ELEN90051 Advanced Communication Systems
Digital Communication Systems in GNU Radio
Workshop week 9 (=*week of 30 April*):
Modulation and Demodulation over Band-Limited Channels

Department of Electrical and Electronic Engineering
The University of Melbourne

Created by Xiangyue Meng and Margreta Kuijper

updated on April 26, 2018

All real-world communication channels are band-limited. Waveforms transmitted over a band-limited channel tend to get smeared in time. Consequently, the transmission over one symbol interval will interfere with the transmission over neighbouring symbol intervals, which is called inter-symbol interference (ISI). The **Nyquist Criterion** gives a necessary and sufficient condition that no ISI occurs at the sampling points. One task of designing a digital communication system is to design a pulse-shaping filter that satisfies the Nyquist Criterion.

The two green blocks are going to be investigated in this workshop.

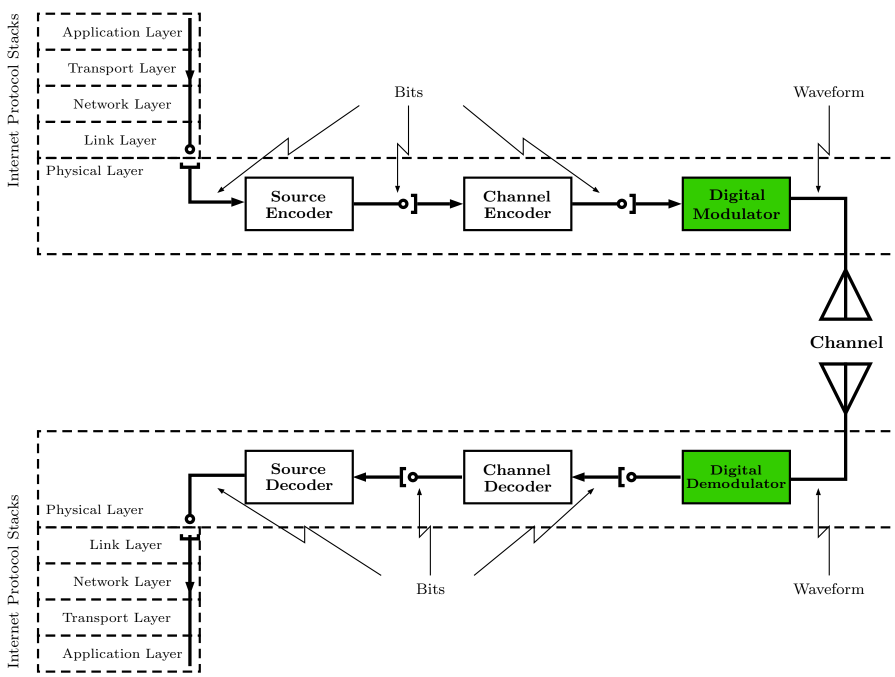


Figure 1: The road map of implementing a digital communication system.

I Objectives and Logistics

In this workshop, you will investigate the effects of a band-limited channel. To counter these effects, you will look at a particular type of pulse-shaping filter, the root raised-cosine (RRC) filter. Specifically, you will

- implement the RRC filter and appreciate its effectiveness of combating ISI;
- observe the eye diagram to understand the effects of band-limited channels.

You are expected to **be prepared before attending the workshop session**. For this, read through the workshop manual. An **individual pre-workshop report**, answering Question 1 to Question 3, worth 10 marks, is to be submitted before the start of the workshop session.
There are **TWO in-workshop check-off points** upon successfully finishing Section III Task 1 and Task 2, worth 5 marks respectively before the end of the workshop.
You are also asked to write a group project report, following the instructions in Section IV. This report is worth 30 marks and should be submitted before the start of your next workshop in week 10 (=week of 7 May). Please read the document "Rules on workshops & report submission" for more information, see `LMS/Workshops/Rules on Workshops and Report Submission/RulesWorkshopsReportsELEN90051.pdf`.

II Background

1 Pulse Shaping

In practical communication systems with band-limited channels, the pulse shaping filter must be chosen carefully not to introduce inter-symbol interference (ISI). Some of the most commonly used pulse shaping filters are:

(a). Rectangular pulse: This pulse shape is easy to implement but has poor spectral properties with large sidelobes.

