ECEN 4652/5002 Communications Lab Spring 2017
2-13-17 P. Mathys

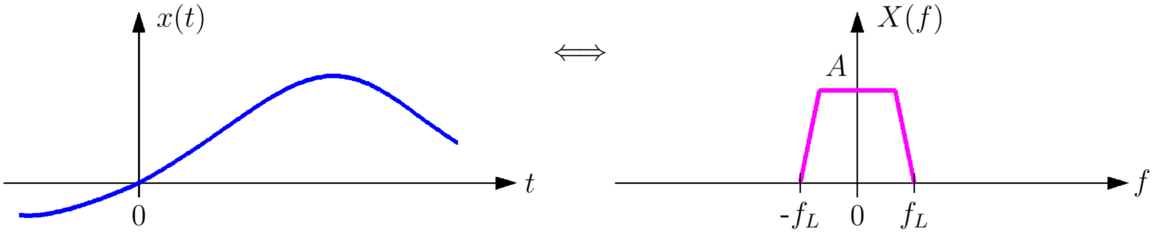
Lab 4: Sampling, Nyquist, Eye Diagrams, PR Signaling

1 Introduction

Pulse amplitude modulation (PAM) and the sampling theorem are closely related. For the sampling theorem the goal is to convert a bandlimited CT signal $x(t)$ to a DT sequence x_n by sampling at rate F_s such that $x(t)$ can be reconstructed exactly from x_n. PAM is concerned with the dual operation, namely converting from a DT sequence a_n with baud rate F_B to a (bandlimited) CT waveform $s(t)$ such that a close replica of a_n can be recovered again by sampling $s(t)$ at rate F_B. A special form of PAM is "digital PAM" for which a_n can only take on discrete values, e.g., $a_n \in \{-1, +1\}$ or $a_n \in \{-3, -1, +1, +3\}$. In this case a_n may even be recovered from a received digital PAM signal $r(t)$ in the presence of noise, intersymbol interference (ISI) and timing jitter, as long as the impairments are not too severe. A nice tool for judging how badly a digital PAM signal is corrupted is the so called eye diagram, which displays the superposition of several pieces of a PAM signal spaced apart in time by integer multiples of $T_B = 1/F_B$. To avoid impairments by ISI, it is possible to design bandlimited pulses that satisfy Nyquist's first criterion. The most prominent and most widely used such pulse is the "raised cosine in frequency (RCf)" pulse with rolloff parameter α. A different philosophy is to either introduce or allow a controlled amount of ISI, thereby shaping the spectrum of the PAM signal in such a way as to make the most of a practical (non-ideal) communication channel. This technique is called partial response (PR) signaling.

1.1 The Sampling Theorem

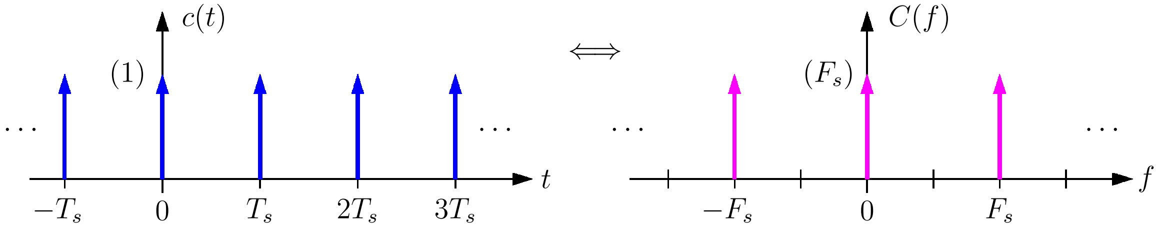
Let $x(t) \Leftrightarrow X(f)$ be a CT signal which is bandlimited to f_L. Graphically, this can be expressed as follows.



To describe the effect of ideal sampling with rate $F_s = 1/T_s$ at times nT_s, the "picket fence" FT pair

$$c(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \qquad \Longleftrightarrow \qquad C(f) = F_s \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \, ,$$

which is shown in the figure below, can be used.



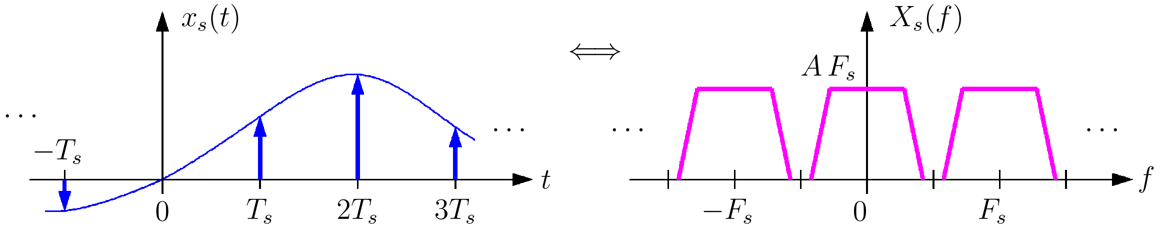
Multiplying $x(t)$ in the time domain by $c(t)$ yields the (impulsive) sampled waveform

