

## Lab 2: Implementation of AM in GNU Radio



# Aim of the experiment

- To understand the basics of analog communication
- Implementing modulation – demodulation flow graphs for both amplitude modulation (AM) in GNU Radio



# Important Note

## ✓ Using QT GUI Tab Widget to monitor multiple signals:-

QT GUI Tab Widget block in gnuradio can be used to monitor multiple graphs by giving a label to each graph to overcome the confusion about the source of a graph. For example, if you want to monitor two signals in the time domain, namely modulated and demodulated, then use two QT GUI time sinks e.g., sink1 and sink2.

- 1 Open the block 'QT GUI Tab Widget'
- 2 Set the ID to 'tab' or any name you want to use
- 3 Set no. of tabs to 2 (the number of labeled graphs you want to monitor)
- 4 In Label 0 enter the label for first graph which is 'Modulated' in our example
- 5 In Label 1 enter the label for second graph which is 'Demodulated' in our example
- 6 Now to connect the GUI sinks to QT GUI Tab Widget, open the block of respective QT GUI Time sink (in the current example)
- 7 In GUI hint field of QT GUI Time sink 1 which is plotting modulated signal enter tab@0 ('ID of GUI Tab Widget'@'Index of tab')
- 8 In GUI hint field of QT GUI Time sink 2 which is plotting demodulated signal enter tab@1 ('ID of GUI Tab Widget'@'Index of tab')



# Important Note

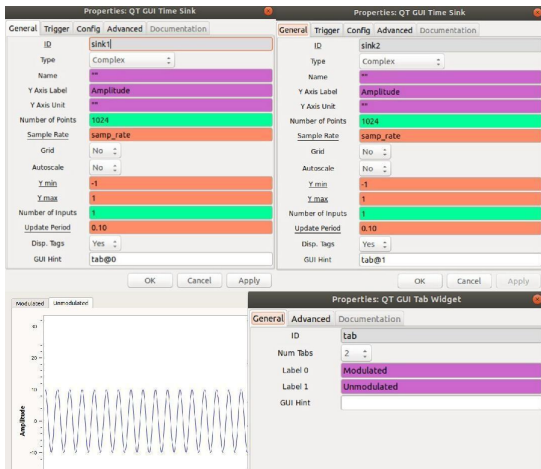


Figure: Using QT GUI Tab Widget to monitor multiple signals



# Important Note

- Use the sample rate of 48KHz for all un-modulated signals (this sample rate is termed as *Audio Rate* in FM blocks-however, you don't have to use the ready made blocks)
- Use the sample rate of 960kHz for all the frequency or phase modulated signals in GNU-Radio (this rate is termed as *Quadrature Rate* in GNU radio FM blocks)
- Debugging steps:
  - If something is not working, trace the point of failure(by checking the signal at various nodes)
  - If you're not able to get the display after a new GNU-Radio block was added in the schematic,most likely you have entered wrong parameters in the new block (check carefully!)
  - **Make sure that you are consistently accounting for the sample rate whenever decimation (for downsampling) and interpolation (for upsampling) are used.**
- IIR filter block implementation:
  - FF coefficients= $[b_0, b_1]$ ; FB coefficients= $[a_0, a_1]$ ; Old Style of Taps=True, implements the discrete time filter:
$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{a_0 - a_1 z^{-1}}$$
  - A bug in implementation always sets the value of  $a_0 = 1$ . Therefore, you must use  $a_0 = 1$  in all your calculations for filter coefficients.



# Task 1: Implementation of DSB-FC

- Implement an entire DSB-FC AM modulation-demodulation flow graph in GNU radio. Use an envelope detector for the demodulation.

*Note: You cannot use the built-in blocks for demodulation like AM-demod*

- Parameters to be used:
  - ◇ Message signal (single-tone): 10 kHz
  - ◇ Carrier signal: 100 kHz
- You are allowed to use the ready-to-use Low Pass Filter block from GNU radio library
- Observe the message signal and modulated signal for DSB-FC in time and frequency domains
- Replace the tone message signal with an audio message. Can you recover the message post-demodulation?