`ToDo:` **Question 1**. Consider a QPSK signal represented by the equivalent lowpass signal

$$u(t) = \sum_n I_n g(t - nT), \tag{1}$$

where I_n takes on one of the four possible values $\frac{1}{\sqrt{2}}(\pm 1 \pm j)$ with equal probability. Assume that the I_n's are statistically independent.
Determine and sketch the power density spectrum of $u(t)$ when

$$g(t) = \begin{cases} A, & 0 \le t \le T, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

(b). Half-sine pulse: Comparing to the rectangular pulse, the half-sine pulse shape provides better spectral properties with smaller sidelobes.

`ToDo`: **Question 2**. Repeat the previous question when

$$g(t) = \begin{cases} A\sin(\pi t/T), & 0 \le t \le T, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Plot the spectrum in (a) and (b) together in Matlab and compare them in terms of the 3-dB bandwidth and the bandwidth to the first spectral zero, where the 3dB bandwidth is defined as the frequency range where the signal is above half of its maximum power.

(c). Sinc pulse: Theoretically, the sinc pulse has ideal spectral properties, because the Fourier transform of a sinc function has an ideal lowpass spectrum. However, the mainlobe of the sinc pulse is too large. It is non-casual. But a good approximation is a delayed and truncated sinc pulse.

(d). Raised-cosine (RC) pulse: This is the pulse widely used in practice. The pulse shape and the excess bandwidth can be controlled by changing the excess bandwidth factor ($0 \le \beta \le 1$), i.e., the bandwidth occupied beyond the Nyquist bandwidth, which is $\frac{1}{2T}$ for baseband modulation ($\frac{1}{T}$ for bandpass modulation). The frequency-domain expression of the raised-cosine filter (RC) is given by

$$X_{rc}(f) = \begin{cases} T, & 0 \le |f| \le \dfrac{1-\beta}{2T}, \\ \dfrac{T}{2}\left(1 + \cos\left[\dfrac{\pi T}{\beta}\left(|f| - \dfrac{1-\beta}{2T}\right)\right]\right), & \dfrac{1-\beta}{2T} \le |f| \le \dfrac{1+\beta}{2T}, \\ 0, & |f| > \dfrac{1+\beta}{2T}. \end{cases} \tag{4}$$

As β approaches zero, the roll-off zone becomes infinitesimally narrow. The frequency response then approaches an ideal rectangular filter, which is given by

$$X_{\mathrm{rc}}(f) = \begin{cases} T, & |f| \le \dfrac{1}{2T}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

When $\beta = 1$, the non-zero frequency response becomes

$$X_{\mathrm{rc}}(f) = \begin{cases} \dfrac{T}{2}(1 + \cos(\pi f T)), & |f| \le \dfrac{1}{T}, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

which is a pure raised cosine. The impulse response of the RC filter is given in the lecture note.

(e). Root raised-cosine (RRC) pulse: The total equivalent filter of the transmission system is the combination of the transmit and the receive filter, i.e., $g_{\mathrm{TX}} \circledast g_{\mathrm{RX}}$, where $\circledast$ is the convolution operation. This equivalent filter (and not the individual filters) must fulfill the Nyquist criterion. We can achieve this goal if both filters have a transfer function that is equal to the square root of that of the RC filter. Such a filter is therefore called root-raised-cosine (RRC) filter. The combination of the two RRC filters then becomes a RC filter and thus fulfills the Nyquist criterion.

Furthermore, since the filters are real-valued and symmetric, the RRC is its own matched filter. Similar to the RC filter, the RRC filter is characterized by the roll-off parameter β. The impulse response of the RRC filter is given by

$$g(t) = \begin{cases} 1 + \beta \left(\frac{4}{\pi} - 1 \right), & t = 0, \\ \frac{\beta}{\sqrt{2}T} \left(\left(1 + \frac{2}{\pi} \right) \sin\left(\frac{\pi}{4\beta} \right) + \left(1 - \frac{2}{\pi} \right) \cos\left(\frac{\pi}{4\beta} \right) \right), & t = \pm \frac{T}{4\beta}, \\ \frac{\sin\left(\frac{\pi t}{T}(1-\beta) \right) + \frac{4\beta t}{T} \cos\left(\frac{\pi t}{T}(1+\beta) \right)}{\frac{\pi t}{T}\left(1 - \left(\frac{4\beta t}{T} \right)^2 \right)}, & \text{otherwise.} \end{cases} \tag{7}$$

`ToDo:` **Question 3**. Show that the impulse response of a filter having a square-root raised cosine spectral characteristic is given by (7). Plot the impulse response of the RC and the RRC filter in together in Matlab and make comparisons.

