Digital Communications

Lab work texts

Bachelor's degree 3: Telecommunications

Faculty of Electrical Engineering

Houari Boumediene University of Science and Technology

https://perso.usthb.dz/~akourgli/Communications%20Num%c3%a9riques/

Content

Introduction :

Various reminders on digital signals and random processes

Chapter 1. Digital Baseband Transmission

Elements of a digital transmission chain, baseband modulation. Line codes (Bit/symbol conversion and shaping), Polar NRZ code, Unipolar NRZ code, Unipolar RZ code, Biphase / Manchester code, HDB3 code (High Density Bipolar of order 3), M-ary line codes (M-ary NRZ codes), Power spectral density of line codes, Criteria for choosing a line code. Concept of complex envelope.

Chapter 2. Optimal Receiver

Structure of a receiver with M signals, vector representation of signals and noise, optimal detection (MAP detector for maximum a posteriori and ML detector for maximum likelihood), Structure of the optimal receiver (autocorrelation or adapted filtering on each of the channels then decision).

Chapter 3. Interference-free transmission between symbols

Effect of Channel on Line Code Waveform, Characteristics of Intersymbol Interference, Eye Diagram, Condition of No Intersymbol Interference, Nyquist Criterion , Raised Cosine Filter, Error Probability Performance of an M-ary System with Nyquist Filtering , Distribution of Filtering Between Transmission and Reception.

Chapter 4. Performance for baseband transmission

Binary signal detection and hypothesis testing, maximum likelihood criterion, likelihood ratio, optimal binary receiver with two correlators, single correlator and matched filter. Error probability for the case of white Gaussian noise with low-pass filter and matched filter.

Chapter 5. Narrowband Digital Modulations

Principle, Amplitude shift keying (ASK), OOK modulation, Symmetrical M-ASK modulations, Physical realization and performance, Phase shift keying (PSK), Constellations, M-PSK modulations, Physical realization and performance, Quadratic dual-carrier modulation (QAM), Physical realization and performance, Frequency shift keying (FSK), MSK modulation, Physical realization and performance of a binary FSK.

LAB n°1: Getting Started with GNU Radio

Purpose of the LAB: To become familiar with the GNU Radio Companion (GRC) environment , the creation of flowgraphs and the manipulation of sinusoidal and noisy signals.

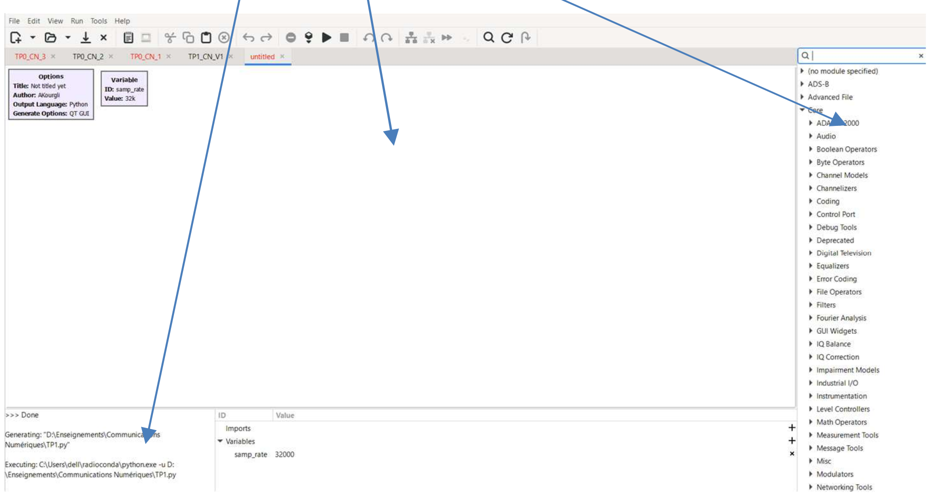
Educational objectives:
- Understanding the structure of a GRC flowgraph .
- Generate and visualize a noisy sinusoidal signal.
- Apply low-pass filtering and analyze the results in time and frequency.
- Manipulate the different types of blocks and understand their configuration **.**

1. GNU Radio
GNU Radio is an **open-source** signal processing framework used in:
- Wireless communication systems (software defined radios)
- Research, teaching and prototyping
- Analysis of analog and digital signals

GNU Radio Companion (GRC) is a **graphical interface** that allows you to create **flowgraphs** by dragging and dropping **functional blocks** : Source → processing → visualization

Installing GNU Radio https://wiki.gnuradio.org/index.php/InstallingGR
Once installed, launch the software: gnuradio-companion

You then access the GRC interface composed of:
- A **library of blocks** on the left
- A **drawing area** in the center (flowgraph)
- A **console** at the bottom for logs

