EE 340: Communications Laboratory
Spring 2021

Lab 6: Implementation of Digital Modulation Schemes in GNU Radio

Legends

Question/Observation: Show it to the TA and explain (carries marks)

Recall/think about something

Caution

Additional information - weblink

Aim of the experiment

To understand the basics of digital communication systems using GNU Radio

- Understanding of different building blocks that are commonly used in a wireless digital communication system viz. a transmitter and a receiver.
- Transmitter: You will be using the standard building blocks available in GNU Radio and customize them (by setting a few parameters) to obtain a desired output.
- Receiver: Again you will be using the standard building blocks and customizing them to receive a QPSK signal (sent by USRP transmitter set up by us).
- Overall, this experiment should help you in understanding the bigger picture. Explicit details about implementation of the blocks will be learnt at a later stage in this course and in the theory course(s).

Pre-lab Work

- The prelab reading material must have helped you in the 'bigger picture' of digital communications.
- if you are interested in more details, any standard digital communication book can be referred to, such as:
  - Simon Haykin, Communication Systems.
  - John Proakis, Digital Communication.

Basic setup in GNU Radio

Using the import block, import the following items to use the constant pi, to use certain digital communication blocks and to use the filter design utility in your flowgraphs:

Import	ID
from math import pi	pi
from gnuradio import digital	digital
from gnuradio import filter	filt

Also Note:

Whenever you have to get the waveform corresponding to a symbol, use 'Polyphase Arbitrary Resampler' with desired SPS (samples-per-symbol'). Why?

Whenever you wish to observe constellation points (for example, from the 'Polyphase Clock Sync') use the output SPS = 1. Why?

Lab Task 1: Local generation of a signal with QPSK modulation(5 marks)

In this part, you will be generating a signal with a digital modulation format such as QPSK and 8-PSK. Basically, the entire transmitter flowgraph has to be put together.

Instructions:

- This experiment is to be performed using ready-made blocks in GNU Radio.
- For each block, after adding the block, check its output constellation, before proceeding further (to get a feel for the signals that are present at various points in the system).

Lab Task 1 steps

Use following variables in your transmitter flow graph:

The input and output sample rates in different blocks may be different – sample rate is not fixed, but symbol rate is constant across the blocks.

ID	Value
samp_rate	Will vary for different blocks (start with a symbol rate of 20 kSymbols/s)
sps	≥ 2 (say 3 or 4 for a baseband signal): samples/symbol for the waveform – must satisfy the Nyquist criterion.
excess_bw	0.35 < n <0.5 (defines the `roll-off factor' for the pulse shaping filter, select a typical value, and take it for granted).
const_points	4 (number of symbols in the contellation, i.e 4 for QPSK)
nfilts	32
ntaps	11*nfilts*sps
tx_taps	firdes.root_raised_cosine(nfilts,nfilts,1.0,excess_bw,ntaps)