Note that the spectra of the RRC filter and the RC filter show the same excess bandwidth. And they are also non-causal.

2 Eye Diagram

The eye diagram is often used to visualize the imperfections of modulated transmission through a band-limited channel. It is generated by superimposing the plots of time-domain signals for a number of symbols, as shown in Figure 2.

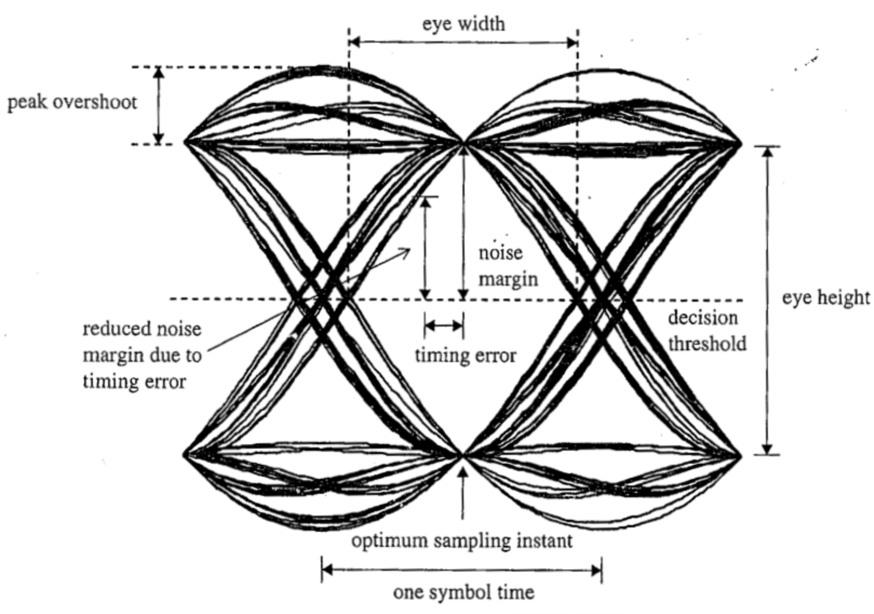


Figure 2: Eye diagram and its features. (Rice 2009)

The vertical opening at the sampling time, called the "eye height", represents the amount of ISI at the sampling point. The horizontal opening, called the "eye width", indicates how sensitive the receiver is to errors in sampling point timing.

For raised-cosine signals, the larger the β, the wider the opening. A smaller β will lead to larger errors if not sampled at the best sampling time which occurs at the center of the eye.

III In-Workshop Tasks (*10 marks*)

In this workshop, you are supposed to complete the following two tasks:

(1) Implement the digital RRC filter;

(2) Investigate the effects of the band-limited channel and observe the eye diagram.

Task 1: Implement the Digital RRC Filter

Let us first understand the effects of band-limited channel on the transmitted signals. You will need the flowgraph that you implemented in the Modulation workshop.

Step 1: Clean the Flowgraph
In this workshop, using baseband signals is sufficient to understand the effects of the band-limited channel.

`ToDo:` Save `.grc` of the Modulation workshop to another file and delete all blocks that relate to the carrier, including the sinks. Delete the **reference** `Interpolating FIR Filter`, i.e., the one with `Taps` as 1.

Step 2: Implement the Rectangular Pulse Shaping Filter
Different from what we did in the Demodulation workshop, here we evenly distribute the DC gains of the pulse shaping filter and the matched filter at the transmitter and receiver side respectively.

Recall that the DC gain of a FIR filter is given by

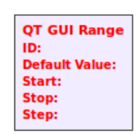
$$G = \sum_{n=0}^{N-1} h[n],$$

where N is the number of the taps and $h[n]$ is the coefficients of the impulse response.