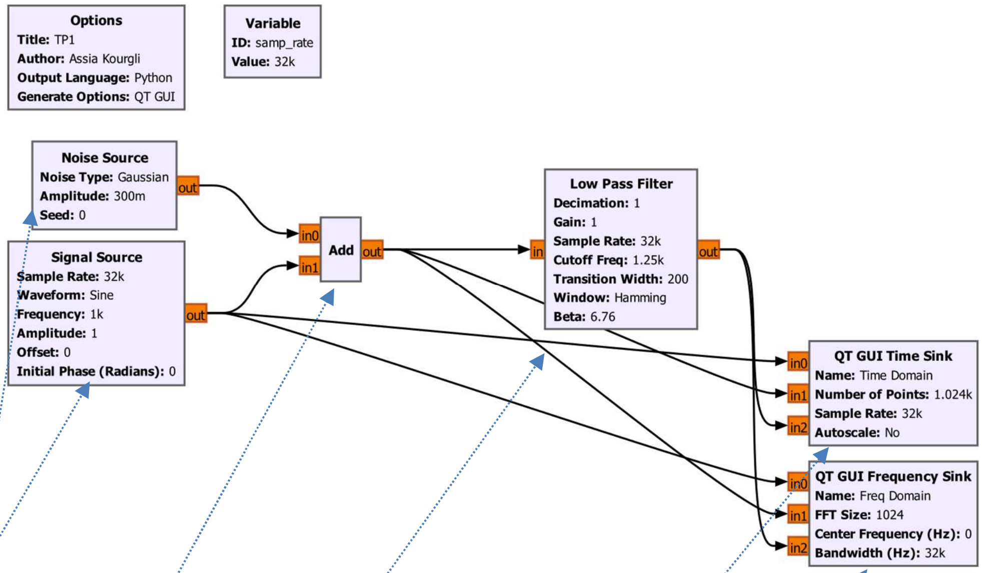


Typical blocks to know from the start

Block	**Role**
• Vector Source / Random Source	Data generation
• Throttle	Flow control (avoids CPU overload)
• QT GUI Time Sink	Time domain display
• QT GUI Frequency Sink	Frequency Domain Display (FDD)
• Add , Multiply , Repeat	Basic treatments
• Selector , Mux	Signal Routing and Switching

.grc files
- Each LAB is delivered in the form of a . grc file
- To open it: **File > Open** in GNU Radio Companion
- To run: click the ▶ button at the top or press Ctrl+R

2. Manipulation 1: Filtering a noisy sinusoid → Flowgraph to be made



Block 1: Signal Source
- Output Type: Float
- Sample Rate: samp_rate
- Frequency: 1000
- Amplitude: 1.0
- Waveform: SINE

Block 2: Noise Source
- OutpBut Type: Float
- Amplitude: 0.3
- Sample Rate: 32000

Block 3: Add
- I/O Type: float
- Inputs: 2

Connect the Signal Source and Noise Source outputs to this block.

Block 4: Low Pass Filter
- FIR Type: Float →Float Decimating
- Decimating: 1
- Gain: 1
- Sample Rate: samp_rate
- Cutoff Freq: 1250
- Transition Width: 200
- Window: Hamming

Connect the output of Add to this filter.

Block 5: QT GUI Time Sink
- Inputs: 3
- Sample Rate: samp_rate
- Autoscale : No

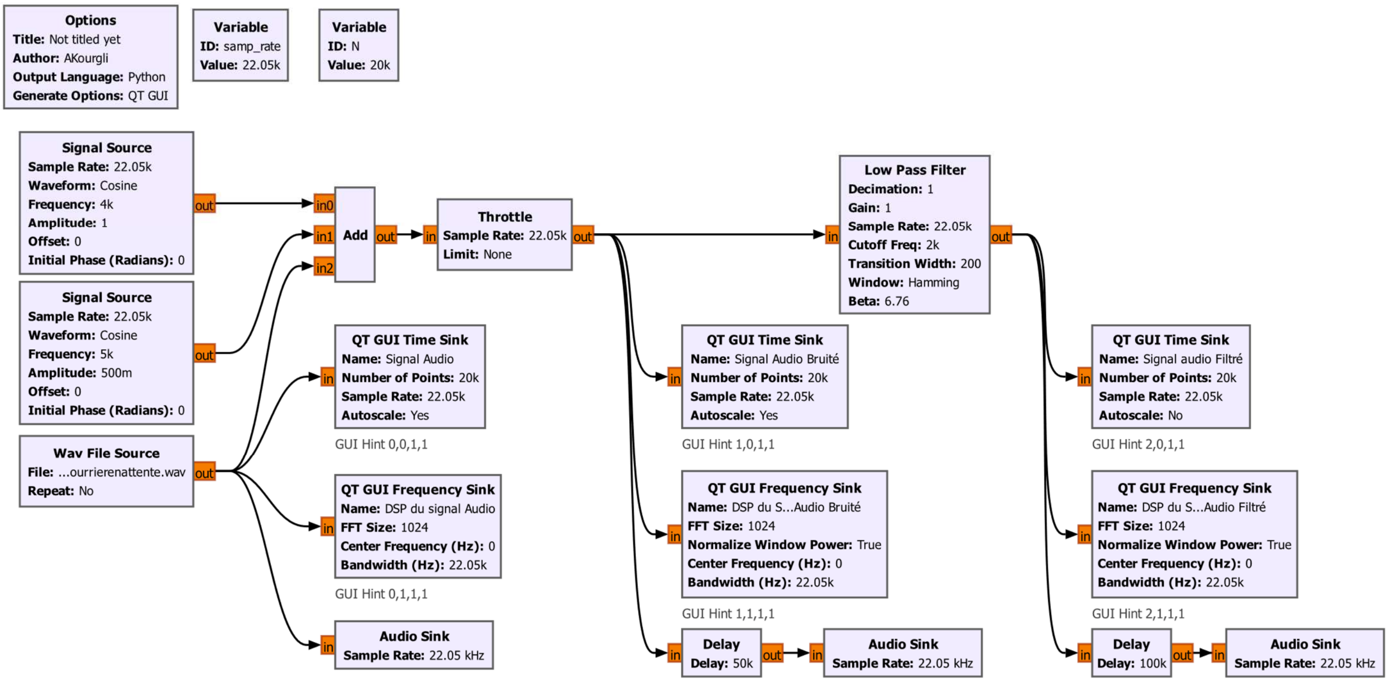
Connect the Signal Source (direct) output to input 0
Connect the filter output to input 1