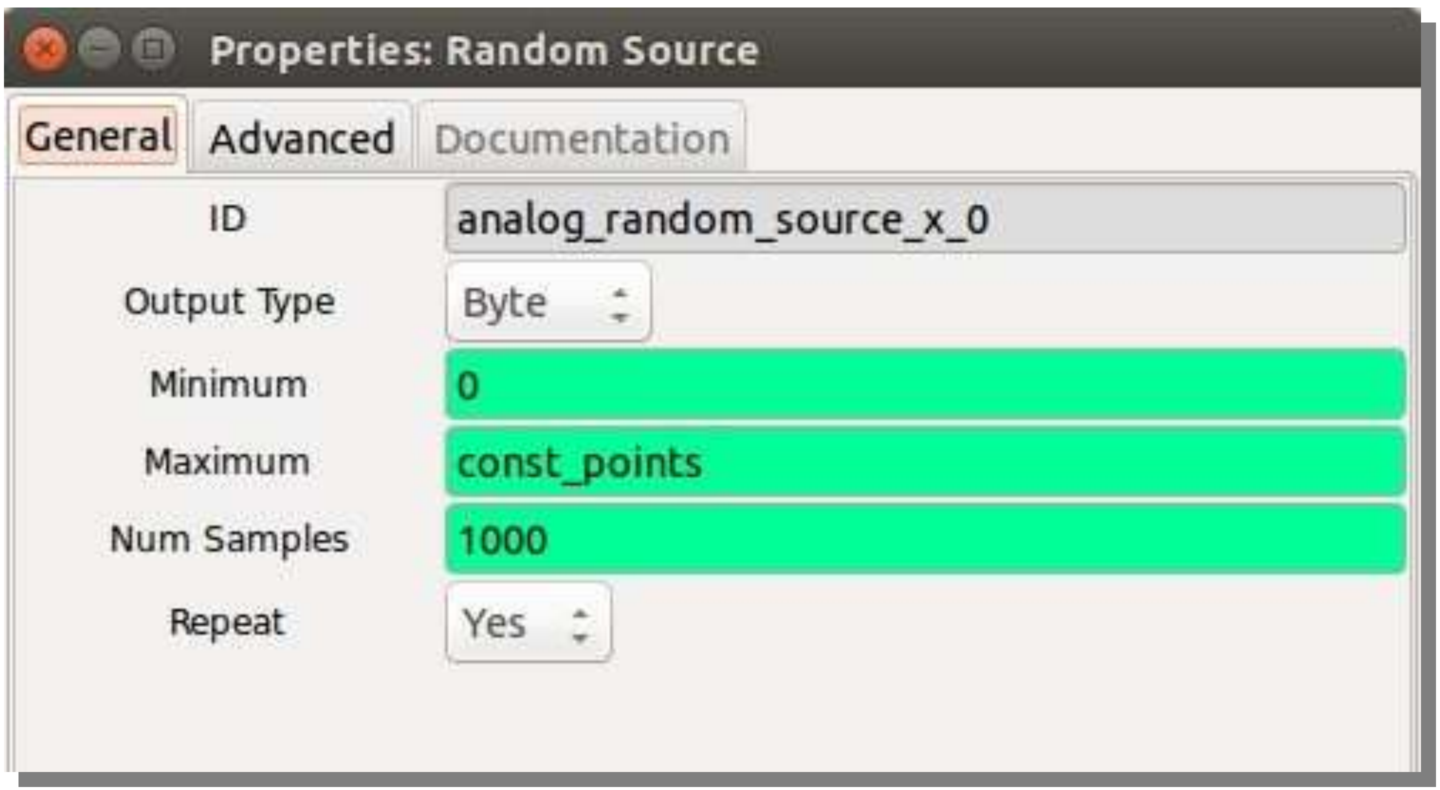
These parameters are used for defining the FIR pulse shaping filter tap coefficients. Take them for granted – its beyond the scope of this course to discuss how they are synthesised.

Important: `tx_taps' are the Root Raised Cosine FIR filter tap coefficients generated by the FIR design utility. Use could you the same coefficients in the receiver blocks (if the SPS is same) for the matched pulse shaping filter.

