Lab 1 - Introduction to Software-Defined Radio



**ECE531 – Software Defined Radio**

**Spring 2022**

Daniel Gallagher, danielgallagher@arizona.edu

Department of Electrical & Computer Engineering

University of Arizona, Tucson AZ 85721

*Due Date: February 3, 2022*

Given the flow graph in Figure 4, complete the following:

1. Construct and execute the GRC flowgraph shown in Figure 4.

Diagram

Description automatically generated

1. Visualize the sinusoidal signal in time and frequency, provided by the “Time” and “Frequency” sink tabs.
2. Using the default 10kHz sample rate: Zoom in on the time domain signal. Does this look as expected? Determine the location of the frequency peak. Hover over the peaks and measure the frequency location.
   1. *The peak occurs at 2kHz. As the sample frequency (10kHz) is more than the Nyquist frequency (4kHz => 2kHz), the signal looks as expected.*

Graphical user interface, chart, histogram

Description automatically generated

1. While continuing to observe frequency tab, slowly adjust the sampling rate from 10kHz to 40kHz. Determine the measured frequency? (Note: you may need to give the frequency sink a moment to adjust to the new sample rate.)
2. What does the time domain signal look like at this new sample rate? Why?
   1. *The signal attenuates to .5kHz. After setting the Bandwidth equal to the sample rate, the frequency remains at 2kHz.*

Graphical user interface, histogram

Description automatically generated

1. Adjust the sampling rate to 3.5kHz. What is the measured frequency? Did the measured frequency remain the same or did it change? Why? If so, how much did the frequency change and why?
   1. *The new measured frequency is 1.5kHz. This due to the sampling rate falling below the Nyquist Frequency (4kHz) and aliasing as 1.5kHz.*

Graphical user interface

Description automatically generated

1. Save the grc file with a unique filename for use in later sections.

## Complex Sampling

Direct conversion transceivers, like shown in Figure 2, utilize in-phase (I) and quadrature (Q) signal paths to sample a complex signal. In-phase refers to the signal that is in the same phase as the local oscillator(LO),and quadrature refers to the part of the signal whose phase is shifted by 90°.

A bandpass signal can be represented by the sum of its in-phase (I) and quadrature (Q) components:

*S*(*t*) = *I*(*t*)cos(2π *fct*)− *Q*(*t*)sin(2π *fct*). (1)

where *I*(*t*) is in-phase amplitude, *Q*(*t*) is quadrature amplitude and *fc* is carrier frequency.

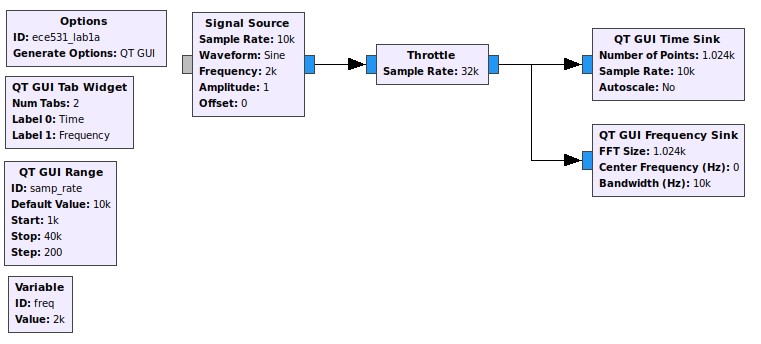


Figure 5: GRC flowgraph for Lab 1, Section 3.3

Given the flow graph used previously in Figure 4:

1. Change all block input/output types from “Float 32” to “Complex Float 32”. The port color on each block should change from orange to blue as shown in Figure 5. (Note: You can quickly change the input/output type by highlighting the block(s) and pressing the up/down arrows). 2. Visualize the sinusoidal signal in the frequency domain. What has changed? Why?

1. What are the two time domain waveforms shown on the “Time” tab? What is the phase relationship between them?
2. While observing the time domain tab, slowly adjust the sampling rate from 10kHz to 40kHz. Does the signal amplitude appear as expected for each channel? (Note: You may wish to zoom in on the time signal).
3. What is the measured frequency peak at 40kHz sampling rate?
4. While observing the frequency domain tab, slowly decrease the sampling rate to 4kHz. What is observed?
5. Continue to slowly decrease to slower sampling rates. What happens? Why?
6. Save the grc file with a unique filename for use in later sections.

## Frequency Observations

### Complex-sampled Flowgraph

Given the complex-sampled flowgraph used previously in Figure 5:

1. Add an adjustable frequency slider using a QT GUI Range block with the following parameters: **Frequency** Default value: 2e3, Start: 0, Stop: 15e3, Step: 100
2. Set the signal source frequency to the variable ID used for the QT GUI Range block above.
3. Using a fixed 10kHz (default) sample rate: Slowly increase the frequency to the maximum frequency value while observing the frequency sink output.
4. Describe what anomaly is observed when increasing the frequency slider. What is this anomaly? Why does it occur?

