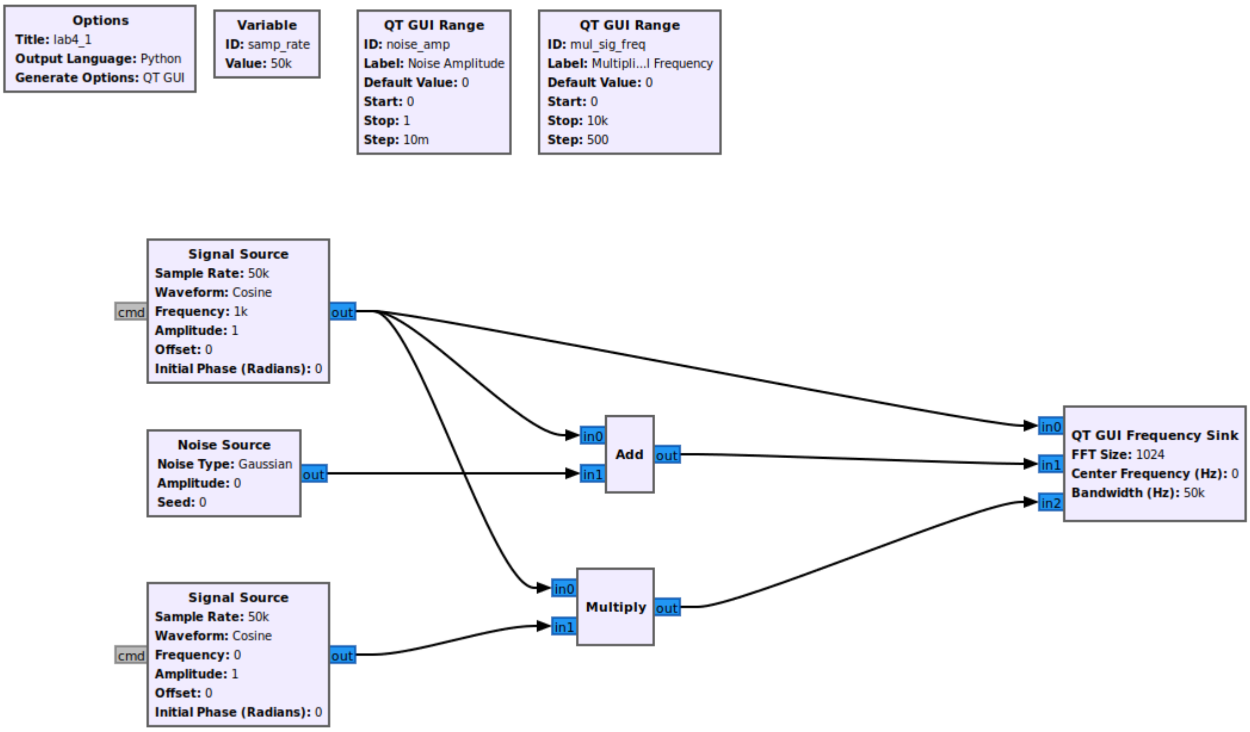
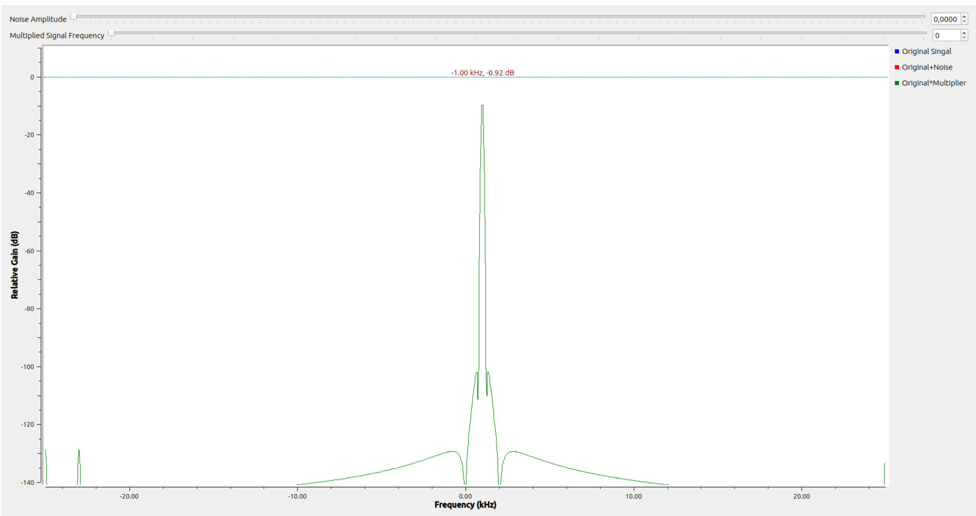
HY330 – Telecommunication Systems
Chris Papastamos | csd4569
Assignment 4

Exercise 1

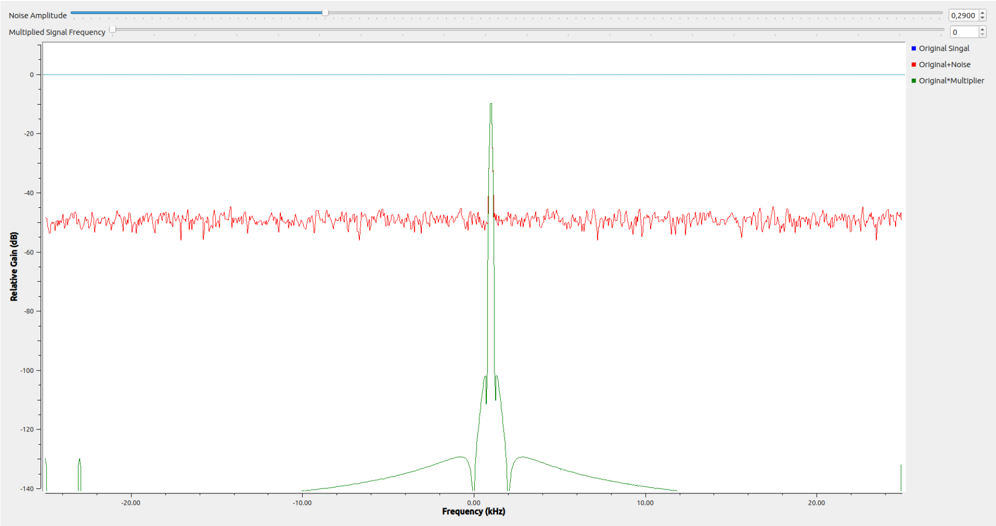
For the purpose of this exercise I created the following flowgraph:



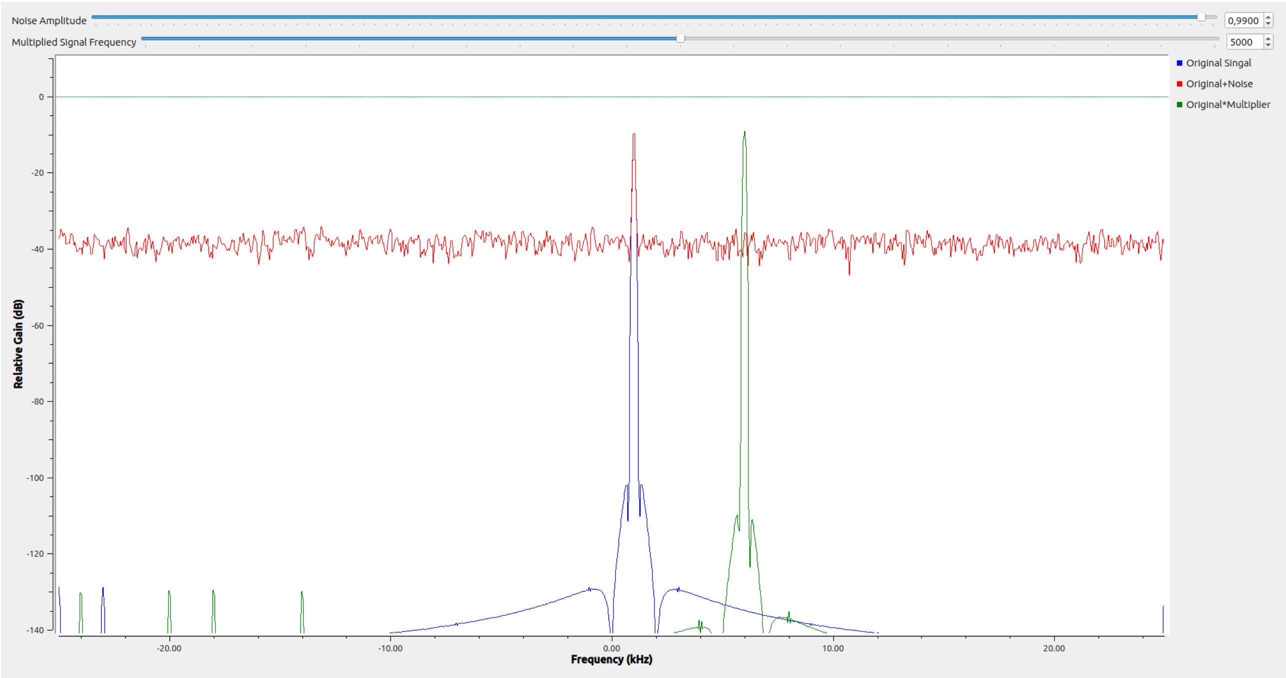
Lets execute and observe the different outputs produced when changing the two sliders:



Starting up we can see the three signals aligned. This is expected as no modifications have been applied.



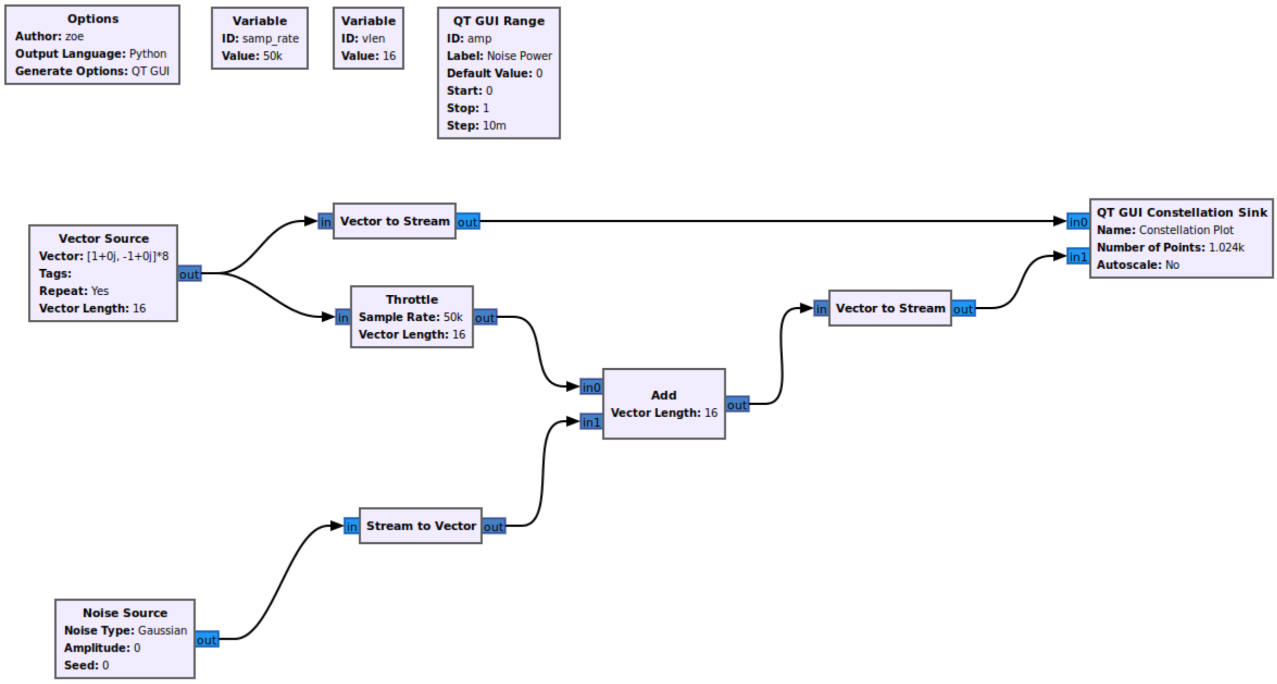
After adding some noise, we can observe additional “noise” frequencies in the signal with the added noise.



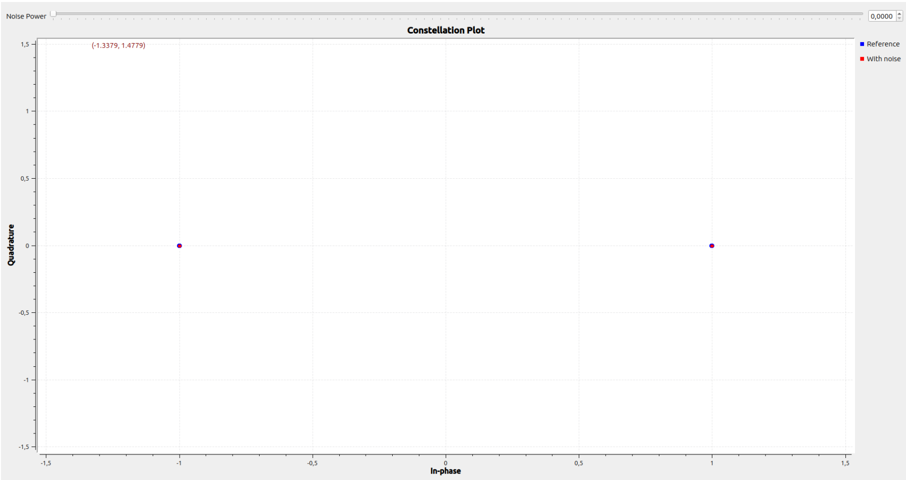
By changing the second signal’s frequency, the multiplication’s output is the sum of the two frequencies. That is the definition of CFO, The specified value is the Frequency Offset from the Carrier frequency.

Exercise 2

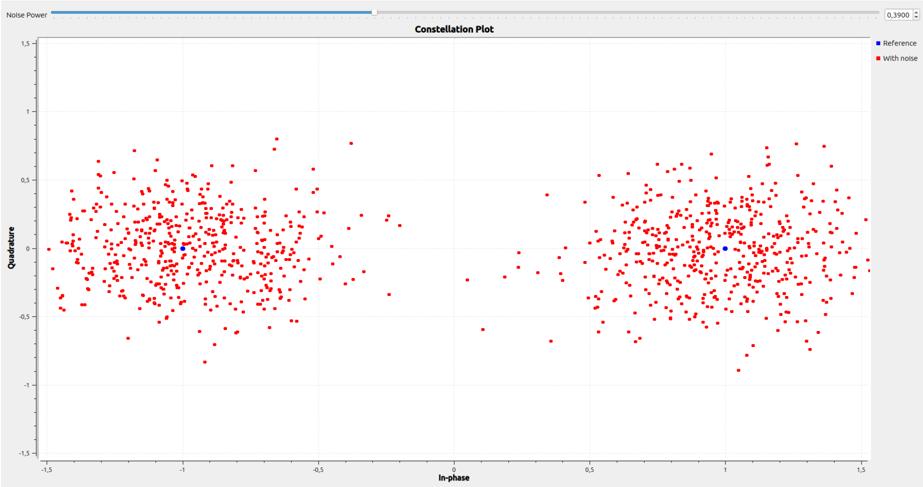
The requested flowgraph for this exercise is the following:



Lets execute and take a look at the output:



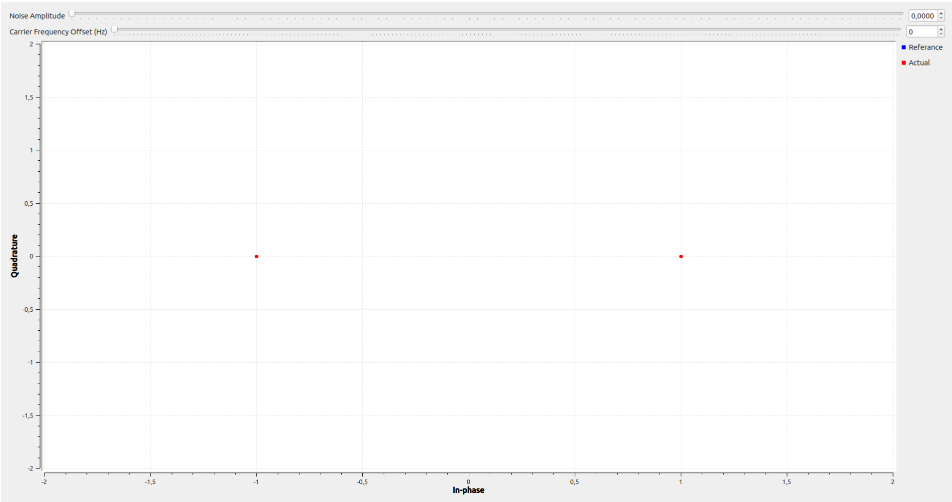
Initially, the red signals constellation points align with the reference points since no noise is introduced to the signal.