Block 6: QT GUI Frequency Sink
- Inputs: 3
- Spectrum Width : Full
- FFT Size: 1024
- Center Frequency (Hz): 0
- Bandwidth: samp_rate

Connect the Signal Source output, then the Low Pass output Filter

- Explain the role of each block.
- Explain the role of each Low pass parameter filter →Documentation→ Visit Documentation Page.
- Do the same for QT GUI Time Sink and QT GUI Frequency Sink .
- Vary the noise power and observe.
- Rework the diagram using: Virtual Sink and Virtual Source for each of the 3 signals.
- Explore the role of the buttons on the menu:

3. Manipulation 2: Denoising an Audio Signal→ Flowgraph to be made



- Copy the audio file 'you have mail waiting.wav' to the working directory.
- Explain the role of each block.
- Use the comments placed below the blocks for the distribution of figures.
- Run and comment.

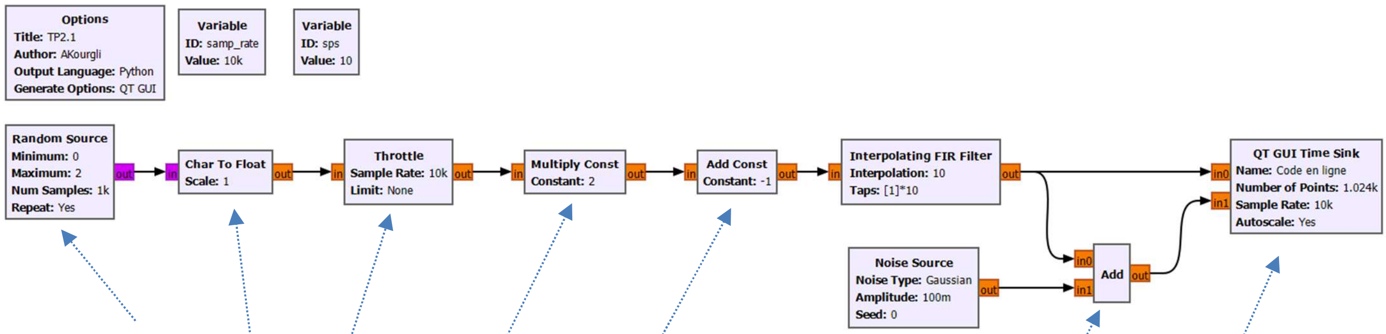
LAB n°2: Line coding

Aim of the LAB: Explore the different line coding techniques (NRZ, RZ, Manchester, M-ary)

Educational objectives :

- Implement line coding schemes suitable for digital transmissions.
- Visualize the time structure of binary codes.
- Analyze the power spectral density of each code.
- Compare performance and spectral efficiency.

1. Manipulation 1: Visualization of a noisy binary sequence→ Flowgraph to be made



Block 1: Random Source (binary source)
- Minimum: 0
- Maximum: 2
- Repeat: Yes
- Output Type: Byte

Block 2: Char To Float
- Scale: 1.0

Block 3: Throttle
- Sample Rate: samp_rate

Block 4: Multiply Const
- Constant: 2

Block 5: Interpolating FIR Filter
- Interpolation: sps
- Taps = [1] * sps

Block 6: Noise Source
- Noise Type: Gaussian
- Amplitude: 0.1

Block 7: Add
- Inputs: 2
- Connect the outputs of Vector Source and Noise Source to this block