`ToDo:` Set `Taps` of the rectangular pulse shaping filter to `1./G*np.ones(sps)`, where `G` is a proper value such that the total gain of the system is normalized to 1.

Step 3: Implement the Root Raised-Cosine Pulse Shaping Filter
There are many ways to implement the RRC filter in GNU Radio. They all boil down to the task of calculating the taps of the RRC filter. In this workshop, we use the built-in `Root Raised Cosine Filter` block.

`ToDo:` Drag and drop a `QT GUI Range` block into the flowgraph. Double click on the block and set the parameters as hinted in Table 1.

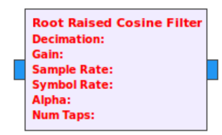


A `QT GUI Range` block in GRC

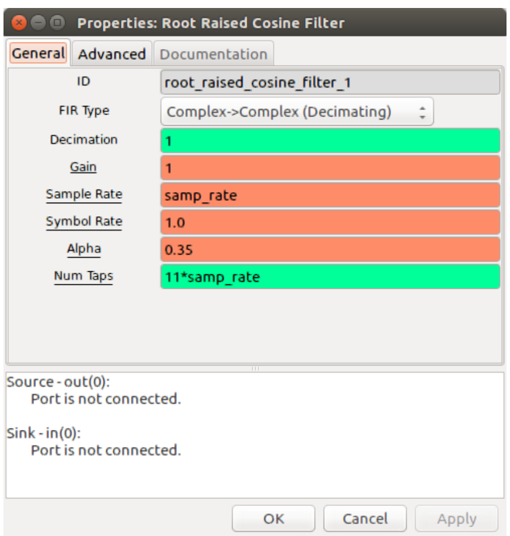
Name	Value
`ID`	`excess_bw`
`Label`	`Excess Bandwidth`
`Type`	`Float`
`Default Value`	`0.35`
`Start`	`0.01`
`Stop`	`0.99`
`Step`	`0.01`
`GUI Hint`	`1,0,1,1`

Table 1: Parameters for `QT GUI Range`

`ToDo:` Drag and drop a `Root Raised Cosine Filter` block into the flowgraph. Double click on the block and set the parameters as hinted in Figure 3.



A `Root Raised Cosine Filter` block in GRC



Name	Value
`FIR Type`	*A proper selection*
`Interpolation`	The up-sampling rate
`Gain`	*A proper value*
`Sample Rate`	`sps`
`Alpha`	`excess_bw`
`Num Taps`	`11*sps`

Figure 3: Setting parameters for `Root Raised Cosine Filter` block in GRC.

Here, the `QT GUI Range` controls the value of `excess_bw` and therefore controls the excess bandwidth of the `Root Raised Cosine Filter`.

`ToDo:` Connect the output of the `Throttle` block to the input of the rectangular pulse shaping filter and the RRC pulse shaping filter respectively.

Step 4: Implement the Band-Limited Channel Model

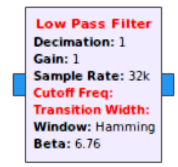
In this workshop, we model the channel as a band-limited channel with AWGN noise.

`ToDo:` Drag and drop a `QT GUI Range` block into the flowgraph. Double click on the block and set the parameters as hinted in Table 2.

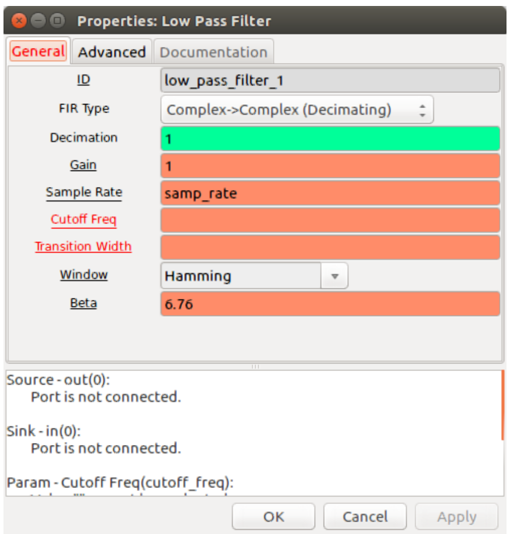
Name	Value
ID	`chn_bw`
Label	`Channel Bandwidth`
Type	`Float`
`Default Value`	`samp_rate*sps/2`
Start	`10000`
Stop	`samp_rate*sps/2`
Step	`100`
`GUI Hint`	`1,1,1,1`

Table 2: Parameters for `QT GUI Range`

`ToDo:` Drag and drop a `Low Pass Filter` block into the flowgraph. Double click on the block and set the parameters as hinted in Figure 4.