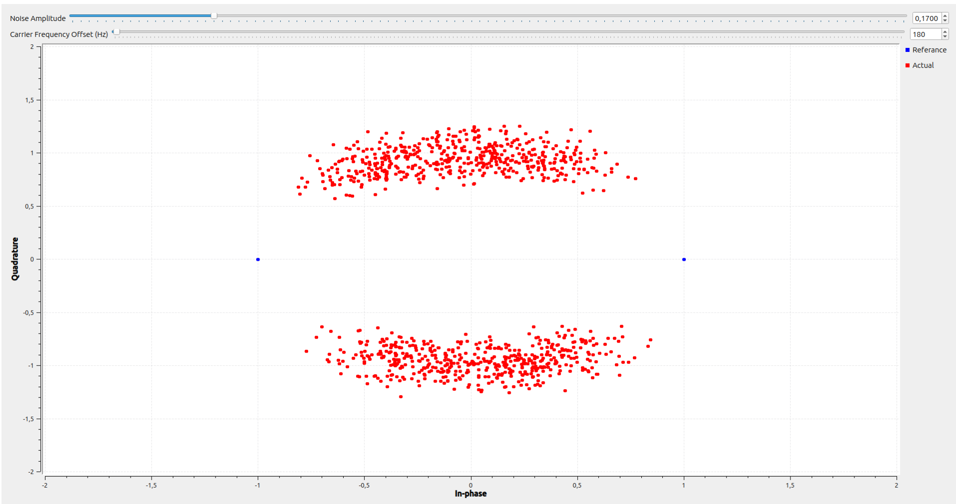
After adding some noise we can see the signal being spread around the reference points.

Exercise 3

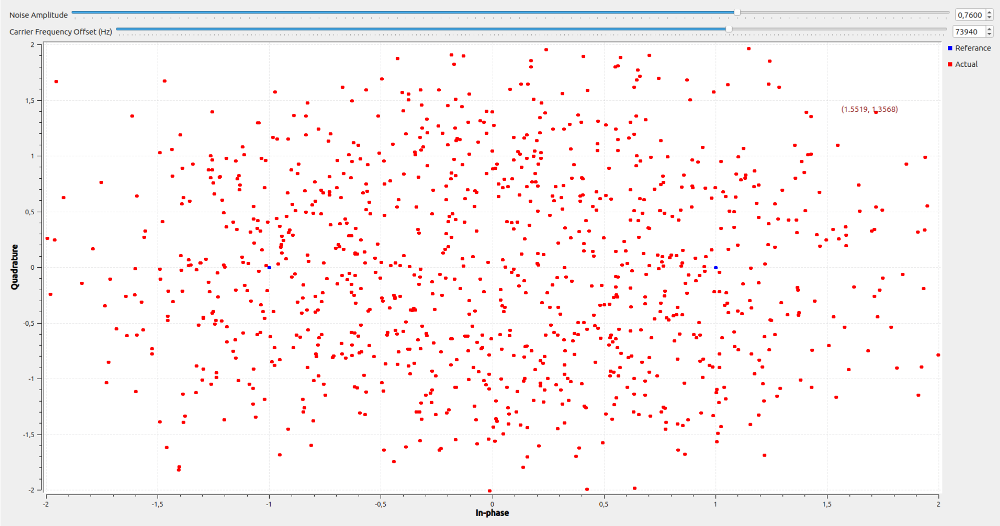
By executing the given flowgraph we observe the following output:



We can see the output signal aligns with the reference constellation points. That is because there is no noise and no CFO. Lets add some:

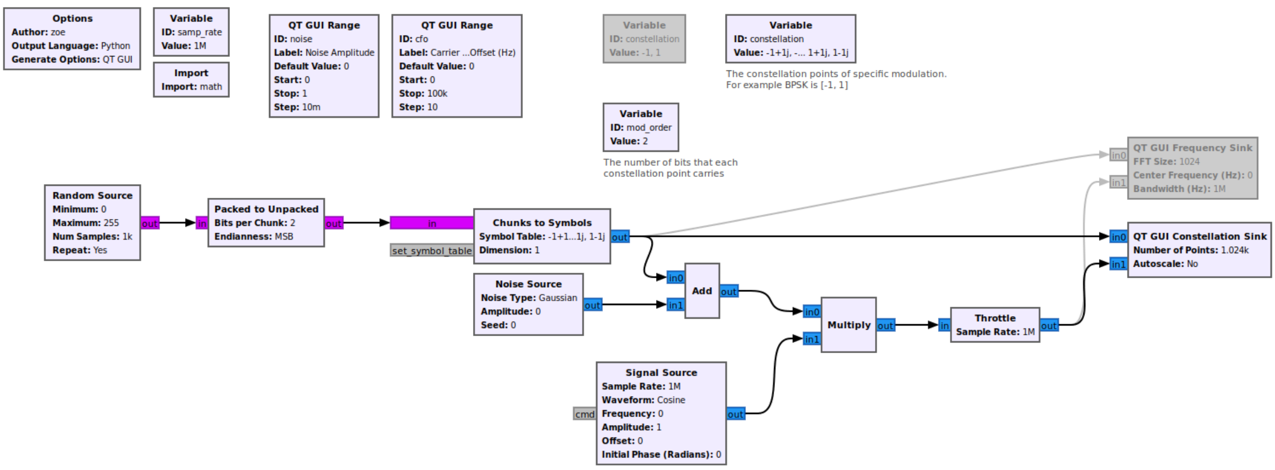


We can see that even with a small amount of noise and CFO the signal goes away a lot from its original constellation points. Lets try pumping up the variables a little bit more:

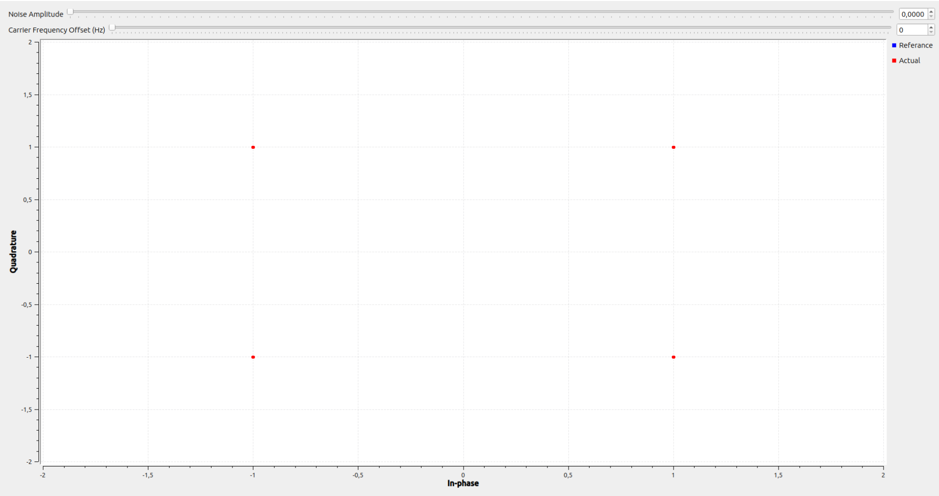


With such high amounts of noise and CFO the output signal is completely ruined. Playing some more with the noise and CFO variables, we can see that **noise** controls the spread of the points and the **CFO** controls the circular movement of the points around the center point