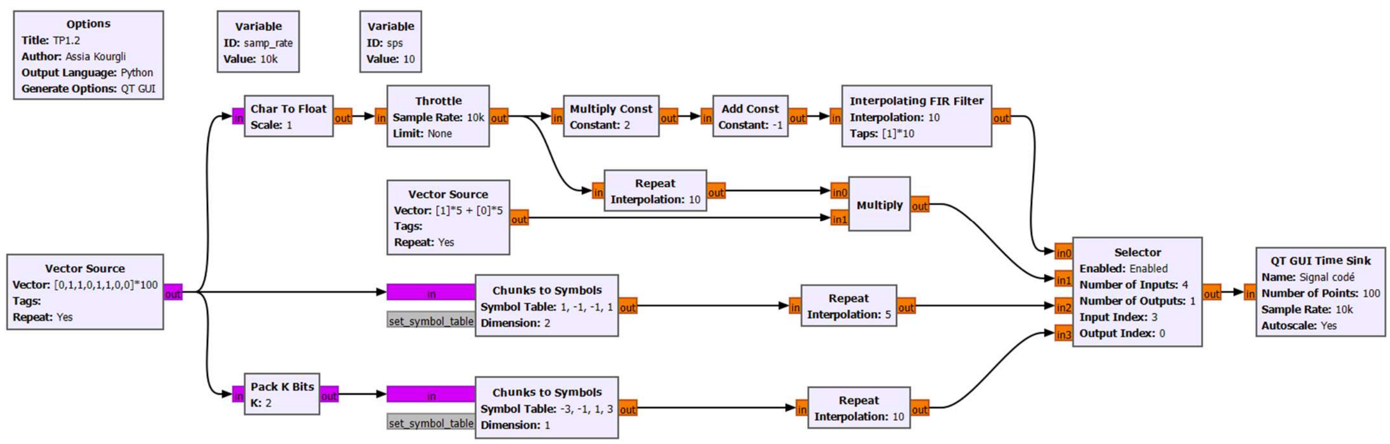
---

Block 8: QT GUI Time Sink
- Inputs: 2
- Sample Rate: samp_rate

Connect the output of Interpolating FIR Filter to input 0
Connect the output of add to input 1

- Explain the role of each block.
- QT GUI Time Sink →Config →Control Panel = Yes.
- Change the value of ' sps ' and observe.
- Change Number of points →100 and observe
- What is the role of this flowgraph ?

2. Manipulation 2: Generation of independent line codes→ Flowgraph to be made



Variables: samp_rate = 10kHz and sps =10
Vector Source (binary)
→ Char to Float
→ Throttle (limits the flow)
→ 4 coding branches (NRZ, RZ, Manchester, M-ary)
→ Selector (choice of active coding : input_index = 0 or 1 or 2 or 3)
→ QT GUI Time Sink (visualization) (Autoscale = Yes)

Polar NRZ (Non-Return to Zero) branch
- Multiply Const (value = 2) Gives 0 → 0, 1 → 2
- Add Const (value = -1) Shift → 0 → -1, 1 → +1
- Interpolating FIR Filter Taps = [1]* sps , Interp = sps → stretches each bit over ' sps ' samples
- Selector (input 0)

Unipolar RZ branch (Return to Zero)
- Repeat (N = sps) Each bit becomes ' sps ' samples
- Multiply with Vector Source (values [1]*5 + [0]*5) Applies the RZ shape
- Selector (input 1)

Manchester Branch
- Map or Chunks to Symbols Map : [1, -1, -1, 1]
- Size: 2→ Map : 0 → [1, -1], 1 → [-1, 1
- Repeat (N = sps //2) Stretches each half-bit
- Selector (input 2)

M-ary branch (e.g. 4-ary)
- Pack K bits K=2 →Groups the bits (2 by 2)
- Chunks to Symbols Map 00,01,10,11 → - 3 , - 1 , + 1 , + 3
- Size: 1
- Repeat (N = sps) Stretches each symbol
- Selector (input 3)

- Take Number of points = 100 and calculate Tb and Ts for each code.
- Replace the Vector Source block (binary)→ Random Uniform Source (min=0 and max=2).
- Add a QT GUI Frequency Sink block to visualize the DSP.
- For each code, compare the DSPs obtained with those calculated theoretically.
- Modify the flowgraph to view unipolar NRZ, polar RZ, 8-Area codes.

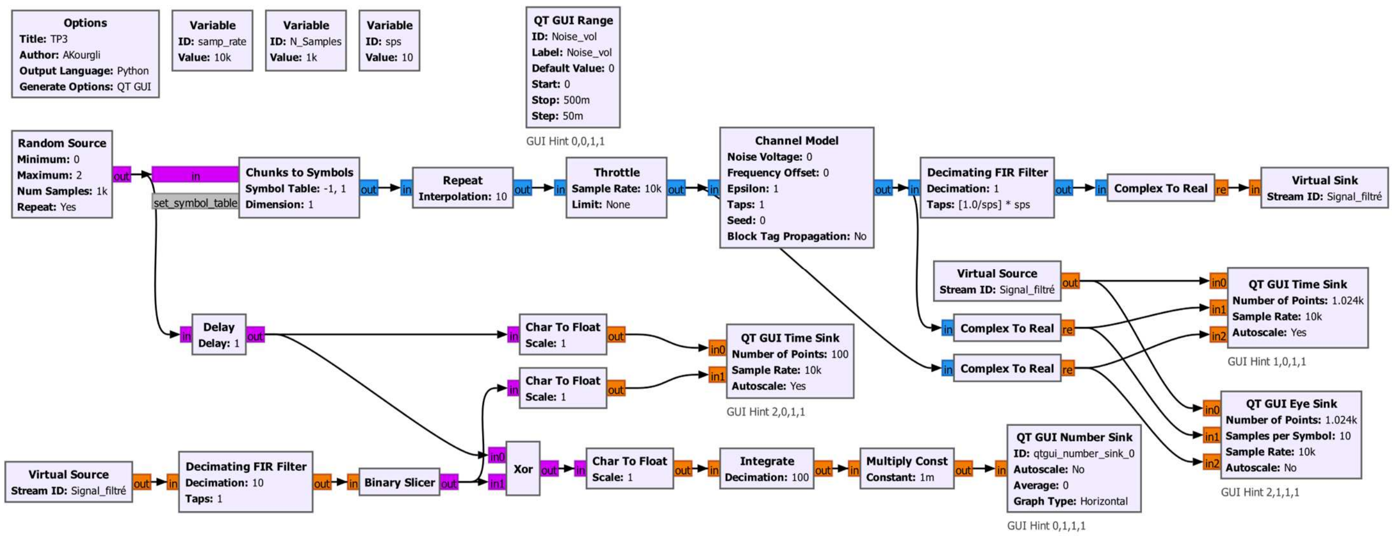
LAB n°3: Baseband transmission and optimal receiver

Purpose of the LAB: Understand the effects of noise on baseband transmission and study the role of adapted filtering in optimal detection.

