEXING**

**Aim:**

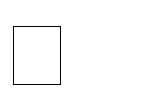
Study of TDM pulse amplitude modulation/ demodulation with transmitter block (clock) and channel identification information linked directly to the receivers.

**Apparatus Required:**

1. Experimental kit DCL-02
2. Connecting chord
3. Power supply
4. 20 MHz dual trace oscilloscope.

**Theory:**

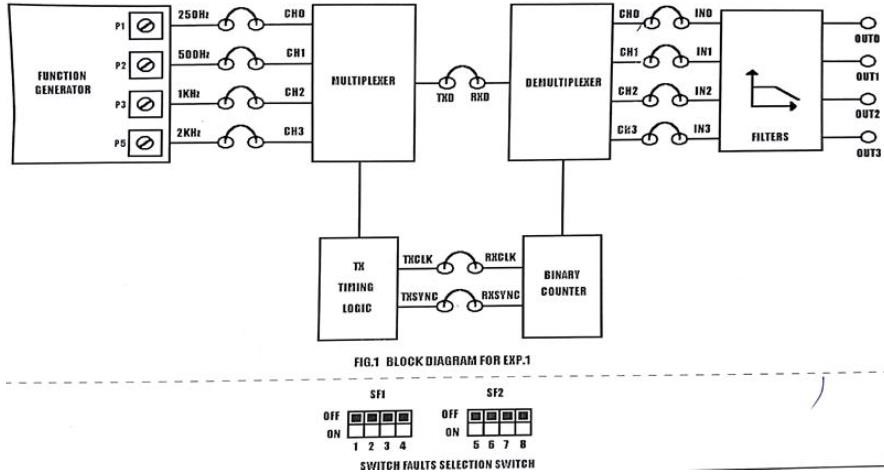
In PAM, PPM the pulse is present for a short duration and for most of the time between the two pulses no signal is present. This free space between the pulses can be occupied by pulses from other channels. This is known as Time Division Multiplexing. Thus, time division multiplexing makes maximum utilization 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. Thus the 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.

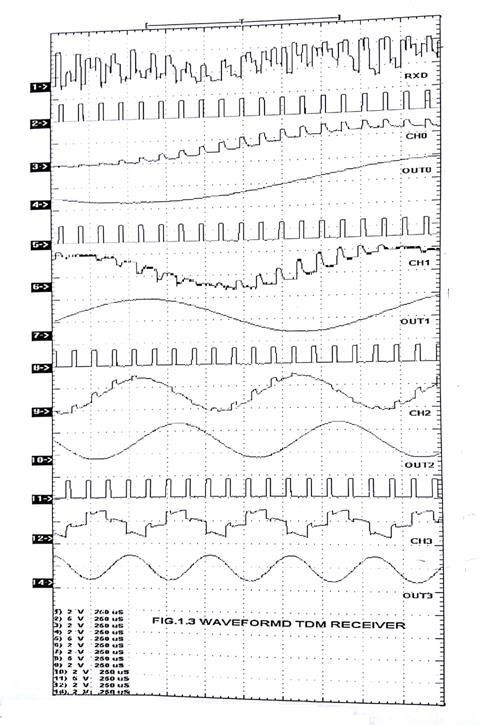
**Procedure:**

1. Refer to the block diagram and carry out the following connections and switch setting.
2. Connect power supply in proper polarity DCL-02 and s it switch it on.
3. Connect 250hz, 500hz, 1khz and 2khz sine wave signals from the function generator to the multiplexer input channels CH0, CH1,CH3 by means of connecting chords.
4. Connect the multiplexer output txd of the transmitted section to the de multiplexer input rxd of the receiver section.
5. Connect the output of the receiver section CH0, CH1, CH2,CH3 to the IN0, IN1, IN2. IN3 of the filter section.
6. Connect the sampling clock TX CLK channel identification clock TXSYNC of the transmitter section to the corresponding RXCLK, RXSYNC of the receiver section respectively.
7. Set the amplitude of the input sine wave desired.
8. Take the observations as mentioned below.