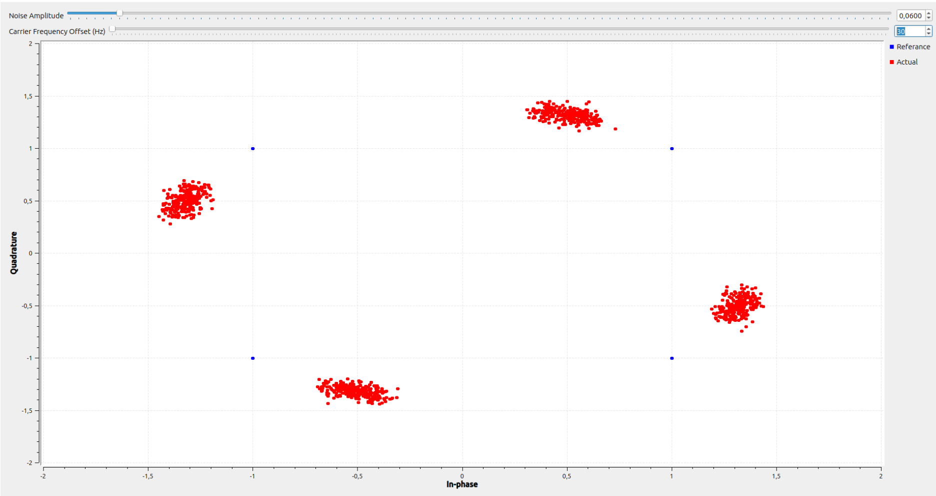
In order to convert the flowgraph to display **QPSK** I have to make 2 changes. The first is the number of bytes per constellation point by changing the mod_order variable to 2 bits. The second change is to change the actual constellation points by changing the constellation variable. The new constellation points will be: -1-1j, -1+1j, +1-1j, +1+1j. The final flowgraph looks like this:



The output of the flowgraph is the following:

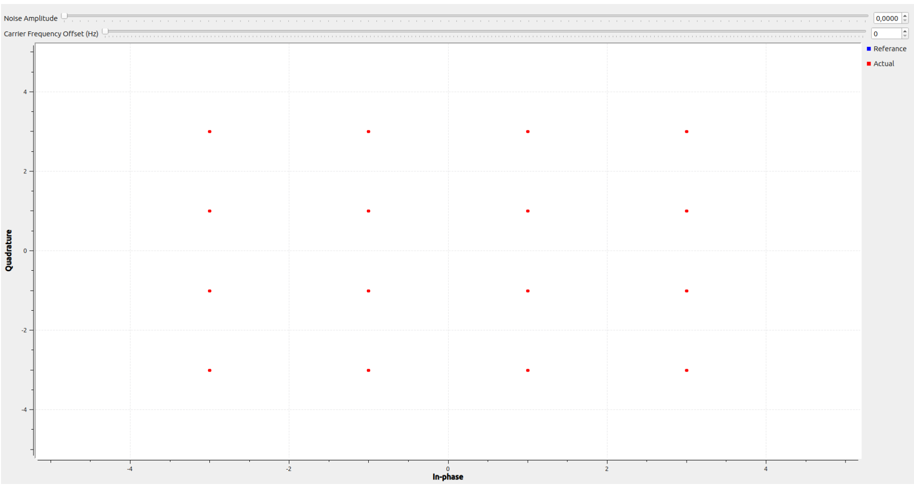


We can see that the four constellation points appear in the desired locations according to the QPSK specification. Lets add some noise and CFO:

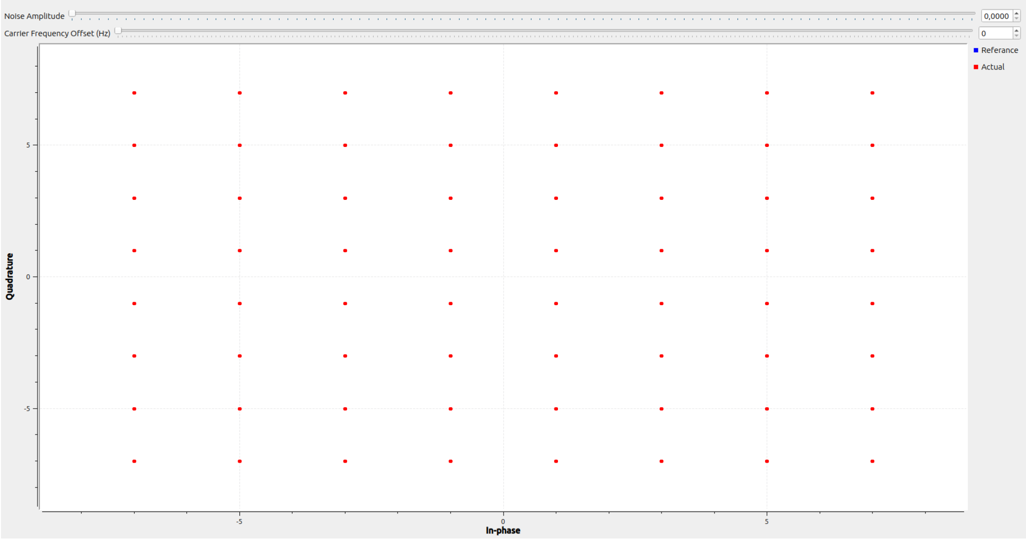


We can observe here the spread and the offset as expected!

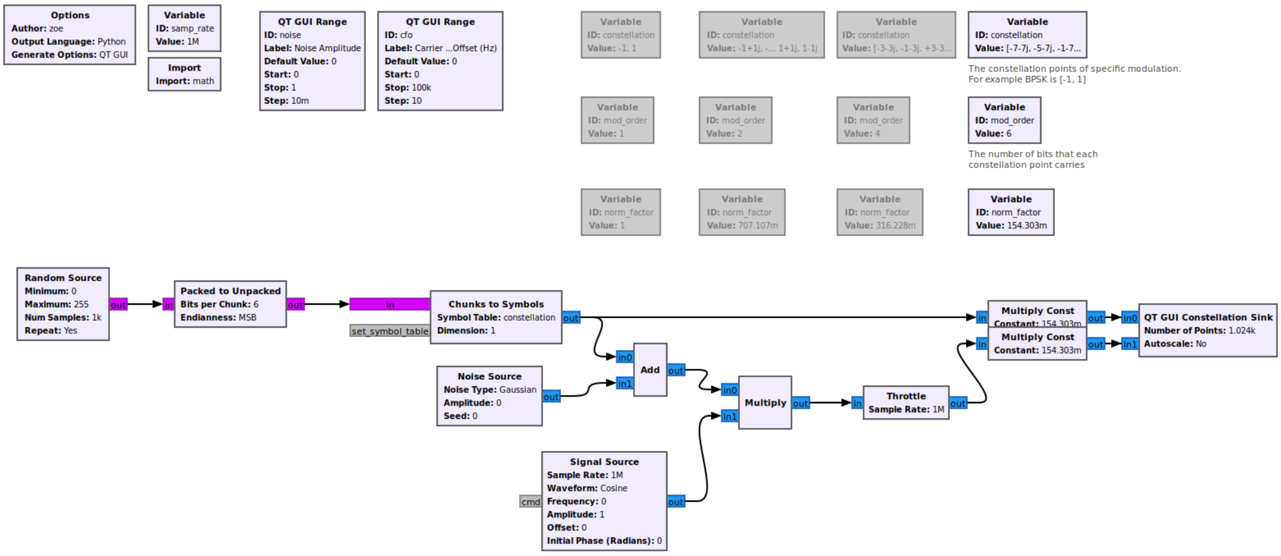
To display the **16QAM** system we will again change the <u>mod_order</u> to 4bits and <u>constellation</u> according to the new points. The new points are going to be: <u>-3-3j, -1-3j, +3-3j, +1-3j, -3-1j, -1-1j, +3-1j, +1-1j, -3+3j, -1+3j, +3+3j, +1+3j, -3+1j, -1+1j, +3+1j, +1+1j</u>
The output of this system is the following:



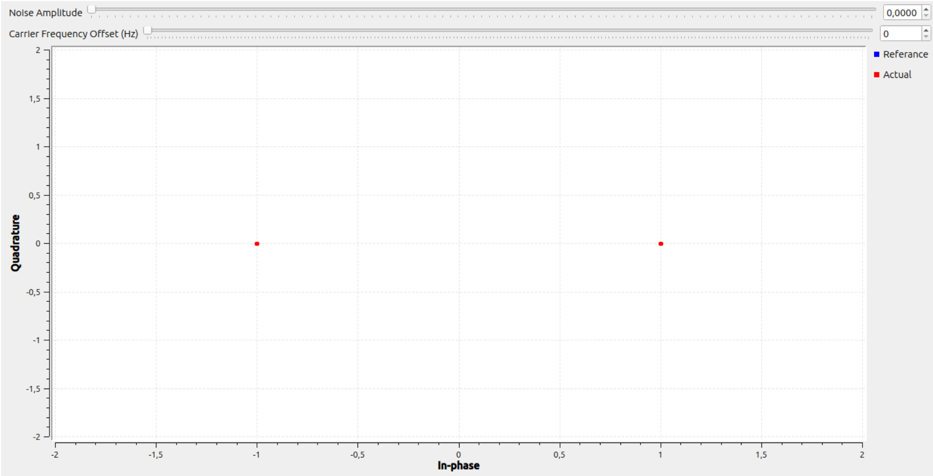
Finally for the **64QAM** system the mod_order is going to be 6bits and the constellation points will be changed to match (I am not going to include the list because of its size). The output of the system is the following:

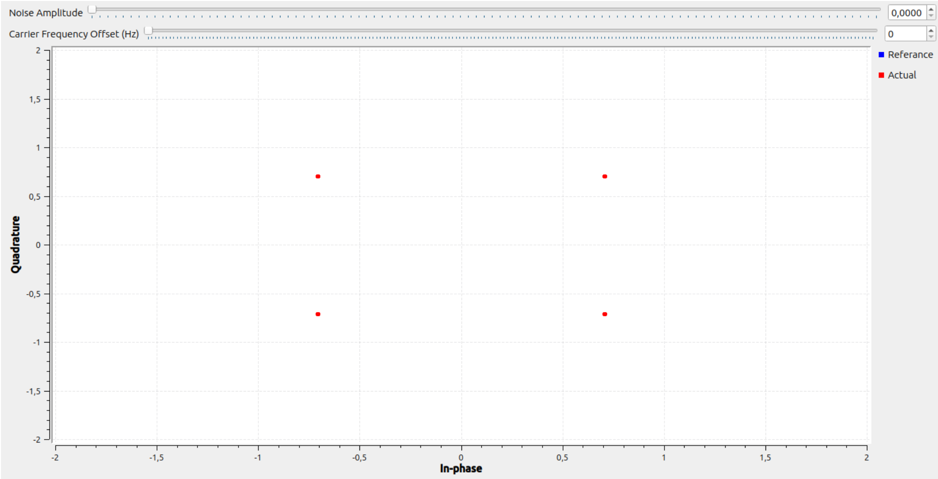


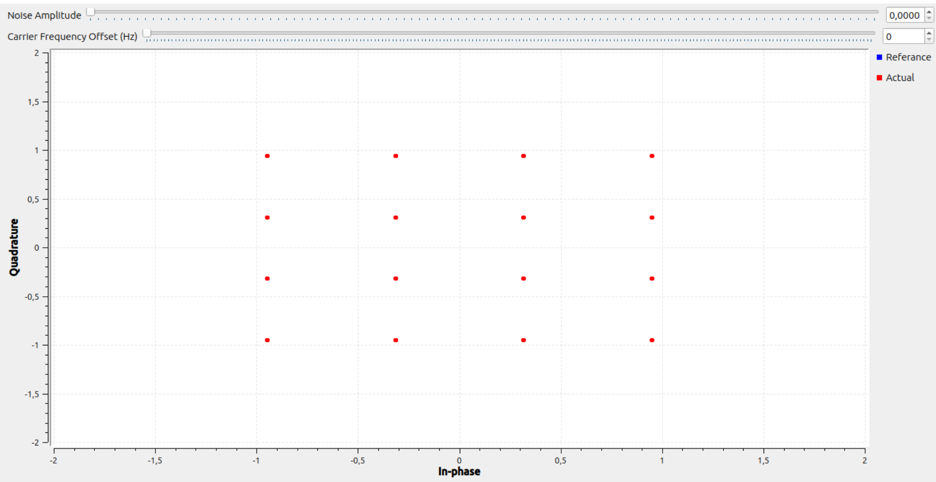
In order to **normalize** the constellation points of each system, I am going to multiply each signal with their system's respective normalization factor. The updated flowgraph is going to look like this:

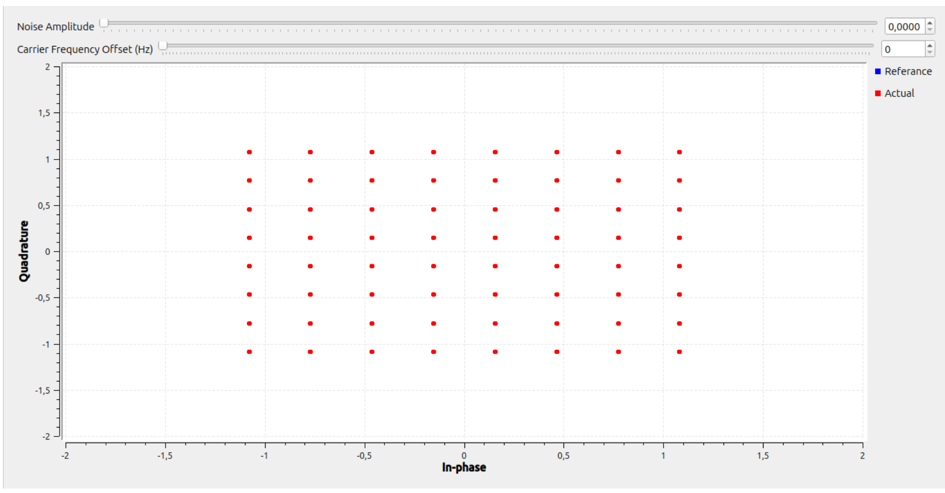


The new output of each system is going to be the following:

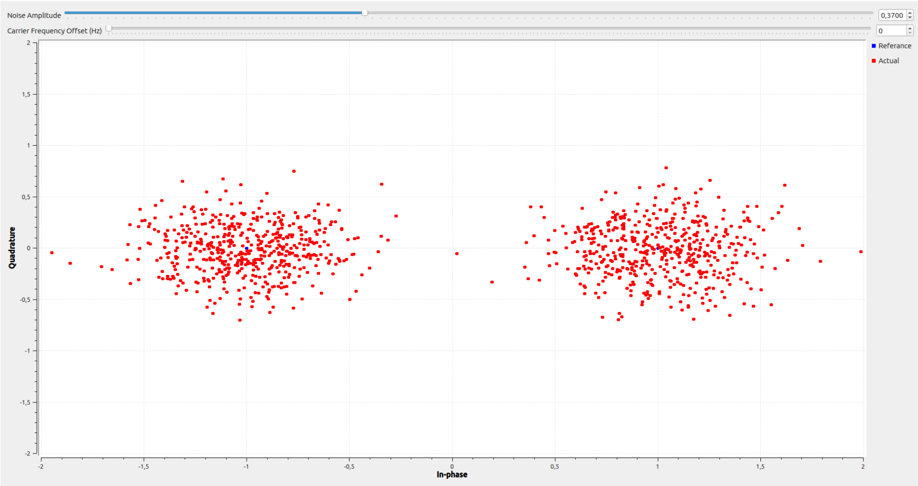

Noise Amplitude
0,0000
Carrier Frequency Offset (Hz)
0
Referance
Actual
Quadrature
In-phase

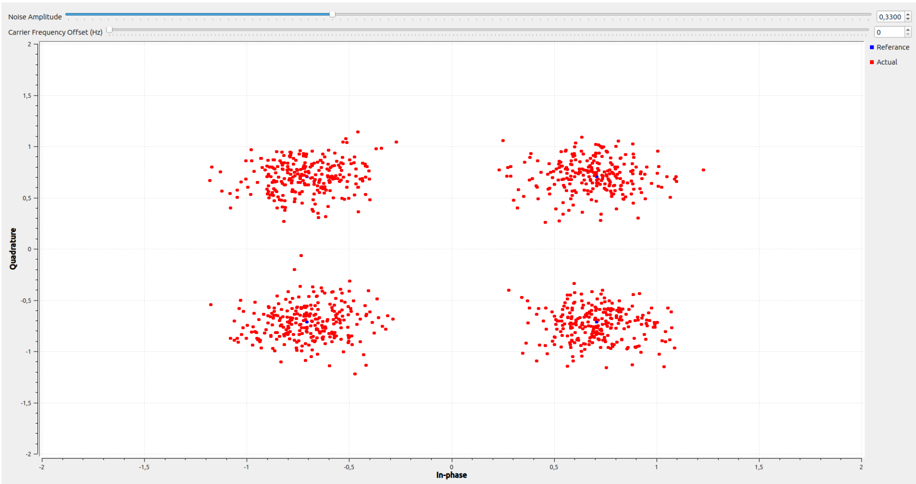

Noise Amplitude
0,0000
Carrier Frequency Offset (Hz)
0
Referance
Actual
Quadrature
In-phase