Educational objectives:

- Building an end-to-end transmission chain.
- Identify the appropriate formatting and filtering blocks.
- Observe the effects of noise on the waveform and BER.
- Experiment with the impact of channel spectral limitation.

1. Manipulation 1: Transmission on an unlimited AWGN channel→ Flowgraph to be made

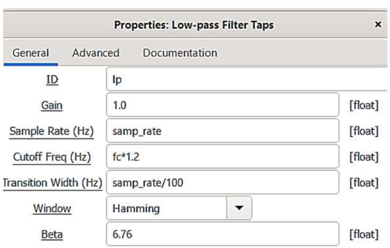


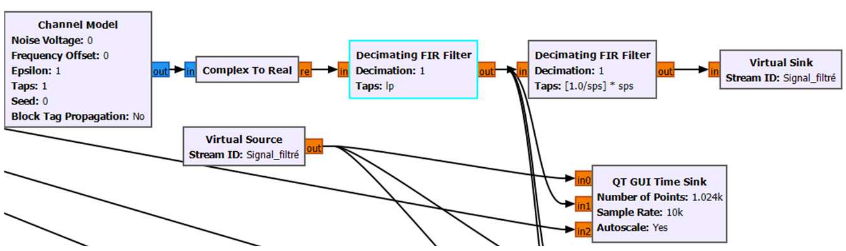
- Briefly explain the role of each block.
- What binary code is used?
- Identify the block that acts as the formatting filter and determine Tb.
- Identify the block that acts as the adapted filter
- Why did we take the value of Taps of [1.0/ sps] * sps in the 1st Decimation block ? filter ?
- Why was a delay of 1 added to the binary sequence?
- Identify the 3 graphs used in the 1st QT GUI Time Sink block by putting the correct titles in Config.
- Do the same for the 2nd QT GUI Time Sink .
- Why was a value of sps taken for Decimation in the 2nd Decimation block ? filter ?
- Why did we take a value of sps //2 for sample delay in the 2nd Decimation block filter ?
- Increase the noise power (at the top of the graph time sink / Widget = Entry+slider) and observe the BER value.
- Increase the noise power and observe the eye diagrams.

2. Manipulation 2: Transmission on a band-limited AWGN channel→ Flowgraph to be made

To simulate a band-limited channel, a low-pass filter is added to the channel output. →The received signal is now recovered at the output of the low-pass filter.

- Add a variable fc = samp_rate / (2 sps) corresponding to 1/(2 Tb)
- Add a low-pass filter after the Channel Model block with fc and Transition Width = samp_rate / 100.
  To do this:
  1. Generate a block that contains the low-pass filter specifications as follows:





Give it an identifier: lp for example, then add a Decimating FIR Filter block by typing in tps: lp

2. Recover the signal received at the filter output and no longer at the channel output (Change source of the middle arrow).
3. Create a variable that gives the added low-pass filter length Len_lp : len (lp)

- Observe the offset between the transmitted and received bits then adjust the Delay value: int ((Len_lp)/2/ sps)+1.
- Observe the effect on the received signal and explain the change in shape.
- Observe the effect on BER and eye diagram.
- Add a QT GUI Frequency Sink to visualize the 3 DSPs and observe the effect of channel limiting.
- Deduce whether the previously defined reception filter is still adequate? suitable?
- Take Cuttof Freq : fc *1.2 then fc * 0.8 and observe again the effect of BER and the eye diagram.
- Deduce the condition on the minimum frequency band allowing one eye to be open.
- Take zero noise power and Cutoff Freq : fc *1.2. Check for interference affecting BER.
- What should be done to reduce interference?
- Explain why this delay value ' int ((Len_lp)/2/ sps)+1'.

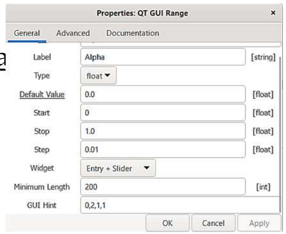
LAB n°4: Baseband transmission on a limited channel without IES

Purpose of the LAB: Carry out a transmission without intersymbols interference (IES) using raised cosine filters (RRC) in transmission and reception.

Educational objectives:

- Nyquist criterion to avoid IES.
- Implement symmetric RRC filtering.
- Observe the effects of roll-off (α) and noise on signal quality.
- Dynamically adjust delays to synchronize signals.