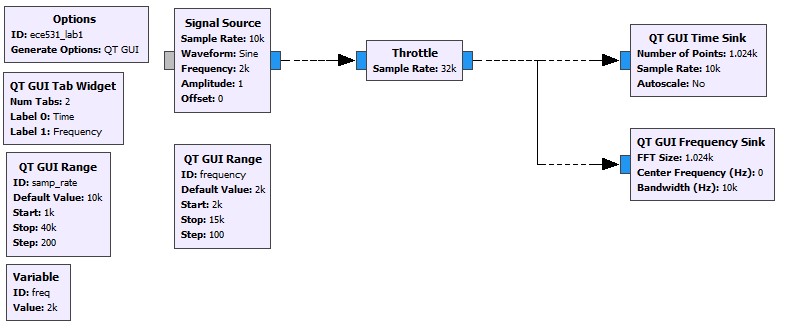


Figure 6: GRC flowgraph for Lab 1, Section 3.4

### Real-sampled Flowgraph

Given the flowgraph used previously in Figure 4, or by changing the types to real; Repeat the steps above, specifically:

1. Add an adjustable frequency slider using a QT GUI Range block with the following parameters: **Frequency** Default value: 2e3, Start: 0, Stop: 15e3, Step: 100
2. Set the signal source frequency to the variable ID used for the QT GUI Range block above.
3. Using a fixed 10kHz (default) sample rate: Slowly increase the frequency to the maximum frequency value while observing the frequency sink output.
4. What is observed when increasing the frequency? How and why is this different than what was observed in Section 3.4.1?

## I/Q Imbalance

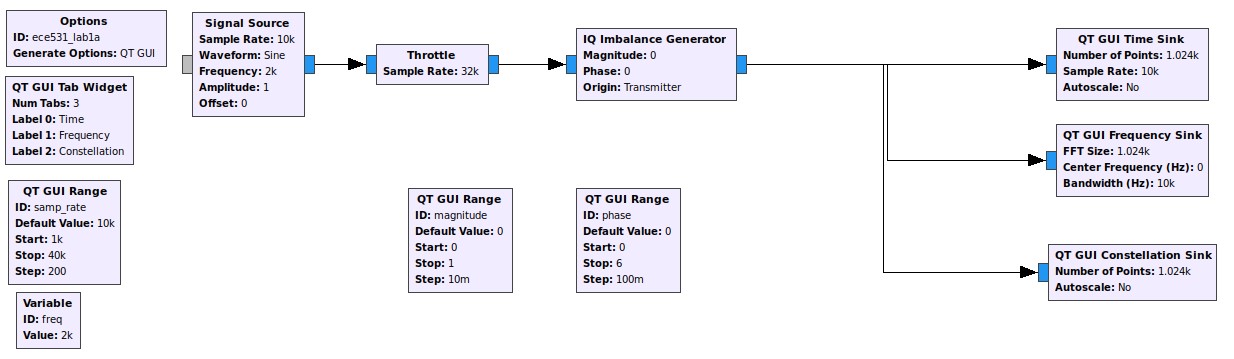


Figure 7: GRC flowgraph for Lab 1, Section 3.5

Given the flow graph used previously in Figure 5:

1. Add an IQ Imbalance Generator block to the flowgraph somewhere in the signal path between the signal source and sink blocks. An example is shown in Figure 7
2. Create two QT GUI Range blocks, one with variable ID “magnitude” and another with ID “phase”. Set the following parameters respectively:

**Magnitude** Default value: 0, Start: 0, Stop: 1, Step: 0.1

**Phase** Default value: 0, Start: 0, Stop: 6.2, Step: 0.1

1. Set the IQ Imbalance Generator values for magnitude and phase to the variable IDs provided to the GUI Range blocks. (Note: these happen to be the Python variable names. You can use Python syntax in most block fields).
2. Add a “QT GUI Constellation Sink” block to a third tab and connect to the same location as the other sinks.
3. Your flowgraph should now resemble the one shown in Figure 7.
4. Execute the new flowgraph. The time and frequency domain should match what was observed previously.
5. While observing the frequency domain tab, slowly increase the phase imbalance from zero. What happens? Why?
6. Return the phase offset to zero and then slowly increase the amplitude imbalance from zero. What happens? Why?
7. While observing the constellation sink output, adjust each of the three variables independently.

What happens when each slider is adjusted?