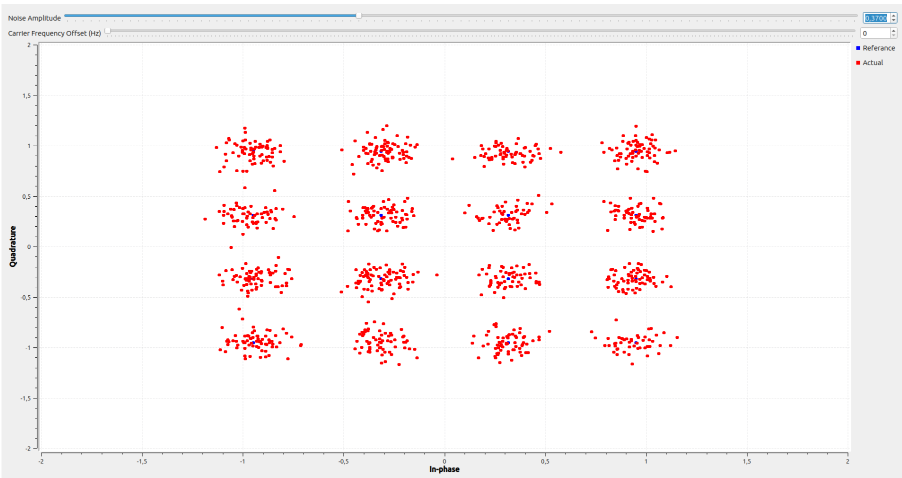

Noise Amplitude
0,0000
Carrier Frequency Offset (Hz)
0
Referance
Actual
Quadrature
In-phase

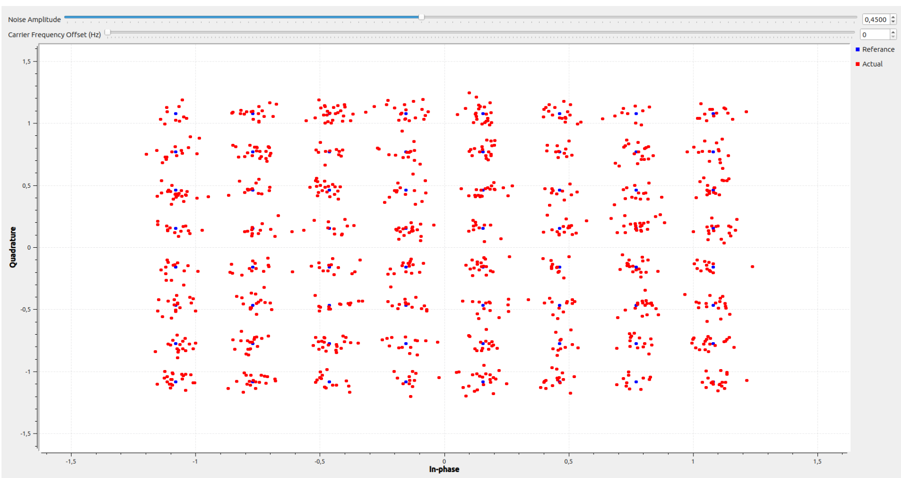

Noise Amplitude
0,0000
Carrier Frequency Offset (Hz)
0
Referance
Actual
Quadrature
In-phase

Lets now look at how much noise each system can handle without having errors:

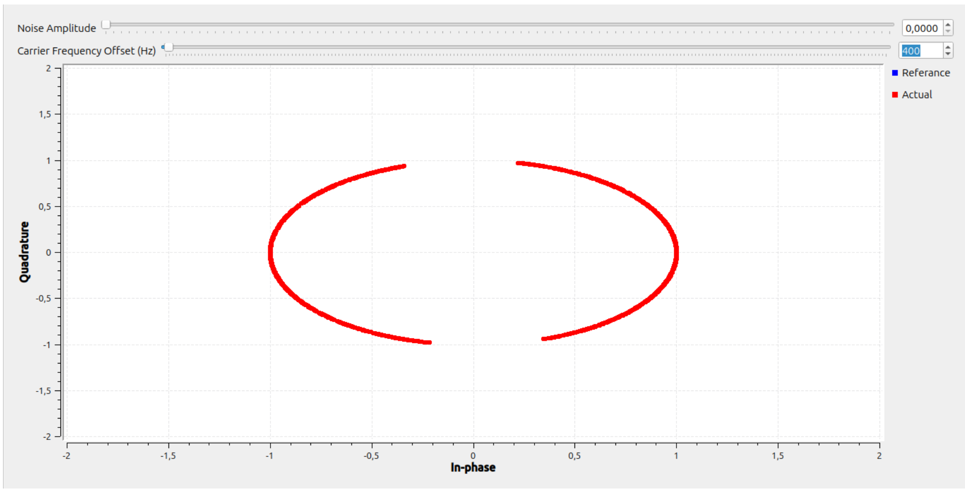


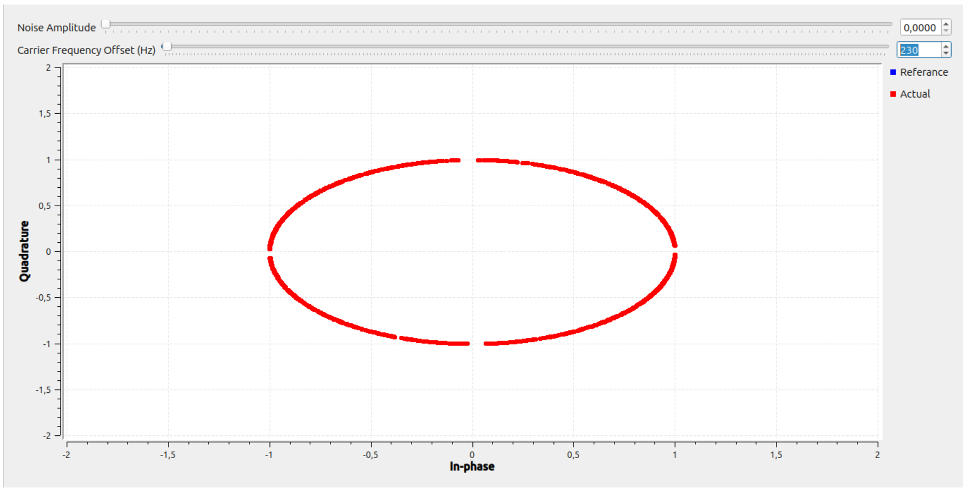


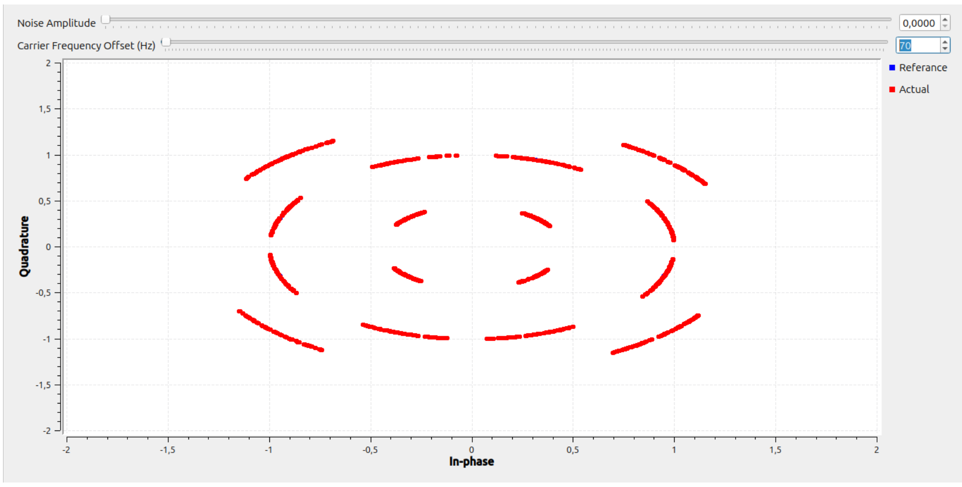


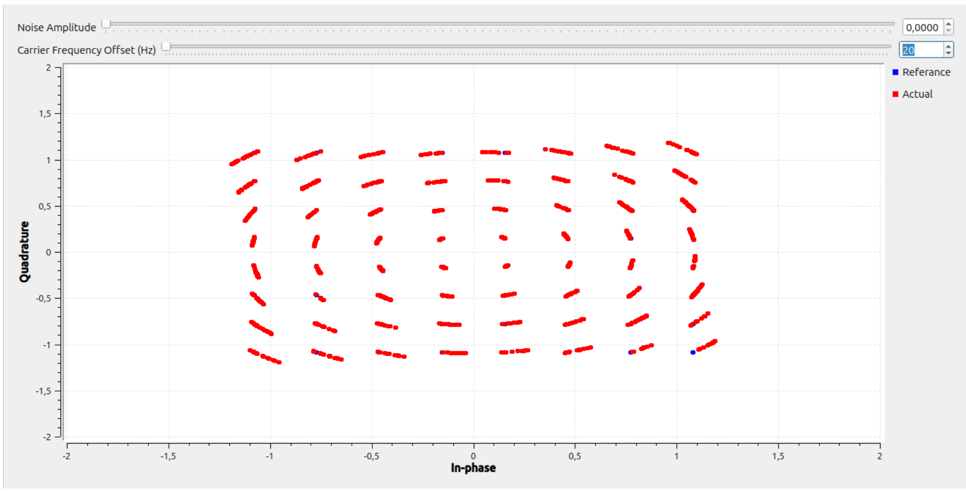


Lets now look at how much CFO each system can handle without having errors:







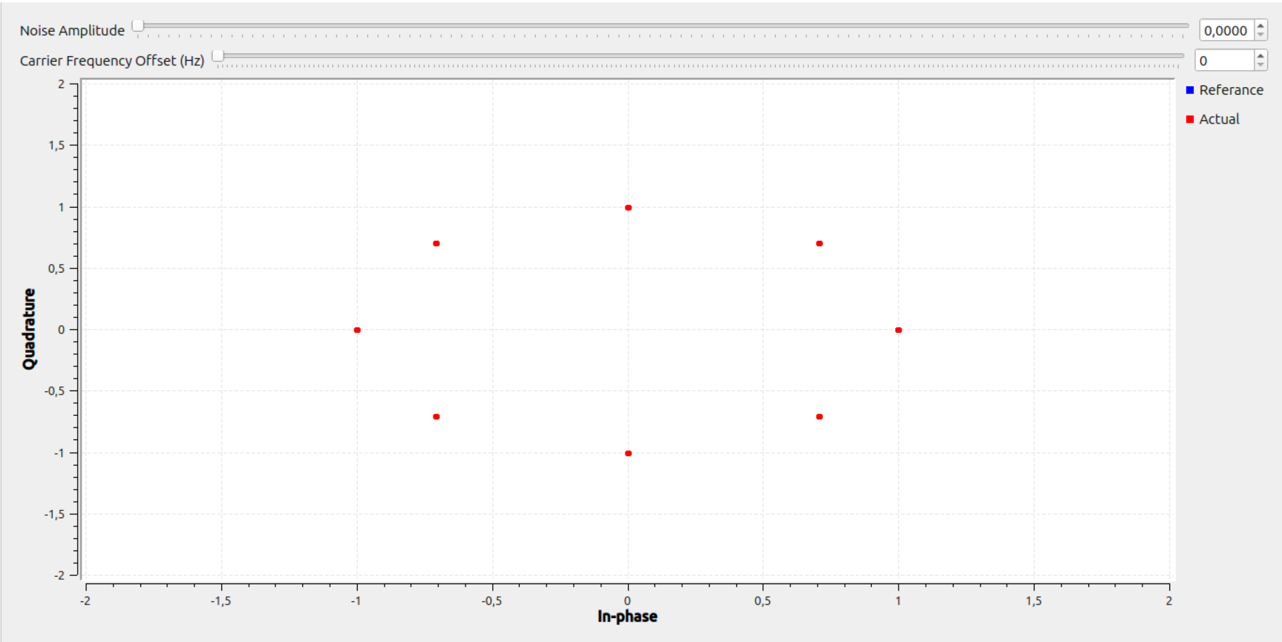


Exercise 4

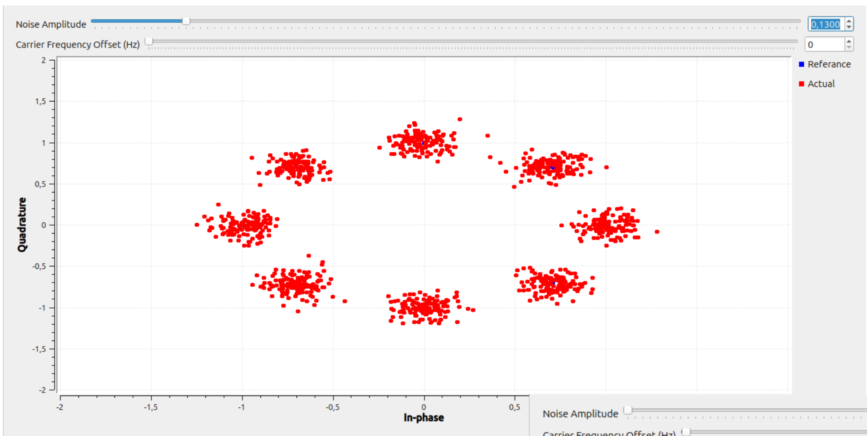
Knowing the first point of the 8PSK system is 1+0j and that all constellation points in the system are in a circle, we will assume that all points will have a distance from the center point of 1. The axis points will be the following: 0-1j, 1+0j, -1+0j, 0+1j. Using basic trigonometric calculations we can calculate that the remaining points will have a distance of $\frac{1}{\sqrt{2}}$ from the axis. Finally, we have the following constellation points:

$$-\frac{1}{\sqrt{2}}-\frac{1}{\sqrt{2}}j, 0-1j, 1+0j, \frac{1}{\sqrt{2}}-\frac{1}{\sqrt{2}}j, -1+0j, -\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}j, \frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}j, 0+1j$$

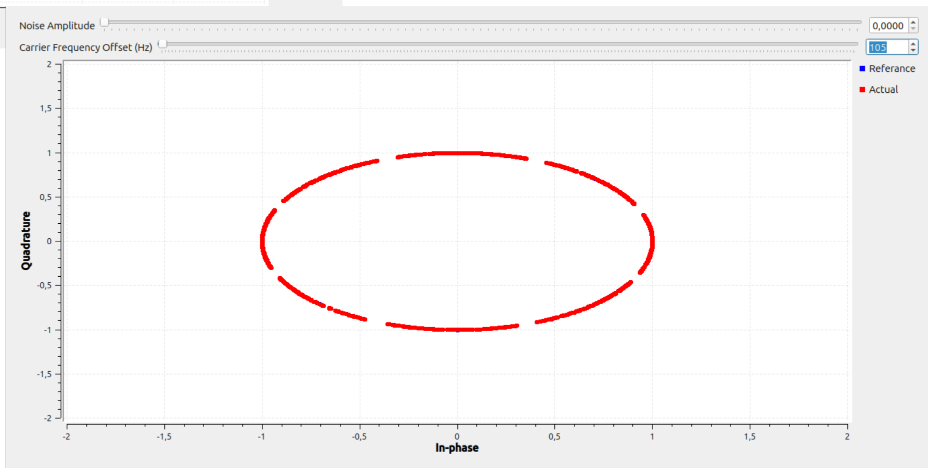
Using the above constellation points and a mod_order of 3bits we get the following output from our system:

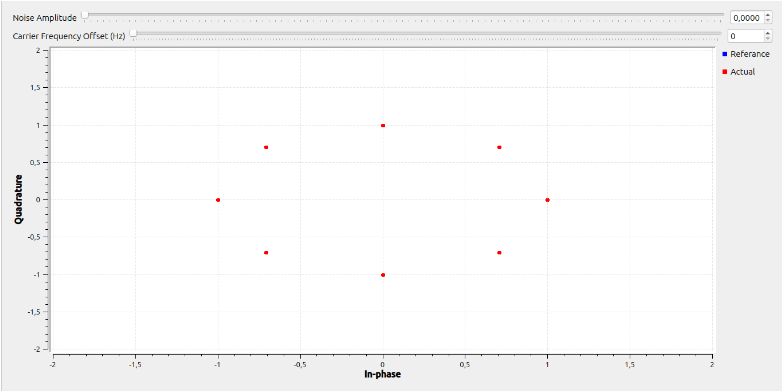


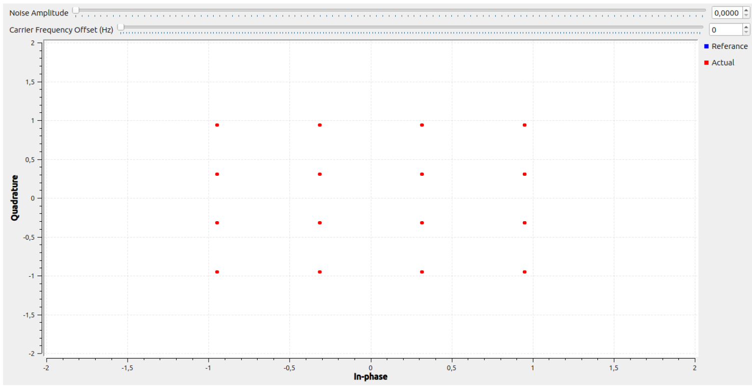
Testing out the maximum values of noise and CFO I came down to the following results:



The maximum value of noise without having bit errors, was 0,13 which is less than QPSK. For the value of CFO, the maximum value was 105Hz which was also less than QPSK. These results are logical since the constellation points in 8PSK are much closer than QPSK



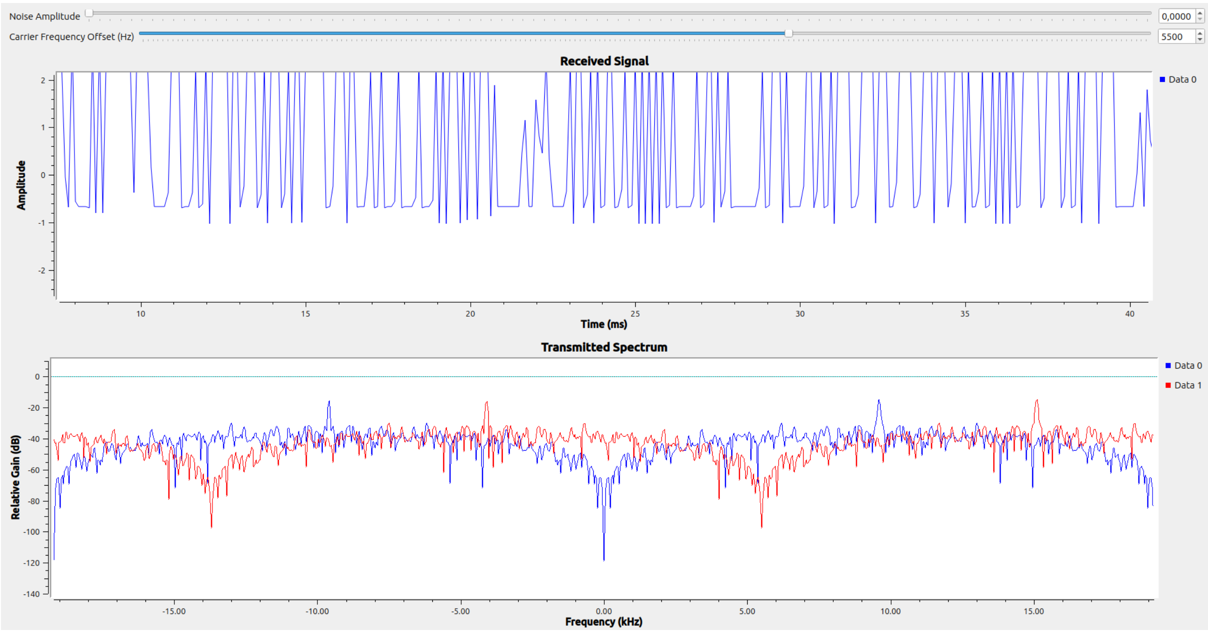


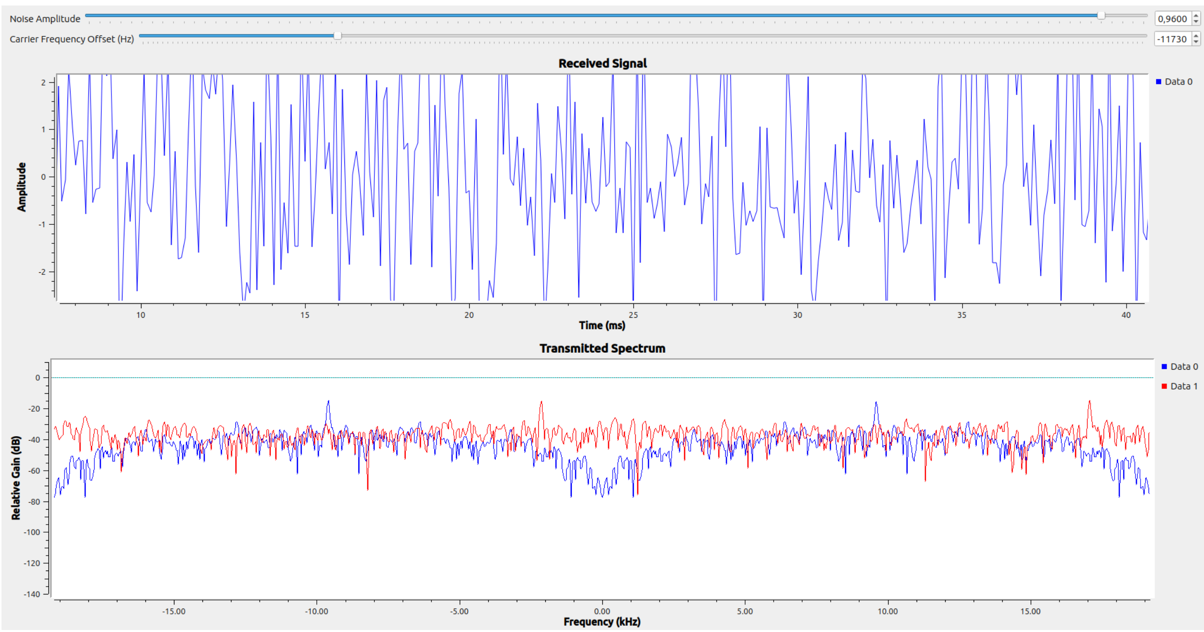


Comparing the 8PSK and 16QAM spectrum, we can observe some obvious differences. First the two systems are keying their constellation points differently. 8PSK uses Phase Shift Keying while 16QAM uses Quadrature Amplitude Modulation. The two systems also have different amounts of constellation points. As their names suggests 8PSK has 8 constellation points while 16QAM has 16 points. Due to each spectrum's form, the constellation regions have different shape also. The 8PSK system's regions are the 'pizza slices" around the center point while the 16QAM system's are in a matrix containing a constellation in the center of each matrix box.

Exercise 5

First up, lets tweak the noise and CFO to observe the changes in the received signal:





As we can see the system is not affected by noise and CFO at the at the point where bit errors happen. In comparison, BPSK could handle about 0,37 points of noise and about 400Hz of CFO. This makes the FSK system more durable when it comes to the addition of noise and CFO.

Using the variable *deviation*, the distance in the frequency domain is described. Using this value, the *Modulation Index* is calculated. This is useful for the frequency mod block which will map each symbol to a frequency.

When CFO is added to the transmitted signal the whole frequency spectrum is shifted according to the amount of CFO. This means that the modulation points are also shifted out of their original place. With an easy transformation of the signals frequency (or the modulation points), the modulation of the signal can be restored.