Building the Transmitter

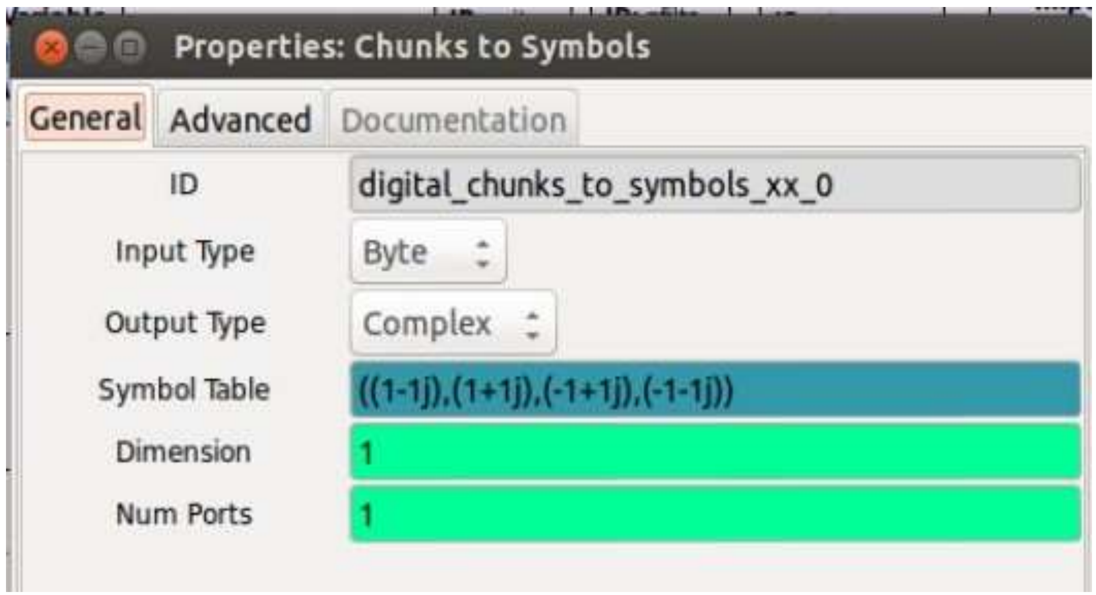
Use following blocks (in the correct sequence to build the transmitter section:

- Random Source: for generating symbols randomly (integers of values 0 to const_points-1)



Contd.

- Chunks to symbols: Maps the symbol values to the contellation points in the complex plane (here QPSK)

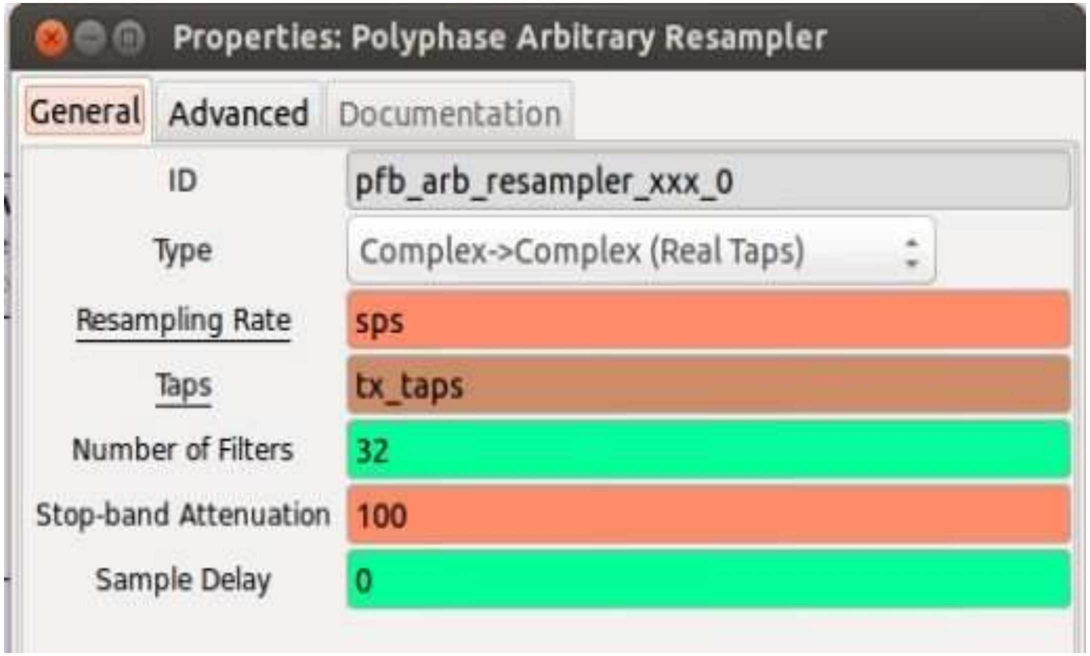


Observe the output in x-y plot in scope sink for the QPSK constellation. Also observe the samples in time domain try to make sense out of it.

What would you observe in the FFT sink?

Contd.

- Polyphase Arbitrary Resampler:



Observe the time domain waveforms of the output and the same output in the x-y graph in the scope. This is basically the pulseshaping filter output!), and see constellation is no more clustered in the X-Y plot. Why? State the difference between spectra before and after this block.