1. Manipulation 1: Transmission on an AWGN channel without IES→ Flowgraph to be ma

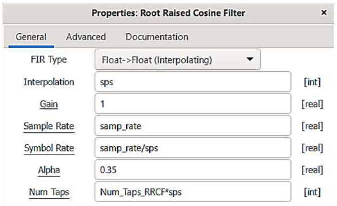


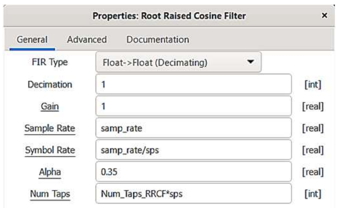
Take the 2nd one again flowgraph from the previous LAB.

1. Add a QT Gui Range to vary alpha on the time sink .
2. Add the following variables:

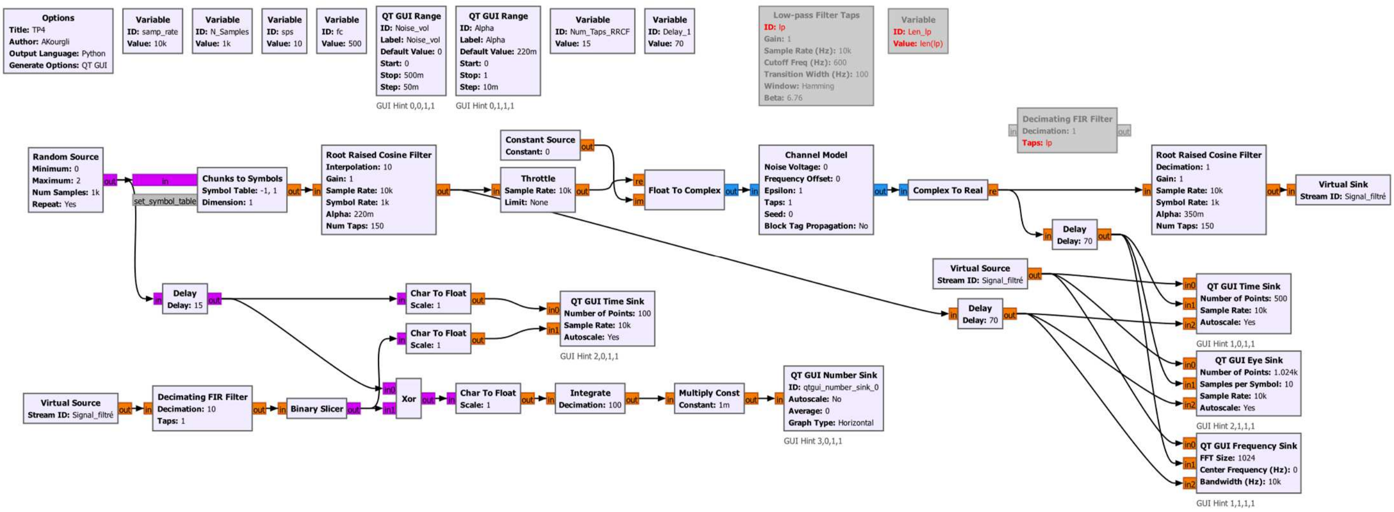
- Num_Taps_RRCF : 11
- Delay_ 1: int(Num_Taps_RRCF * sps / 2)

3. Add the following blocks: The 1st will replace the repeat block and the 2nd will replace the decimating block filter .





4. Change the value of Delay to ' Num_Taps_RRCF '.
5. Add delays of an amount of 'Delay_1' before viewing the transmitted and received signals.
6. Pass Block Filter



- Explain the role of ' Num_Taps_RRCF '.
- Observe the spectral occupancy of the 3 signals and explain.
- Observe the effect of a noise amplitude of 0.1 on the eye diagram of the received and filtered signals.
- Increase the noise gradually (on the graph) and observe the effect on the BER and the eye diagram.
- Vary the value of alpha (on the graph) and observe the impact on the 3 displays. Also try 0 and 1
- delay value of ' Num_Taps_RRCF+int ((Len_lp)/2/ sps)' applied to the binary sequence?

2. Manipulation 2: Transmission on a limited AWGN channel without IES→ Flowgraph to be made

Reconnect the Decimation FIR Filter block (Taps : lp) and adjust:

- Delay: ' Num_Taps_RRCF+ int ((Len_ lp) /2/ sps)'.
- Delay_1: ' int (Num_Taps_RRCF /2)* sps' before displaying the transmitted and received signals
- Delay_2: ' int (Len_lp /2)+Delay_1' before displaying the transmitted and received signals
- Recover the signal received at the output of the added filter
- Vary the Cutoff Frequency of the Cutoff Filter Freq : fc *1.2 then fc * 0.8 and observe the effect on the eye diagrams and BER.

