EXPERIMENT NUMBER : 6

EXPERIMENT NAME : QUADRATURE AMPLITUDE MODULATION (QAM)

DATE: 26/11/2022, SATURDAY

---

* **AIM:**

To perform Quadrature Amplitude Modulation (QAM) of the input digital signal and to verify the output.

* **SOFTWARE REQUIRED:**

① Oracle VM VirtualBox 6.1.38, Oracle Corporation

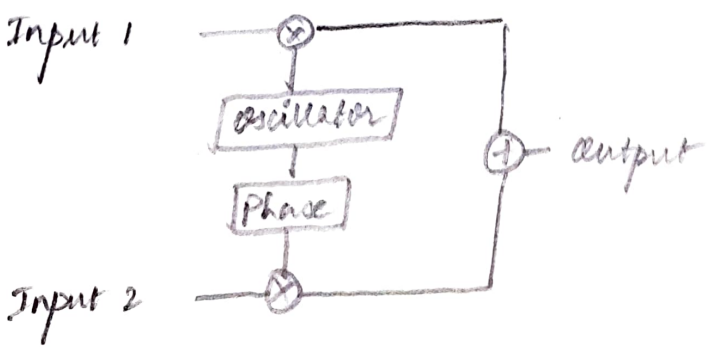
② Ubuntu 22.04 (64-bit) Operating system

③ GNU Radio Companion Application, v3.10.1
(sudo apt-get install gnuradio)

* **THEORY:**

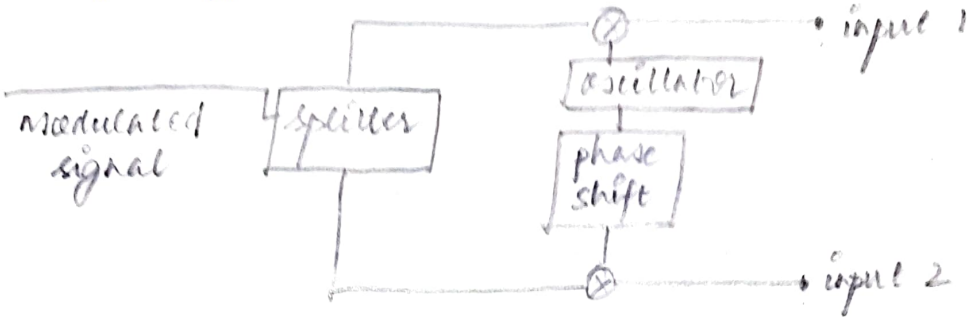
Quadrature Amplitude Modulation (QAM) is a form of modulation that is a combination of phase and amplitude modulation.

In modulation, there are two carrier signals with a phase shift of 90° between them. These are then amplitude modulated with two data stream known as in-phase and quadrature phase. These signals are added thus converting them to required frequency and amplifying them.