The obtained signal is basically your complex baseband QPSK modulated signal!

Upconversion to a carrier frequency

- Upconvert the baseband signal to a 100 kHz carrier by multiplying it with a complex ~100 kHz and obtaining its real part.

  Before upconversion, you will have to change the sample rate to say 500 kHz (to conveniently satisfy Nyquist criterion for the resultant signal) – you can make use of Rational Arbitrary Resampler for this purpose.

- You can save the resultant **real** signal to a file using File Sink block (it basically represents a practical signal that one can actually transmit from the antenna). Don't run the code for too long, otherwise the file will become too big.

Lab Task 2: Demodulation of a local QPSK signal (4 marks)

Instructions:

- Access the stored QPSK modulated wave using a File Source.
- After adding each new block, check its output using Scope Sink and FFT Sink block, and make sure that the observed signals make sense.

✓ No Need to use FLL band edge and CMA equalizer in this task, Why ?

Lab Task 2 steps

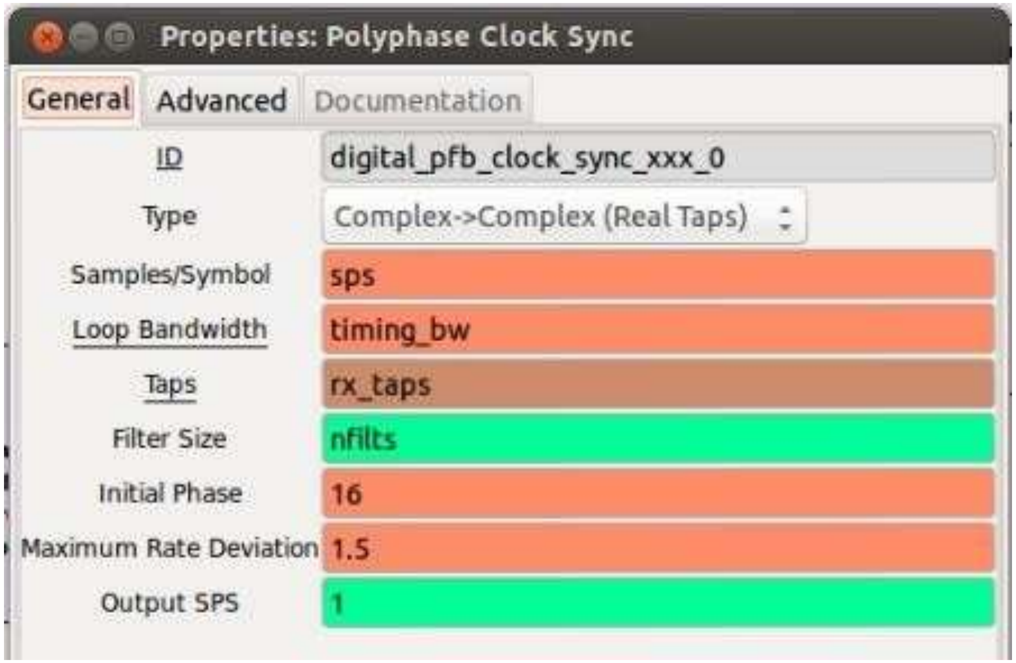
Variables: sps, samp_rate, excess_bw, nfilts, ntaps, import blocks are same as described in transmitter step.

Use following variable values in your receiver flow graph for defining your receiver blocks (listed in the next few slides)

ID	Value
freq_bw	2*pi/100 (normalised loop bandwidth for Costas loop)
fll_ntaps	55 (number of taps used in FLL filter)
timing_bw	2*pi/100 (normalised loop bandwidth for Polyphase Clock Synchronizer)
rx_taps	filter.firdes.root_raised_cosine(nfilts, nfilts*sps, 1.0, excess_bw, ntaps)

Building the Receiver

- Down-convert the 100 kHz modulated signal to complex baseband signal (you must be knowing by now how to do that).
- Polyphase Clock Sync:



Observe the constellation in the scope sink (x-y) and state the difference between spectrums before and after this block.