A `Low Pass Filter` block in GRC



Name	Value
`FIR Type`	*A proper selection*
`Decimation`	1
`Gain`	1
`Sample Rate`	`samp_rate*sps`
`Cutoff Freq`	`chn_bw`
`Transition Width`	100

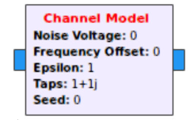
Figure 4: Setting parameters for `Low Pass Filter` block in GRC.

`ToDo:` Drag and drop a `QT GUI Range` block into the flowgraph. Double click on the block and set the parameters as hinted in Tablee 3.

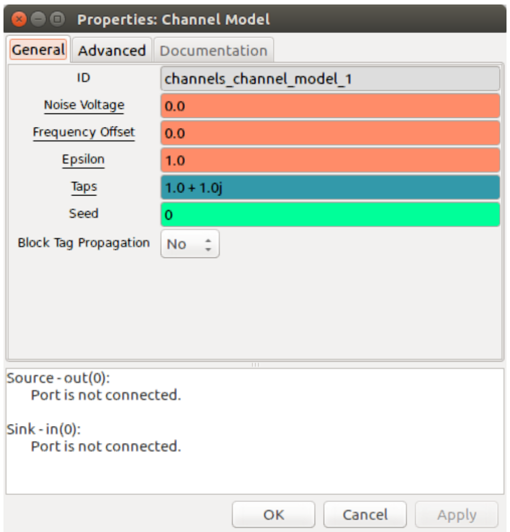
Name	Value
ID	`noise`
Label	`Noise Voltage`
Type	`Float`
`Default Value`	`0`
Start	`0`
Stop	`1`
Step	`0.01`
`GUI Hint`	`2,0,1,1`

Table 3: Parameters for `QT GUI Range`

ToDo: Drag and drop a `Channel Model` block into the flowgraph. Double click on the block and set the parameters as hinted in Figure 5.



A `Channel Model` block in GRC



Name	Value
`Noise Voltage`	`noise`
`Frequency Offset`	`0`
`Taps`	`1`

Figure 5: Setting parameters for `Channel Model` block in GRC.

ToDo: Connect the rectangular pulse shaping filter, the `Low Pass Filter` block, and the `Low Pass Filter` block in series. Build the same branch for the RRC pulse shaping filter.

Step 5: Implement the Matched Filter

We already know that the rectangular pulse shaping filter and its matched filter are the same.

ToDo: Copy the rectangular pulse shaping filter and set the field `Interpolation` to `1`. Connect its input to the output of the corresponding `Channel Model` block. Do the same for the other branch.

Step 6: Align the Samples

Since the rectangular pulse shaping filter and the RRC pulse shaping filter have different numbers of taps, the samples out of the two branches have different delays.

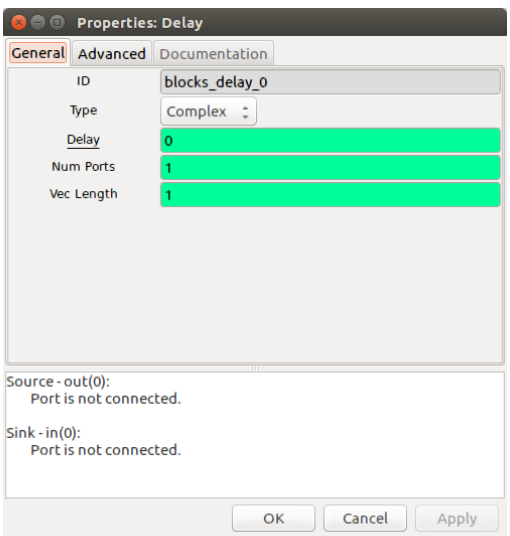
ToDo: Drag and drop a `QT GUI Range` block into the flowgraph. Double click on the block and set the parameters as hinted in Table 4.