$$x_s(t) = x(t)\,c(t) = \sum_{n=-\infty}^{\infty} x(t)\,\delta(t-nT_s) = \sum_{n=-\infty}^{\infty} \underbrace{x(nT_s)}_{=\,x_n}\,\delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x_n\,\delta(t-nT_s)\,,$$

where the third equality follows from the “sifting” property of the unit impulse $\delta(t)$ (i.e., $x(t)\,\delta(t-t_0)$ behaves like $x(t_0)\,\delta(t-t_0)$). In the frequency domain $X_s(f) \Leftrightarrow x_s(t)$ is obtained by convolving $X(f)$ and $C(f)$

$$X_s(f) = X(f) * C(f) = F_s \sum_{k=-\infty}^{\infty} X(f) * \delta(f-kF_s) = F_s \sum_{k=-\infty}^{\infty} X(f-kF_s) = X(f/F_s)\,.$$

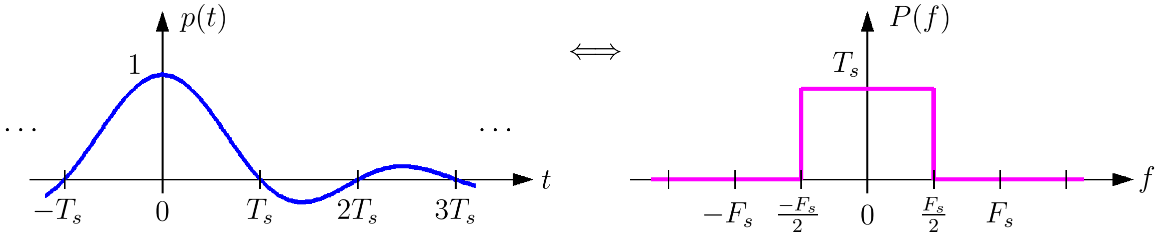
Note that $X(f/F_s)$ is a DTFT in the normalized frequency $\phi = f/F_s$. The last equality follows from the fact that $X_s(f)$ is periodic in f with period F_s (and thus periodic in f/F_s with period 1, a necessary requirement for a DTFT). The pair $x_s(t) \Leftrightarrow X_s(f) = X(f/F_s)$ is shown in the next figure.



The impulses at $t = nT_s$ have areas x_n and instead of the impulsive CT waveform $x_s(t)$ one could also draw a stem plot of the DT sequence x_n versus n. In the frequency domain it is easy to see that if the bandwidth f_L of $X(f)$ exceeds $1/(2T_s) = F_s/2$ then the shifted versions $X(f-kF_s)$ of $X(f)$ overlap for different k, a well-known phenomenon called **aliasing**. If there is no aliasing, i.e., if $|X(f)| = 0$ for $|f| \geq F_s/2$, then $x(t)$ can be recovered uniquely from $x_s(t)$ (or, equivalently, from the sequence x_n) by using an ideal LPF with cutoff frequency $F_s/2$. The unit impulse response of this LPF

$$h(t) = p(t) = \frac{\sin(\pi F_s t)}{\pi F_s t}\,,$$

and its FT $P(f)$ are shown next.



Clearly, the multiplication $X_s(f)\,P(f)$ in the frequency domain yields $X(f)$ if there is no aliasing. In the time domain this corresponds to the convolution

$$x_s(t) * p(t) = \Big(\sum_{n=-\infty}^{\infty} x_n\, \delta(t - nT_s) \Big) * p(t) = \sum_{n=-\infty}^{\infty} x_n\, p(t - nT_s) \;,$$

which is equal to $x(t)$ if there is no aliasing. This leads to the following theorem.

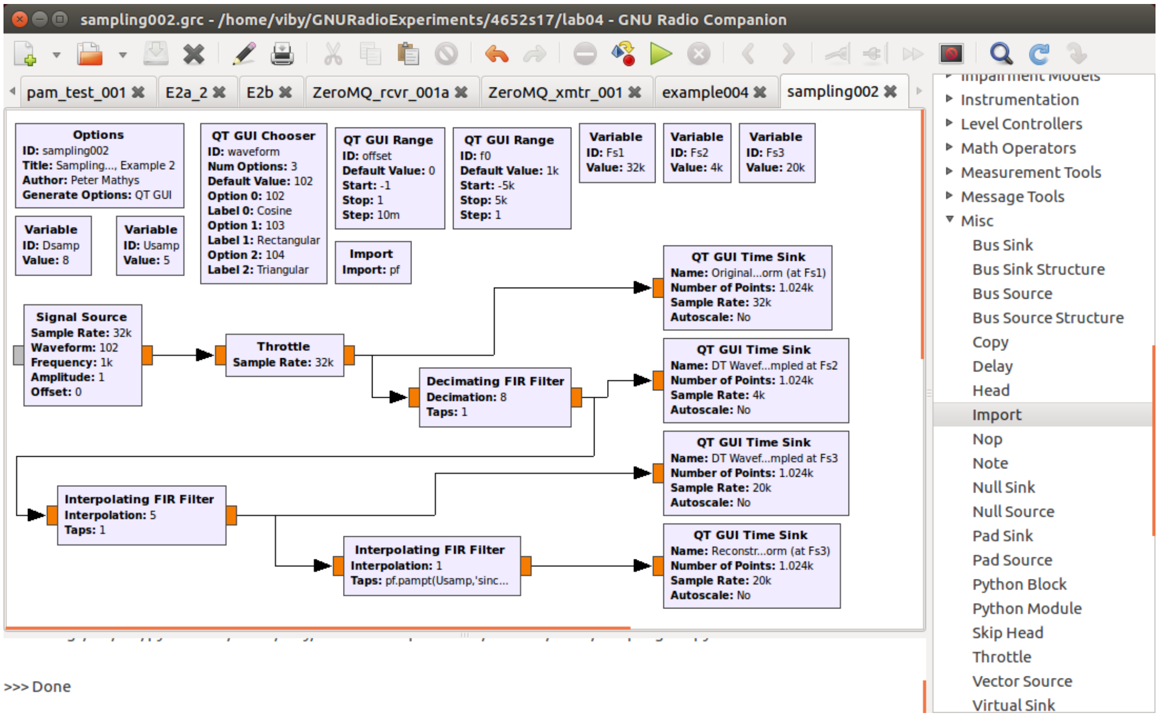
Theorem: Sampling Theorem. If $x(t)$ is bandlimited to the **Nyquist frequency** $F_s/2$, then it can be recovered uniquely from its samples $x_n = x(nT_s)$, sampled at rate $F_s = 1/T_s$, by using