- If you are not getting the desired output you may try using a costas loop block after polyphase clock sync.

Lab Task 3a: Demodulating QPSK Signal Transmitted with frequency and phase offset and inside multipath enviroment

Instructions: Read the transmitted signal with frequency and phase offsets from the file input2.bin using file source block and try using the demodulator flowgraph built in task 2 to demodulate it. Use **4** as the **sps** value

- Is the message signal being demodulated correctly ? If not, Why ?

Lab Task 3b: Demodulating QPSK Signal Transmitted with frequency offset and inside multipath enviroment

Instructions: Note that the QPSK tx-rx pair we made in task 1 and 2 was for ideal signals. For more realistic (non-ideal) signals, some extra blocks will be required - there are frequency/phase offsets, multipath enviroment propagations and amplitude fluctuations.

Use your receiver built in task 2 as a starting point.

- Carrier freq=(f_c – frequency shift)

- Value of fc used is 100Khz with some frequency offset. The transmitted signal symbol rate is 25k Symbols/s with 500k samples per second sampling rate. The modulating signal was upsampled using a rational resampler with an interpolation factor of 5. Choose appropriate sps, sample rates etc. Observe the spectrum of output and **calculate the frequency shift**.

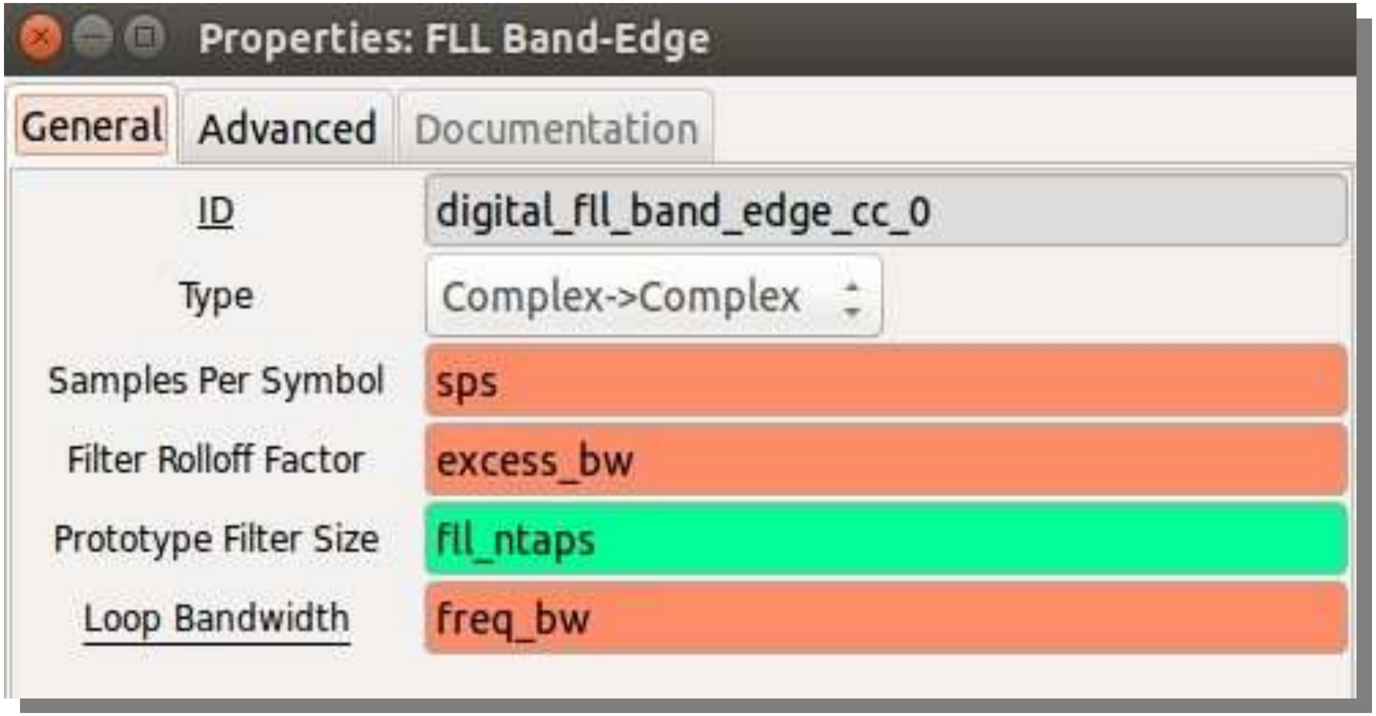
- To obtain the transmitted QPSK constellation, you need to add some more blocks (to counter the frequency shift and multipath propagation) in flow graph of task 2.