Name	Value
`ID`	`delay`
`Label`	`Sample Delay`
`Type`	`Int`
`Default Value`	`0`
`Start`	`0`
`Stop`	`100`
`Step`	`1`
`GUI Hint`	`2,0,1,1`

Table 4: Parameters for `QT GUI Range`

`ToDo:` Drag and drop a `Delay` block into the flowgraph. Double click on the block and set the parameters as hinted in Figure 6.

A `Delay` block in GRC



Name	Value
`Type`	*A proper selection*
`Delay`	`delay`

Figure 6: Setting parameters for `Delay` block in GRC.

Step 7: Observe the Output

`ToDo:` Add a `QT GUI Constellation Sink` block, a `QT GUI Frequency Sink`, and a `QT GUI Time Sink` block with the field `Number of Inputs` setting to 2 and the field `GUI Hint` setting to `0,0,1,1`, `0,1,1,1`, and `3,0,1,2` respectively.

`ToDo:` Connect those plot sinks to the two branches properly.

`ToDo:` Run the flowgraph, change the value of `Excess Bandwidth`, and observe the modular T and frequency plot. A sample output is provided for your reference in Figure 7.
What is the value of `Excess Bandwidth` when the spectrum of the output of the RRC pulse shaping branch overlaps with the main lobe of the spectrum of the output of the rectangular pulse shaping branch?
What is the value of `Excess Bandwidth` when the spectrum of the output of the RRC pulse shaping branch approaches an ideal rectangular

filter?
Confirm that your observation is consistent with the theory in Section II.

