Eye-diagram in GNURadio

September 7, 2019

1 Eye-diagram, Inter-symbol Interference (ISI)

Suppose we plot the delayed copies of a communication signal to the same plot window. Many overlapping waveforms will be present, and the plot may look all jumbled up. However, if the delays are integral multiples of the symbol duration, then the common time-reference allows us to deduce properties of the medium through which the signal reached the receiver. The multiple copies, each delayed by a symbol duration from the previous copy, when plotted is called an eye diagram. This was a useful technique to characterize the channel inter-symbol interference (ISI), much before the Fourier domain techniques, which boasts more accurate quantification.

In order to make a eye-diagram, we need a time scope accepting many inputs, however GNURadio allows a maximum of 10. Thus we will plot 10 delayed copies. We can construct a new block which accepts 10 inputs and plots each of them in the same window. This can be done by using a hierarchical block.

2 Hierarchical Block

We can group together a set of GNURadio blocks, and make it a single block with a name. This is called a hierarchical block. In this exercise, let us generate a *hier* block with the name *eyeplot*. This is easy, all you need to do is to open a new grc file, but opt of a hier (QT) block. In the topblock, by changing the id to 'eyeplot' (or 'nameulike'), you can create a new block called *eyeplot*. Now add the block diagram for the eyescope as as shown in Figure 1.

In this figure, we have chosen to plot 80 samples, and this scope is suitable when the communication data symbol has 40 or 80 samples per symbol. Remember to add the pad sink, which will act as a soft interface to the scope block you have generated. This file can be saved by any name you like (not so important), and after compiling (i.e. click the button adjacent to execute), this will be listed as a block under the name-id you have given. You can search for it in the gnuradio right window.

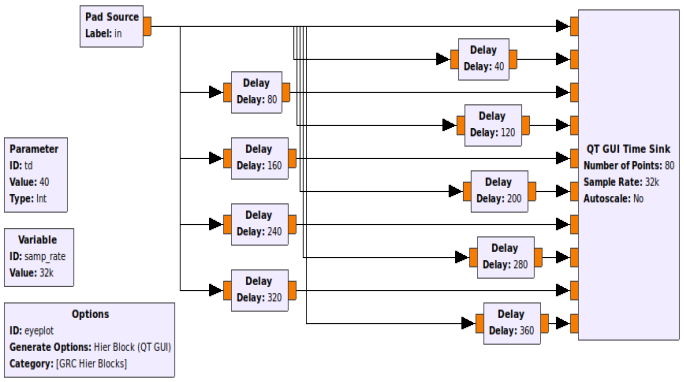

Figure 1: Name the topblock and connect

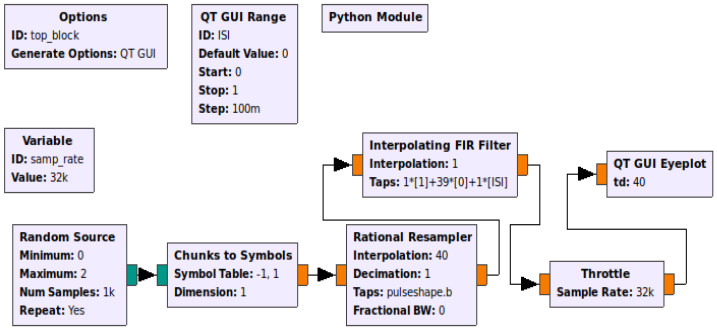

Figure 2: EyeDiagram for a Communication Signal

3 Eyediagram

We will now generate a communication signal and display on the eyescope we have generated. For simplicity choose a BPSK signal, and a sinc pulse-shaping filter. We have chosen to display 40 samples per symbol, just to mimic a continuum of values, yet matching the eyescope we have made. The connections are as shown in Figure 2

We have used a python-module to generate the filter taps. In principle one can employ the filter design functions of gnuradio, but for clarity, it is better to put your own design values initially. The python-module with id *sincfilt* can be opened in the editor, and including the following commands there will give the filters.

```
#-------------------------------------------------------
# this module will be imported in the into your flowgraph
import numpy as np

c=np.zeros(241);
x=range(-120,121,1);
val=0

#below is a linear interpolation for 40 samples per symbol.
for val in x:
         c[val+120] = 1-abs(val)/40.0 if abs(val)<40 else 0
a=c.tolist()

#below is a sinc filter for 40 samples per symbol.
for val in x:
        c[val+120] = np.sinc(val/40.0)
b=c.tolist()

#-------------------------------------------------------------
```

Notice that specifying the taps parameter of any block as sincfilt.a will give linear interpolation, while sincfilt.b will give sinc interpolation.

4 Some Comments

Since both the main file and the *hier* block use the parameter *sampling rate*, ensure they are set to the same value. Also, if you do not wish to use a hierarchical block, just include the relevant connections in the main grc file itself, and connect the communication signal directly.

By varying the knob for ISI, one can see that the eye moves from an open state to a closed state as ISI is increased.