Building the Receiver

Use following blocks to build the rx section:

Read input2.bin file using File Source block.

- FLL Band-Edge:



- Polyphase Clock Sync: Use the same parameters as used in Task 2.

Contd.

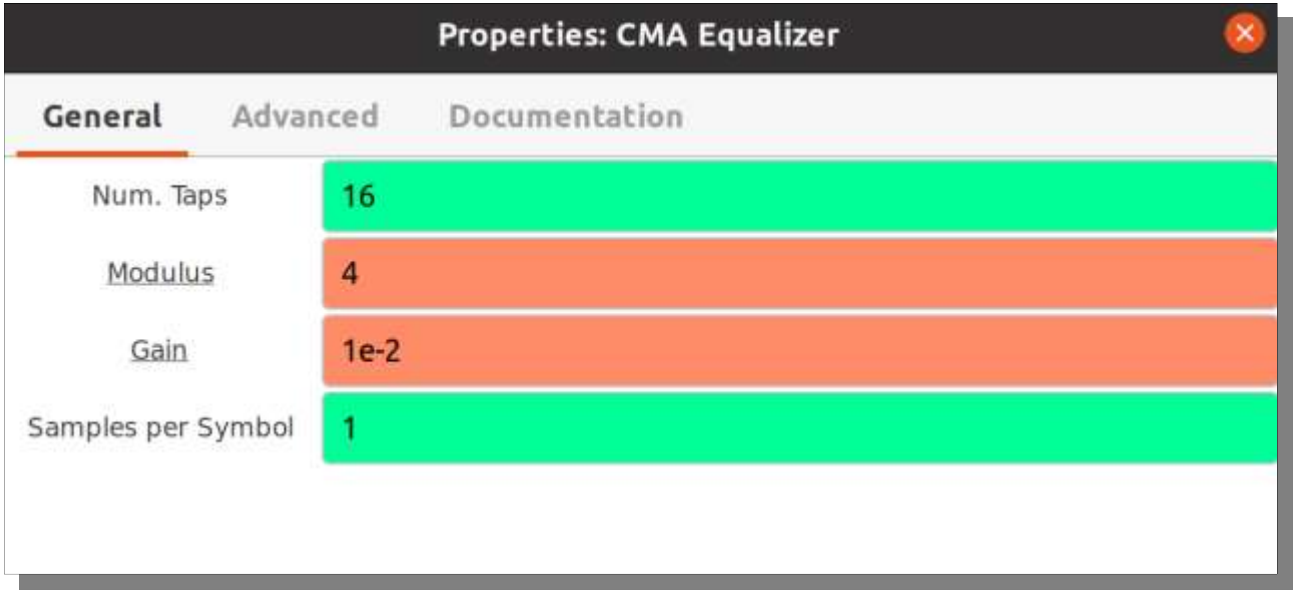
- Costas Loop:

✓ Try to obtain the transmitted QPSK constellation diagram in scope sink(x-y).

- You may need to use CMA equalizer block **BEFORE** the Costas loop to get required/better constellation due to mutipath effect (this equalizer will be discussed in Experiment 7)
- An AGC (automatic gain control) block may also help if amplitude is fluctuating.

Contd.

- CMA Equalizer:



- CMA Equalizer can be used to remove the multipath effect from signal. It should be used before the costas loop and after removing the pulse shaping (More about CMA equalizer will be discussed in Exp 7).
- Experiment by varying the different parameters of CMA equalizer to observe the effect on clustering of constellation of demodulated signal (You can read the documentation of CMA block to get an idea of purpose of these parameters).