## Task 2: Implementation of DSB-SC

You now have to implement DSB-SC modulation and demodulation. Assume perfect carrier synchronisation at the receiver.

- 1 Implement the modulation flowgraph for DSB-SC transmission. Use a single tone at 10kHz as the message, and 100kHz as the carrier frequency.
- 2 Observe the spectrum of the modulated signal.
- 3 Implement the demodulation flowgraph.
- 4 Observe the spectrum of the demodulated message signal.
- 5 Replace the tone message signal with an audio message. Can you recover the message post-demodulation?

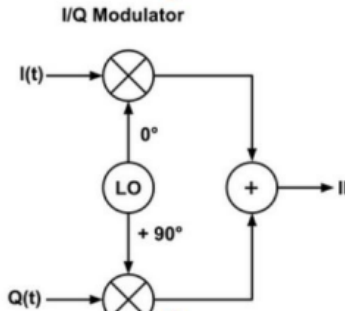
*Q: What is the impact of a carrier frequency mismatch between the transmitter and receiver? How about a phase mismatch?*



# Task 3: IQ modulation and Demodulation

## Part1: IQ Modulation

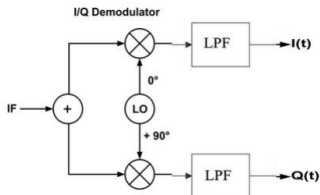
- 1 Implement the IQ modulation using two music files given (background.wav & vocal.wav). Treat one as the in-phase signal  $I(t)$  and the other as the quadrature-phase signal  $Q(t)$ .
- 2 Set the carrier frequency as 100KHz
- 3 Store the modulated signal in data1.dat file in file sink





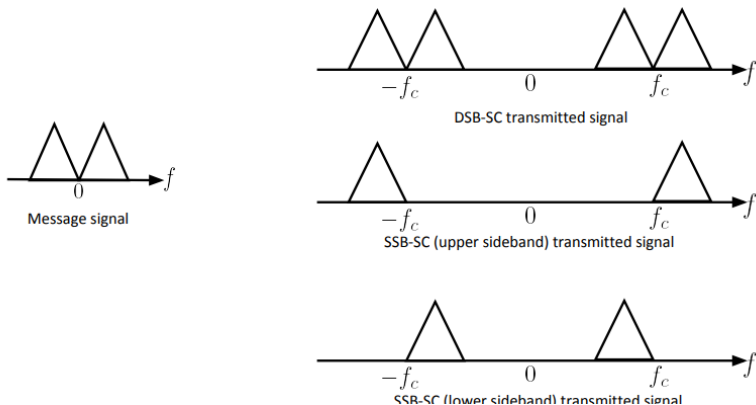
## Part2: IQ Demodulation

- 1 Demodulate the IQ modulated signal as shown in the block diagram, from data stored in data1.dat
- 2 Use a low pass filter to filter higher unwanted frequencies
- 3 Play the demodulated audio files. Observe if you get a faithful reconstruction.



## Task 4: Implementation of SSB

You have already implemented DSB-SC modulation. Here, only one sideband of the message is transmitted. For real messages, note that this is sufficient. This requires only half the bandwidth of DSB transmission!



- 1 Implement the modulation flowgraph for SSB-SC (upper sideband) transmission. Use a single tone at 10kHz as the message, and 200kHz as the carrier frequency.

*Hint: Think of the transmitted (passband) signal as*

$$\begin{aligned}s_p(t) &= \text{Re}([s_I(t) + js_Q(t)]e^{j2\pi f_c t}) \\ &= s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)\end{aligned}$$

What should  $s_I$  and  $s_Q$  be for SSB transmission?

- 2 Observe the spectrum of the modulated signal.
- 3 Implement the demodulation flowgraph.
- 4 Observe the spectrum of the demodulated message signal.
- 5 Replace the tone message signal with an audio message. Can you recover the message post-demodulation?
- 6 Repeat the above to achieve lower sideband transmission.