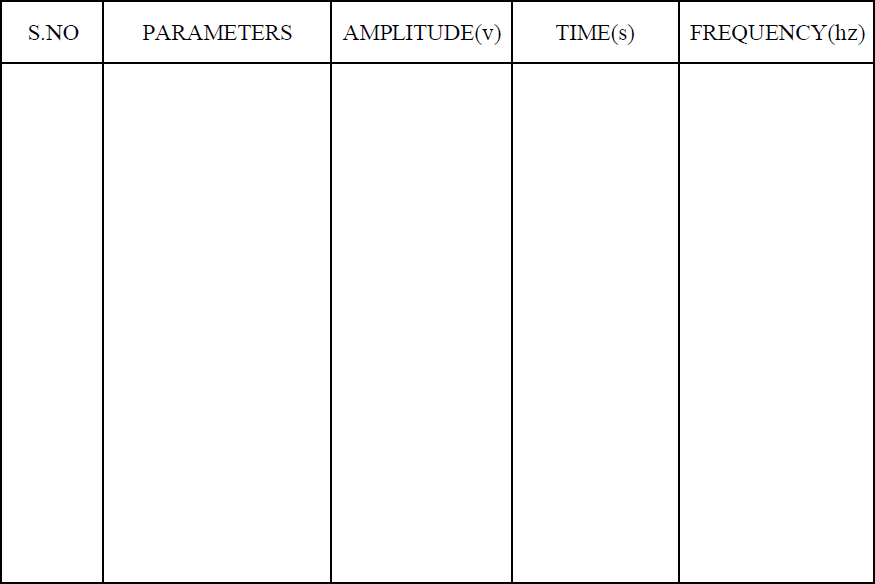
**Kit Diagram:**

****

**Model Graph:**



**Tabulation:**



**Result:**

Thus the time division multiplexing is done experimentally and output is verified.

Viva Questions:

1. What is Time Division Multiplexing (TDM)?

Answer: Time Division Multiplexing (TDM) is a technique in telecommunications that allows multiple signals to share the same communication channel by dividing the signal into time slots. Each signal is assigned a specific time slot in which it can transmit its data, ensuring that multiple signals can be transmitted simultaneously over the same medium without interference.

1. How does TDM differ from Frequency Division Multiplexing (FDM)?

Answer: TDM divides the communication channel into time slots, with each signal using the channel in its assigned time slot, while FDM divides the channel into frequency bands, with each signal using a different frequency band simultaneously. In TDM, time is the distinguishing factor, while in FDM, it is the frequency.

1. What are the two main types of TDM? Answer: The two main types of TDM are:

Synchronous TDM: Time slots are pre-assigned to each signal and occur at regular intervals, regardless of whether data is present.

Asynchronous (or Statistical) TDM: Time slots are dynamically allocated to signals based on demand, allowing for more efficient use of the channel.

1. What is the purpose of a multiplexer in TDM?

Answer: A multiplexer in TDM combines multiple input signals into a single output signal by assigning each input signal to a specific time slot. This allows multiple signals to be transmitted over a single communication channel.

1. What role does a demultiplexer play in TDM?

Answer: A demultiplexer separates the combined signal back into its original individual signals at the receiving end. It extracts each signal from its corresponding time slot and routes it to the appropriate receiver.

1. What is a frame in the context of TDM?

Answer: A frame in TDM is a complete cycle of time slots, with each slot allocated to a different input signal. A frame contains one time slot for each signal being multiplexed. In synchronous TDM, frames occur at regular intervals.

1. What is meant by "guard time" in TDM?

Answer: Guard time is a small period between time slots in TDM that prevents overlapping of signals and reduces the risk of interference between adjacent time slots. It ensures that the transition from one signal to another is smooth and distinct.

1. How does TDM improve the utilization of a communication channel?

Answer: TDM improves channel utilization by allowing multiple signals to share the same channel without interference. It divides the channel's capacity among multiple users, making efficient use of the available bandwidth and reducing the need for separate channels for each signal.

1. What is interleaving in the context of TDM?

Answer: Interleaving in TDM refers to the process of arranging multiple signals in a sequence of time slots. Each signal is assigned a specific slot within a frame, and the signals are interleaved based on their time slot assignments, allowing them to be transmitted in a regular and orderly fashion.

1. What are the common applications of TDM? Answer: Common applications of TDM include:

Telecommunications: For transmitting multiple telephone calls over a single communication line.

Data Communications: For multiplexing digital data streams, such as in digital subscriber line (DSL) services. Broadcasting: For multiplexing different television or radio channels over a single transmission medium.

Satellite Communication: For efficiently utilizing the available bandwidth by transmitting multiple signals over a single satellite link.