EE 340: Communications Laboratory
Autumn 2021-22

Lab 2: Implementation of Analog Modulation schemes in GNU Radio

Aim of the experiment

- To understand the basics of analog communication (i.e., amplitude and frequency modulation schemes)
- Implementing modulation – demodulation flow graphs for both amplitude modulation (AM) and frequency modulation (FM) 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 source of a graph. For example, if you want to monitor two signals in 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.

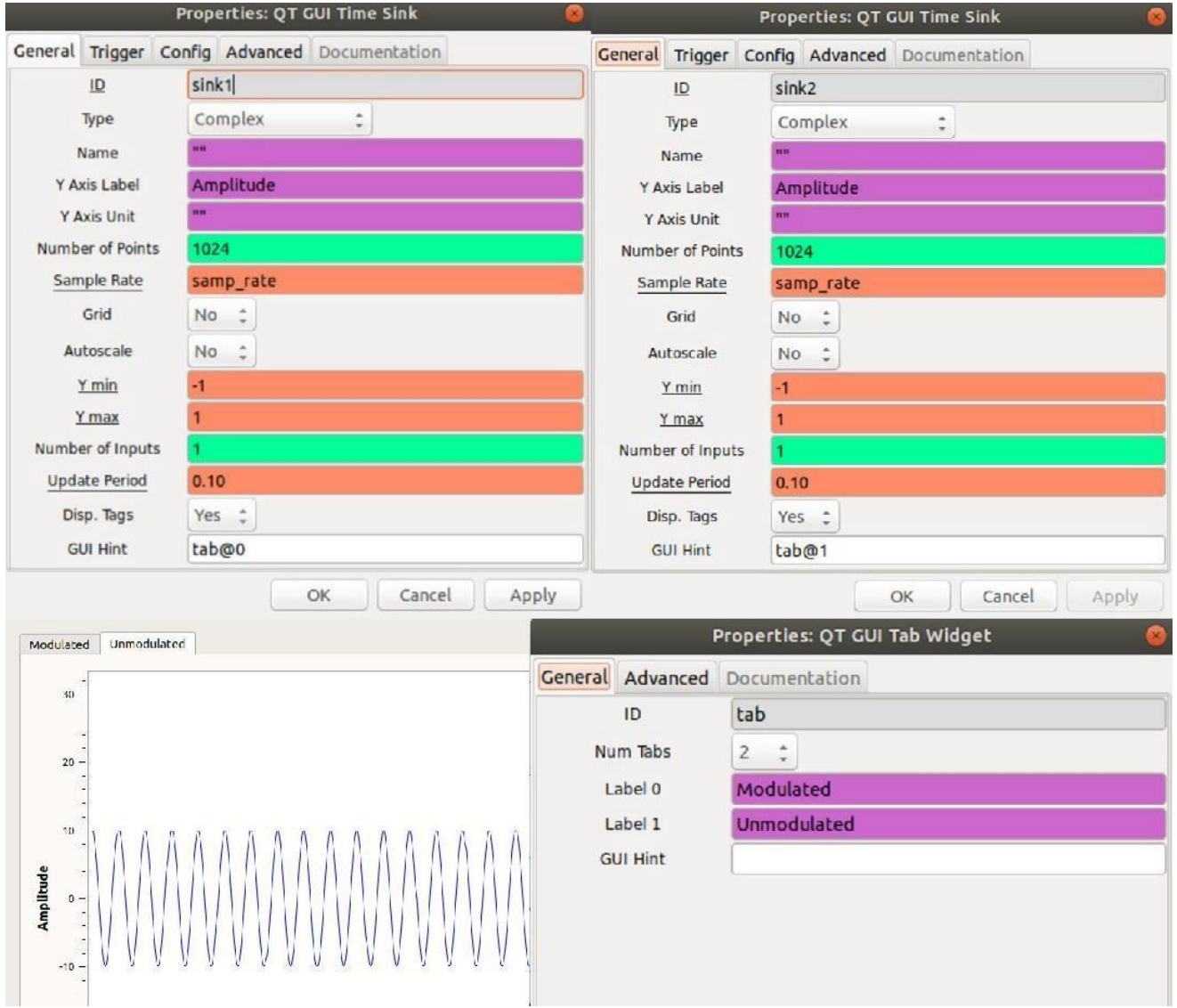
6. Now to connect the GUI sinks to QT GUI Tab Widget open the block of respective QT GUI Time sink (in the current example).

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

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 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 are not able to get the display after a new GNU-Radio block was added in the schematic, most likely you've 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 the 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.

  Note: The modulation part of your flowgraph should take the message signal as input and generate the transmitted (AM modulated) signal as output. The demodulation part should take the transmitted signal as input and generate the recovered message signal as output.

  Naturally, you cannot use the built-in blocks for modulation/demodulation (like AM-demod). You are, however, allowed to use the ready-to-use Low Pass Filter block from the GNU Radio library.

  DSB-FC has an additional DC component, which can be removed by the 'DC Blocker' block in GNU radio.

- Parameters to be used:
  - Message signal (single-tone): 10 kHz
  - Carrier signal: 100 kHz
- Observe the message signal, the modulated signal, and the recovered signal in time and frequency domains

Task 2: Demodulation of AM

- Perform DSB-FC demodulation on the transmitted signal provided to you as ‘lab1_task2.dat’

  The transmitted signal is provided to you at a sampling rate of 960 kHz, and the carrier frequency is 100 kHz. The message signal is an audio clip, containing frequency components ranging from 20Hz to 15kHz.

  Note: There might be a carrier phase offset between the transmitted and (your) receiver.

- Feed the de-modulated signal to the Audio Sink block. If the demodulation is done properly, you should be able to listen to the transmitted music signal.

Task 3: Implementation of a Frequency Modulator

- For making a Frequency Modulator, you need to first integrate the signal and then add the resultant signal to the phase of the carrier wave.
- To implement an integrator, use the IIR filter with: FF coefficients = [b_0]; FB coefficients = [1, 1]; Old Style of Taps = TRUE; Choose the sample rates judiciously, i.e., the Nyquist criterion should be satisfied comfortably
  - Choose b_0 =T, i.e., the sampling period of the signal
- This output should go to the phase modulator: For an input ϕ, the phase modulator outputs $e^{jk\Phi}$, where k_p is the phase modulator sensitivity.
  - What should be the sensitivity for achieving modulation index =1 (it should be of order 1)? Remember that you've already scaled the signal using b_0
  - Observe the modulated spectrum

Task 4: Making your own FM demodulator flow-graph

To demodulate FM signals, you need to find the phase of the incoming sample and differentiate it with respect to time to obtain the transmitted signal.

- Get the phase of the incoming signal: You can use ‘Complex to Arg’ block for this operation
- Take the difference between the arguments of nth and (n-1)th samples to obtain the demodulated message signal.
  Note: Try performing differentiation with complex signal only. (Why?)
- You can use the ‘Low Pass Filter’ block after demodulation to filter out the out-of-band noise and down-sample (using decimation value) the signal to 32 kHz (audio card sample rate). What would be the decimation factor?
- Observe the demodulated spectrum and listen to the Audio.

The above steps have to be carried out for the signal generated by your own FM modulator

Optional: Demodulate the signal given in file ‘lab1_task4.dat’, sampled at 960k. Message signal bandwidth 20 Hz to 15kHz.