## Adding Noise

Given the flow graph used previously in Section 3.5:

1. Add gaussian noise to the signal using the “Noise Source” and “Add” blocks.
2. Describe what happened to the signals displayed on each tab; time, frequency, and constellation.
3. Use the “QT GUI Histogram Sink” blocks to verify the pdf of the various noise source types by connecting directly to the noise source.

## Interpolation and Decimation

The following flowgraph uses the three signal sources with identical parameters; including frequency.

Each source is resampled using a rational resampler block and displayed on three-port sink blocks.

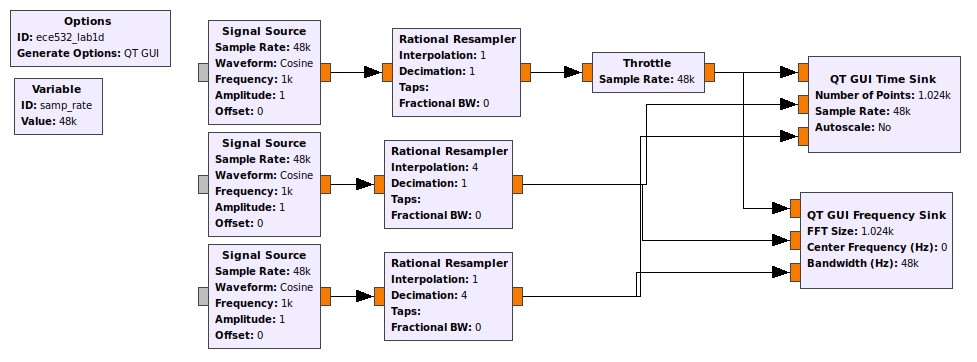


Figure 8: GRC flowgraph for Lab 1, Section 3.7

Given the flow shown in Figure 8:

1. Construct and run the given flowgraph.
2. Observe the time and frequency domain outputs.
3. What do you observe with the three waveforms?
4. The three signal sources have identical properties including frequency. Why do they differ when the frequency spectrum is visualized?
5. What are the relationships between the waveforms? Which one is interpolated and decimated?

# Questions

1. What are the benefits of using in-phase and quadrature (I & Q) samples for SDR? Fully describe at least three benefits of IQ sampling.
2. In GNU Radio What is the Throttle block for? When should it be used? What happens if you use more than one throttle block? When is a throttle block unnecessary?
3. What are Nyquist zones? How are they useful with software-defined radio?
4. Why is dither noise used on Digital-to-Analog converter (DAC) circuits?

# Lab Report Preparation & Submission Instructions

Include all your answers, results, and source code in a laboratory report formatted as follows:

* Cover page: includes course number, laboratory title, name, submission date.
* Suggested: Table of contents, list of tables, list of figures.
* Commentary on designed implementations, responses to laboratory questions, captured outputs, and explanation of observations.
* Meaningful conclusions to the lab.
* Source code (as an appendix). You may also upload source files with report submission.

Remember to write your laboratory report in a descriptive approach, explaining your experience and observations in such a way that it provides the reader with some insight as to what you have accomplished. Furthermore, please include images and outputs wherever possible in your laboratory report document.

# References

1. Canonical Ltd. (2018) Download ubuntu desktop. [Online]. Available: [https: //www.ubuntu.com/download/desktop](https://www.ubuntu.com/download/desktop)
2. Corgan Labs. (2017) GNU Radio Live SDR Environment. [Online]. Available: <https://wiki.gnuradio.org/index.php/GNU_Radio_Live_SDR_Environment>
3. B. Bloessl, “Github: Instant gnu radio.” [Online]. Available: [https://github.com/bastibl/ instant-gnuradio](https://github.com/bastibl/instant-gnuradio)
4. GNU Radio Foundation, Inc. GNU Radio. [Online]. Available: <https://www.gnuradio.org/>
5. The Comprehensive GNU Radio Archive Network. [Online]. Available: [http://cgran.org](http://cgran.org/)
6. E. Blossom, “gr\_modtool: Swiss Army Knife for editing GNU Radio modules and -components.” November. [Online]. Available: [https://wiki.gnuradio.org/index.php/OutOfTreeModules#gr\_ modtool\_-\_The\_swiss\_army\_knife\_of\_module\_editing](https://wiki.gnuradio.org/index.php/OutOfTreeModules#gr_modtool_-_The_swiss_army_knife_of_module_editing)
7. The Mathworks. ADALM-PLUTO Radio Support from Communications Toolbox. [Online]. Available: <https://www.mathworks.com/hardware-support/adalm-pluto-radio.html>
8. Analog Devices, “GitHub: Analog Devices Inc.” [Online]. Available: [https:](https://github.com/analogdevicesinc)

[//github.com/analogdevicesinc](https://github.com/analogdevicesinc)