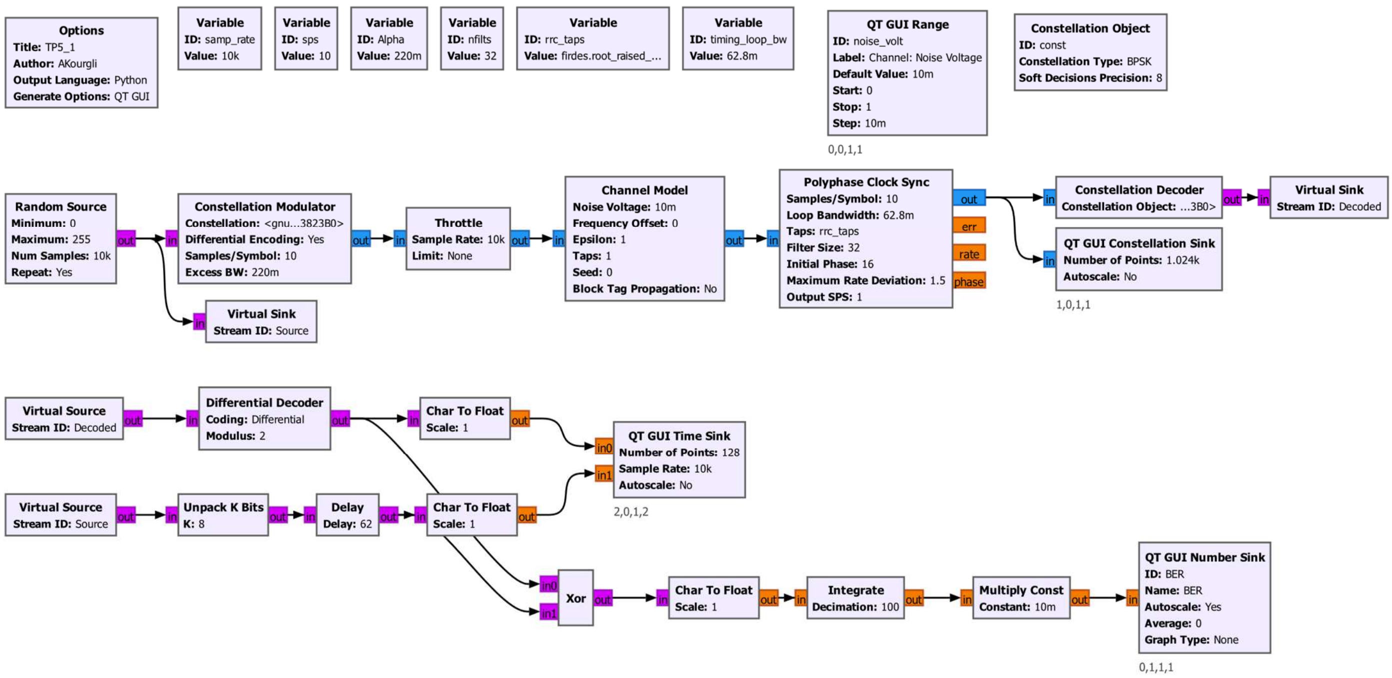
Lab n°5: Digital Modulation

Purpose of the LAB: To put into practice M-ASK, M-PSK, M-QAM and M-FSK modulations and observe their properties.

Educational objectives:

- M-ary modulations with different schemes.
- Visualize the constellation of modulations.
- Analyze their spectral efficiency and noise sensitivity.
- Compare performance via BER.

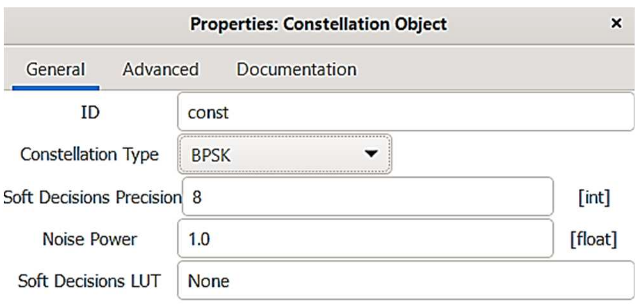
1. Manipulation 1: 2-ASK/BPSK modulation→ Flowgraph to be made

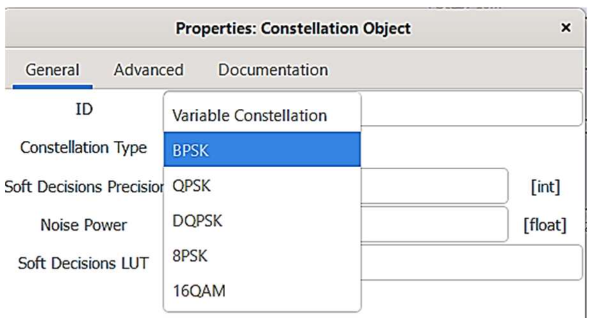


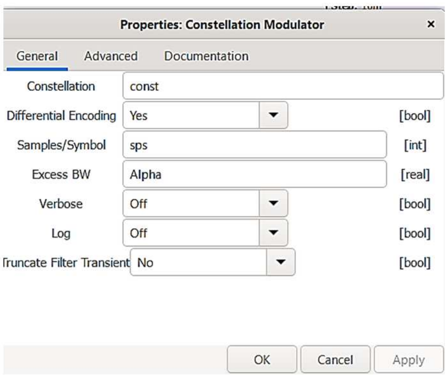
Contents of the Delay and rrc-taps variables

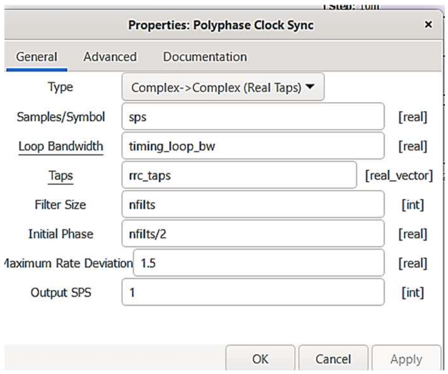
- Delay: int (5.5 * sps + 7)
- rrc_ taps : firdes.root _raised_ cosine (nfilts , nfilts , 1.0/float(sps), Alpha, 11* sps * nfilts)
- timing_loop_bw : 0.0628