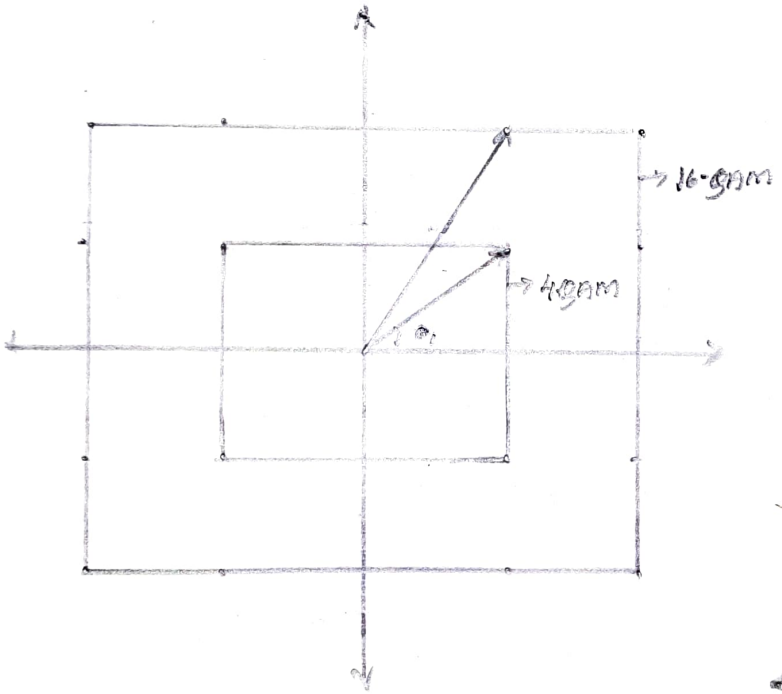


PTO

During modulation, the modulated signal is split and each side is applied to a mixer. One half has impulse local oscillator applied, then the another half has quadrature oscillator signal applied.



Since it includes both amplitude and phase modulation, we obtain a grid called constellation map.



Each vector changes in amplitude and phase. The signal waveform may be expressed as -

$$S_m(t) = Re\left[(A_{mi} + jA_{mq})\, g(t)\, e^{j2\pi f_c t}\right]$$

$$= A_{mi}\, g(t) \cos 2\pi f_c t - A_{mq}\, g(t) \sin 2\pi f_c t \;;$$

$$m = 1, 2, \ldots, M$$

PTO

QAM signal waveform can be expressed as -

$$s_m(t) = Re\left[s_m e^{j\theta_m} e^{j2\pi f_c t}\right]$$

$$= s_m \cos(2\pi f_c t + \theta_m)$$

where $s_m = \sqrt{A_{mi}^2 + A_{mq}^2}$ and $\theta_m = \tan^{-1}\left(\frac{A_{mq}}{A_{mi}}\right)$

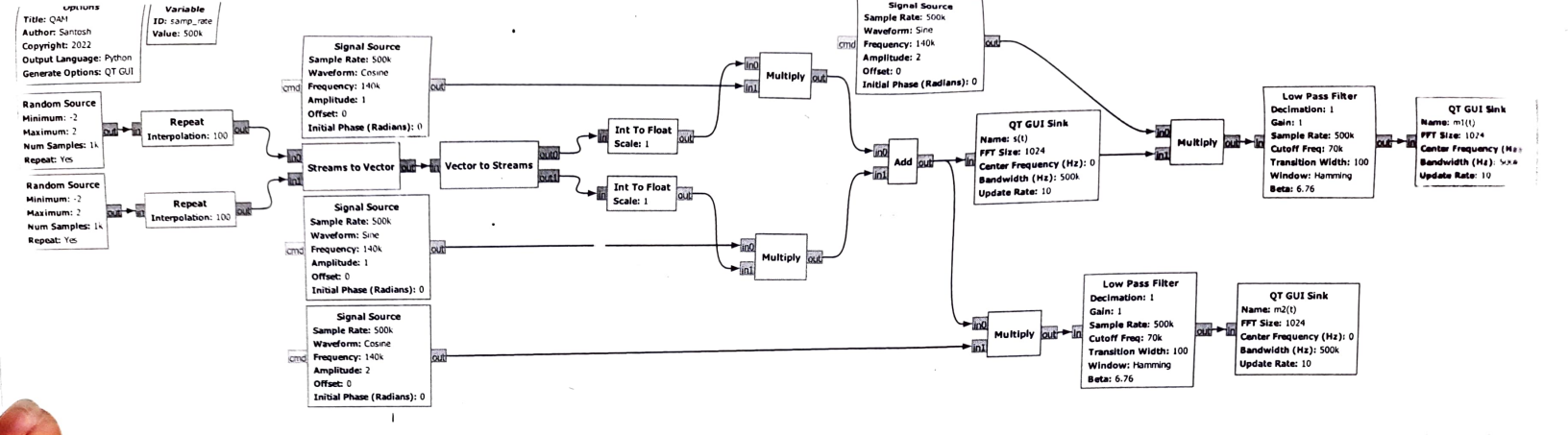
Euclidean distance, $d_{mn} = \sqrt{\|s_m - s_n\|^2}$

$$= \sqrt{\frac{\varepsilon_q}{2}\left[(A_{mi} - A_{ni})^2 + (A_{mq} - A_{nq})^2\right]}$$

Distance between adjacent points, $d_{mn} = \sqrt{2}g$.