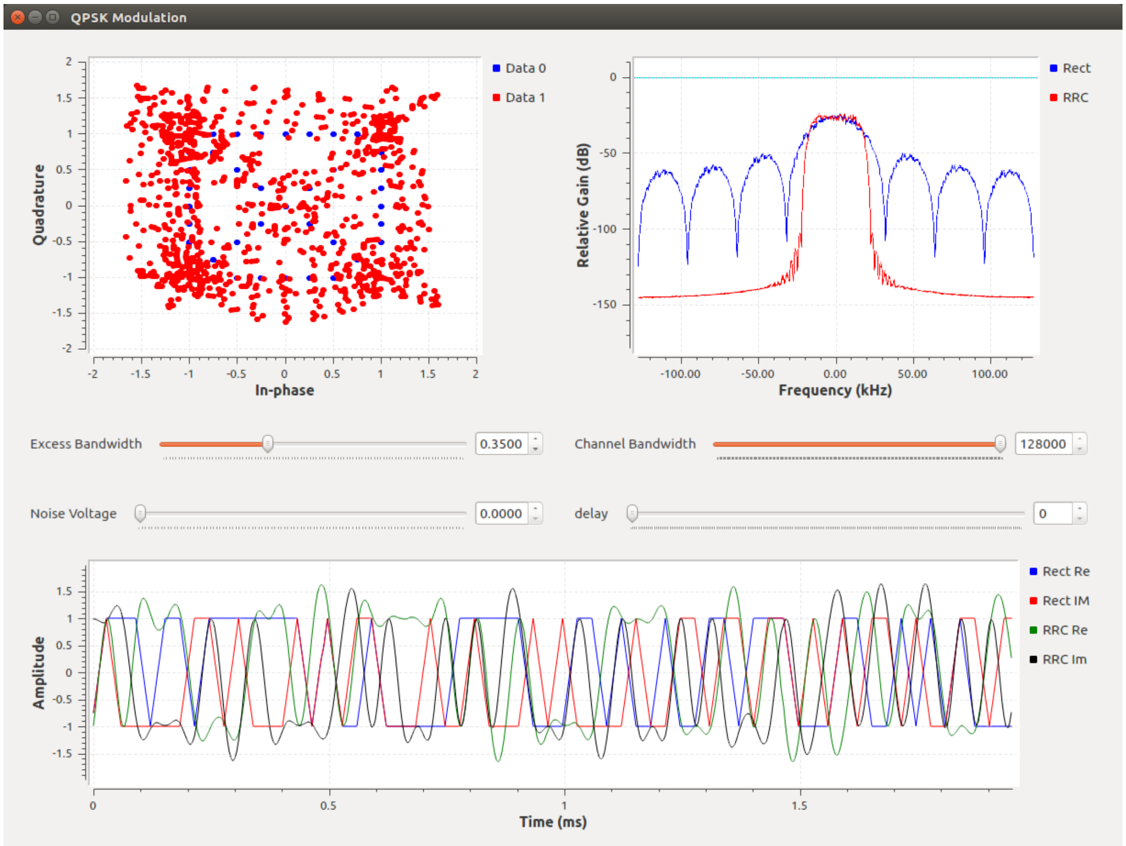


Figure 7: A sample output for this section

Step 8: Verification

Show your working flowgraph to the demonstrator to check off the in-workshop assessment. You may be asked some questions relating to your flowgraph.

Task 2: Investigate the Effects of the Band-Limited Channel

Step 1: Effects of the AWGN Noise

`ToDo:` Execute the flowgraph and drag the slider of the `Noise Voltage`. What are the effects of the value of `noise` on the Modular T, frequency, and time plot? Describe your observation in detail.

Clicking the middle button of the mouse on the plot will open the context menu.

Step 2: Effect of Channel Bandwidth

`ToDo:` (1) Execute the flowgraph. Change the `Excess Bandwidth` to the value when the spectrum of the output of the RRC pulse shaping branch overlaps with the main lobe of the spectrum of the output of the rectangular pulse shaping branch.

(2) Make sure that the value of the `Noise Voltage` is zero.

(3) Turn off the traces of the imaginary part of the output of the rectangular and RRC pulse shaping branches by clicking on their legends in the time sink.

(4) Experimentally determine the value of `Delay` such that the traces of the real part of the rectangular and RRC pulse shaping branches perfectly overlap with each other.

(5) Middle-click on the time sink. Change `Number of Points` to 64, and click on `Stem Plot` to change the plot type.

(6) Stop the time plot. You should see an output similar to Figure 8.

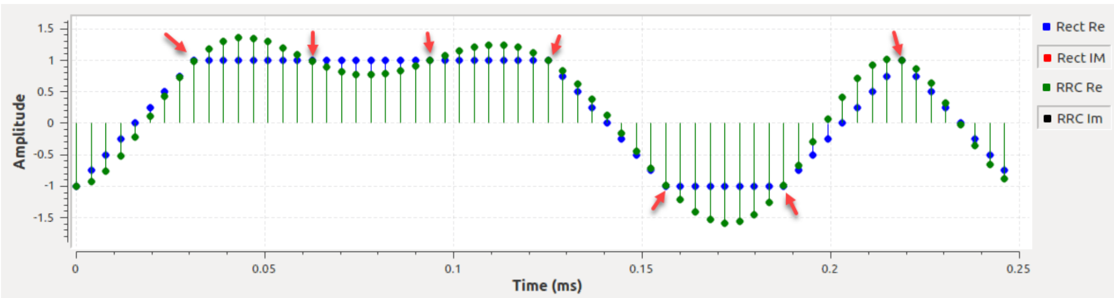


Figure 8: A sample output of the time sink.

Look for the the points that exactly lie on the value 1 and -1. Since the up-sampling rate is 8, there should be seven points in between. As we can see from Figure 8, apparently there are more points from the rectangular pulse shaping branch that lie on the value 1 and -1. Does this imply that the rectangular pulse is better than the RRC pulse?

(7) Reduce `Channel Bandwidth` to the value when the channel only passes the main lobe of the spectra of the output of the rectangular pulse shaping branch and repeat (5) and (6). You should see an output similar to Figure 9.