Block Settings

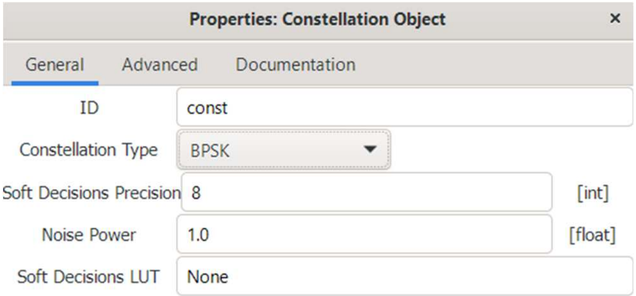


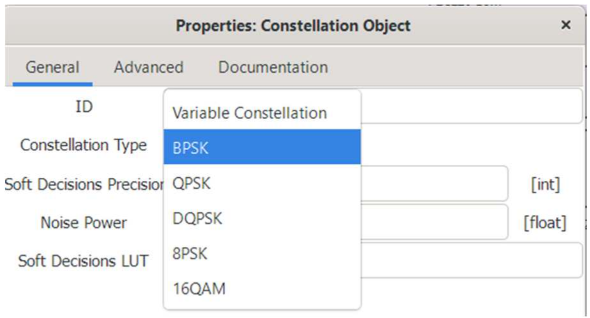






-
- Consider the Constellation Object block and the 5 possible modulations:

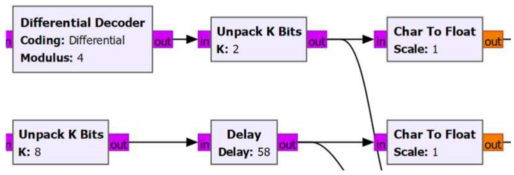




- Consider the other new blocks by determining their roles: Constellation modulator , Polyphase Clock Sync, Constellation Decoder , Differential Decoder .
- Vary the noise and observe the impact on the constellations and the BER

2. Manipulation 2: M-PSK modulation→ <u>Flowgraph to be made</u>

A. Modify the previous flowgraph to obtain a QPSK.
- Edit Constellation object →QPSK.
- Modulus Edit →4.
- Add a K-bit Unpack block (K: 2) to be placed as shown opposite
- Multiply the delay by 2 →(int (5.5 * sps + 7)*2
- Observe the BER as the noise power increases .



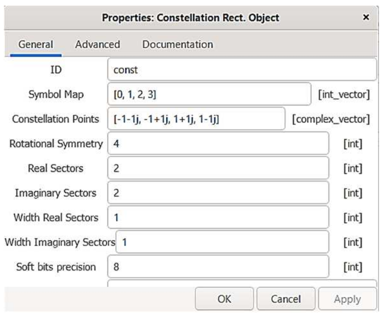
B. Modify the previous flowgraph to obtain an 8-PSK
- Edit Constellation object →8PSK
- Modulus Edit →8
- Add a K-bit Unpack block (K: 3).
- Observe the BER as the noise power increases.
- Multiply the delay by 2 →(int (5.5 * sps + 7)*3

3. Manipulation 3: M-QAM modulation→ <u>Flowgraph to be made</u>

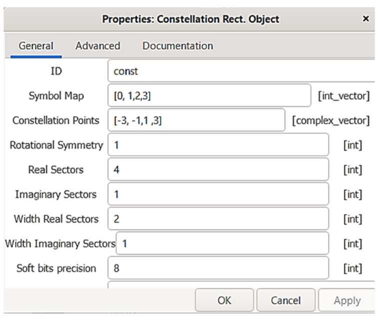
Modify the previous flowgraph to obtain a 16-QAM
- Edit Constellation object →16QAM
- Modulus Edit →16
- Add a K-bit Unpack block (K: 4).
- Observe the BER as the noise power increases.
- Multiply the delay by 2 →(int (5.5 * sps + 7)*4

4. Manipulation 4: M-ASK modulation→ <u>Flowgraph to be made</u>

Replace the Constellation Object →Constellation Rect Object block



A. Modify the previous flowgraph to obtain a 4-PSK.
- Modify Constellation Rect Object as shown opposite→
- Modulus Edit →4.
- Add a K-bit Unpack block (K: 2) to be placed as shown opposite.
- Multiply the delay by 2 →(int (5.5 * sps + 7)*2
- Observe the BER as the noise power increases.



B. Modify the previous flowgraph to obtain a 4-ASK
- Edit Constellation Rect Object as shown opposite→
- Constellation points →[-3, -1, 1, 3].
- Rotational Symmetry →1
- Real Sectors →4
- Imaginary Sectors →1
- Width Real Sectors →2
- Observe the BER as the noise power increases.

C. Modify the previous flowgraph to obtain an 8-ASK

- Observe the BER as the noise power increases

5. **Manipulation 5: M-FSK modulation**→ Flowgraph to be made