* RESULT:

Quadrature Amplitude modulation (QAM) has been performed for digital signal and signal added with noise. All the simulation results were verified successfully.


Options
Title: QAM
Author: Santosh
Copyright: 2022
Output Language: Python
Generate Options: QT GUI
Variable
ID: samp_rate
Value: 500k
Random Source
Minimum: -2
Maximum: 2
Num Samples: 1k
Repeat: Yes
Random Source
Minimum: -2
Maximum: 2
Num Samples: 1k
Repeat: Yes
Repeat
Interpolation: 100
Repeat
Interpolation: 100
Signal Source
Sample Rate: 500k
Waveform: Cosine
Frequency: 140k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0
Streams to Vector
Vector to Streams
Signal Source
Sample Rate: 500k
Waveform: Sine
Frequency: 140k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0
Signal Source
Sample Rate: 500k
Waveform: Cosine
Frequency: 140k
Amplitude: 2
Offset: 0
Initial Phase (Radians): 0
Int To Float
Scale: 1
Int To Float
Scale: 1
Multiply
Multiply
Signal Source
Sample Rate: 500k
Waveform: Sine
Frequency: 140k
Amplitude: 2
Offset: 0
Initial Phase (Radians): 0
Add
QT GUI Sink
Name: s(t)
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 500k
Update Rate: 10
Multiply
Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 500k
Cutoff Freq: 70k
Transition Width: 100
Window: Hamming
Beta: 6.76
QT GUI Sink
Name: m1(t)
FFT Size: 1024
Center Frequency (Hz)
Bandwidth (Hz): 500k
Update Rate: 10
Multiply
Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 500k
Cutoff Freq: 70k
Transition Width: 100
Window: Hamming
Beta: 6.76
QT GUI Sink
Name: m2(t)
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 500k
Update Rate: 10

OUTPUTS :

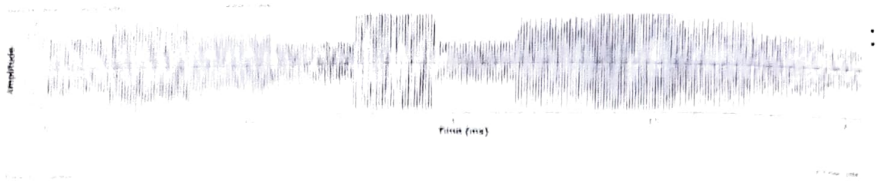


Figure 1 - Quadrature Amplitude modulated Signal, s(t)

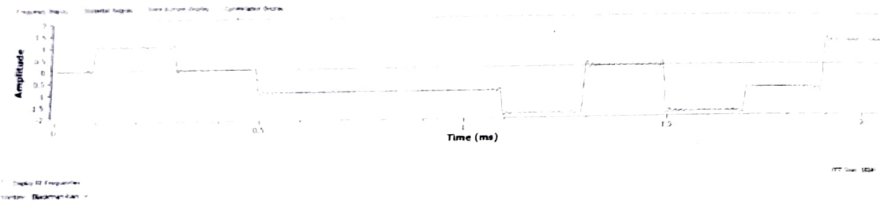


Figure 2 - Demodulated Signal, $m_1(t)$

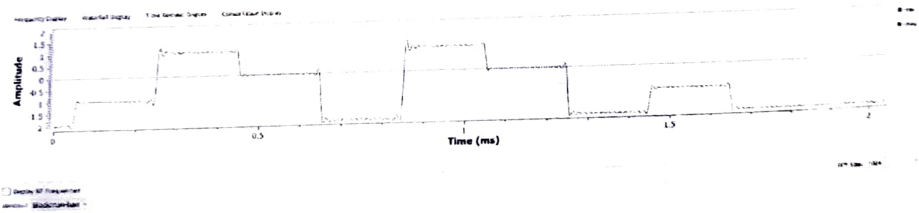


Figure 3 - Demodulated Signal, $m_2(t)$