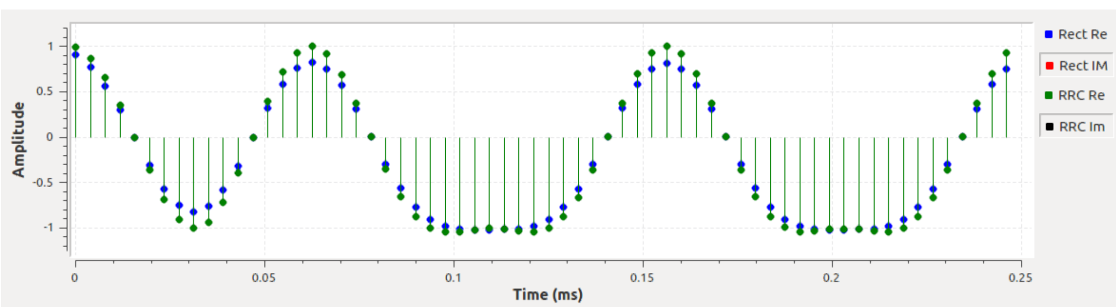


Figure 9: A sample output of the time sink.

Can you still find, for both branches, the points that exactly lie on the value 1 and -1? How does the output indicate that the ISI occurs? How does the RRC pulse help to remove the ISI?

(8) Reduce `Channel Bandwidth` further down and repeat (7).

Step 3: Observe the Eye Diagram

In this step, we will reproduce Figure 2 and understand the effects of the effect of the bandwidth limitation of the channel via the eye diagram.

`ToDo:`

(1) Double click on `Options` block and change the value of the field `Generate Options` to `WX GUI`.

(2) Disable all `QT GUI` components by right-clicking on the block and selecting `Disable`.

(3) Replace all `QT GUI Range` with `WX GUI Slider` and set the parameters accordingly.

(4) Drag and drop a `WX GUI Scope Sink` and a `WX GUI FFT Sink` block into the flowgraph and set the parameters accordingly.

(5) Connect the `WX` sinks to the rectangular pulse shaping branch and terminate the other branch by using a `Null Sink` block.

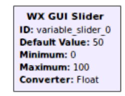
(6) Execute the flowgraph. Click the tab of `Ch2` on the control panel of the time sink. Select `None` in the dropdown list of `Marker`.

(7) Drag `T Offset` up to the maximum value. Tick `Persistence` and drag `Analog Alpha` down to a small value. Now you should see the eye diagram of the I channel of the received QPSK signals.

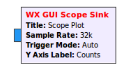
(8) Change the value of the `Noise Voltage` and the `Channel Bandwidth`, and observer the eye diagram. Describe your observations in detail and draw conclusions from your observations.

(9) Repeat (5) - (7) for the RRC pulse shaping branch. Change the values of `Excess Bandwidth` to the maximum value, and repeat (8). In particular, compare the eye diagram corresponding to the rectangular pulse with the eye diagram corresponding to the RRC pulse. Draw conclusions from your observations.

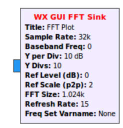
`QT GUI` and `WX GUI` are two types of GUI framework implemented in `C++`. However, `WX GUI` will be completely dropped with the next release of GNU Radio 3.8.



A `WX GUI Slider` block in GRC



A `WX GUI Scope Sink` block in GRC



A `WX GUI FFT Sink` block in GRC

Step 4: Verification

Show your working flowgraph to the demonstrator to check off the in-workshop assessment. You may be asked some questions relating to your flowgraph.

IV Report Tasks (*30 marks*)

You should structure your report as follows:

1. Introduction: Introduce your report, e.g., what are the imperfections of physical channels; what approaches can be used to combat those imperfections; why the RRC filter is needed in practice.

2. Background: Give necessary knowledge and mathematical background related to your topic, e.g., what are the mathematical and other tools of doing the analysis.

3. Theoretical analysis: Similar to the pre-workshop questions, derive theoretical expressions of the RRC filter; what are the trade-offs of using different values of excess bandwidth of a RRC filter.

4. Implementation: Describe your implementation.

5. Observation: What is the impact of channel bandwidth and noise on the received signals; how does the eye diagram indicate the imperfections of physical channels.

6. Conclusion: Draw conclusions based on your analyses and observations. In particular, compare your theoretical analysis with the observations that you found and discuss the effect of reduced channel bandwidth and increased noise levels.

7. Reference: Use IEEE reference style. See `LMS/Workshops/Workshop week 3: source coding 2018/ieeecitationref.pdf` for details.

End of Workshop