$$x(t) = \sum_{n=-\infty}^{\infty} x_n\, \frac{\sin\big(\pi F_s (t - nT_s)\big)}{\pi F_s (t - nT_s)} = \sum_{n=-\infty}^{\infty} x_n\, \frac{\sin\big(\pi (F_s t - n)\big)}{\pi (F_s t - n)} \;,$$

to interpolate between the DT samples x_n.

1.2 The Sampling Theorem in the GRC

The different stages of the sampling theorem can be easily implemented in the GNU Radio companion (GRC) and nicely visualized using QT GUI Sinks. The basic flowgraph looks as follows.



Options
ID: sampling002
Title: Sampling..., Example 2
Author: Peter Mathys
Generate Options: QT GUI

Variable
ID: Dsamp
Value: 8

Variable
ID: Usamp
Value: 5

QT GUI Chooser
ID: waveform
Num Options: 3
Default Value: 102
Option 0: 102
Label 0: Cosine
Option 1: 103
Label 1: Rectangular
Option 2: 104
Label 2: Triangular

QT GUI Range
ID: offset
Default Value: 0
Start: -1
Stop: 1
Step: 10m

Import
Import: pf

QT GUI Range
ID: f0
Default Value: 1k
Start: -5k
Stop: 5k
Step: 1

Variable
ID: Fs1
Value: 32k

Variable
ID: Fs2
Value: 4k

Variable
ID: Fs3
Value: 20k

Signal Source
Sample Rate: 32k
Waveform: 102
Frequency: 1k
Amplitude: 1
Offset: 0

Throttle
Sample Rate: 32k

Decimating FIR Filter
Decimation: 8
Taps: 1

Interpolating FIR Filter
Interpolation: 5
Taps: 1

Interpolating FIR Filter
Interpolation: 1
Taps: pf.pampt(Usamp,'sinc...

QT GUI Time Sink
Name: Original...orm (at Fs1)
Number of Points: 1.024k
Sample Rate: 32k
Autoscale: No

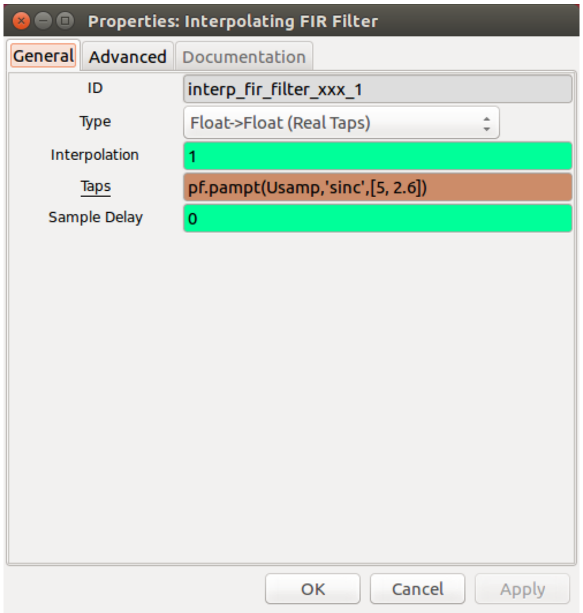
QT GUI Time Sink
Name: DT Wavef...mpled at Fs2
Number of Points: 1.024k
Sample Rate: 4k
Autoscale: No

QT GUI Time Sink
Name: DT Wavef...mpled at Fs3
Number of Points: 1.024k
Sample Rate: 20k
Autoscale: No

QT GUI Time Sink
Name: Reconstr...orm (at Fs3)
Number of Points: 1.024k
Sample Rate: 20k
Autoscale: No

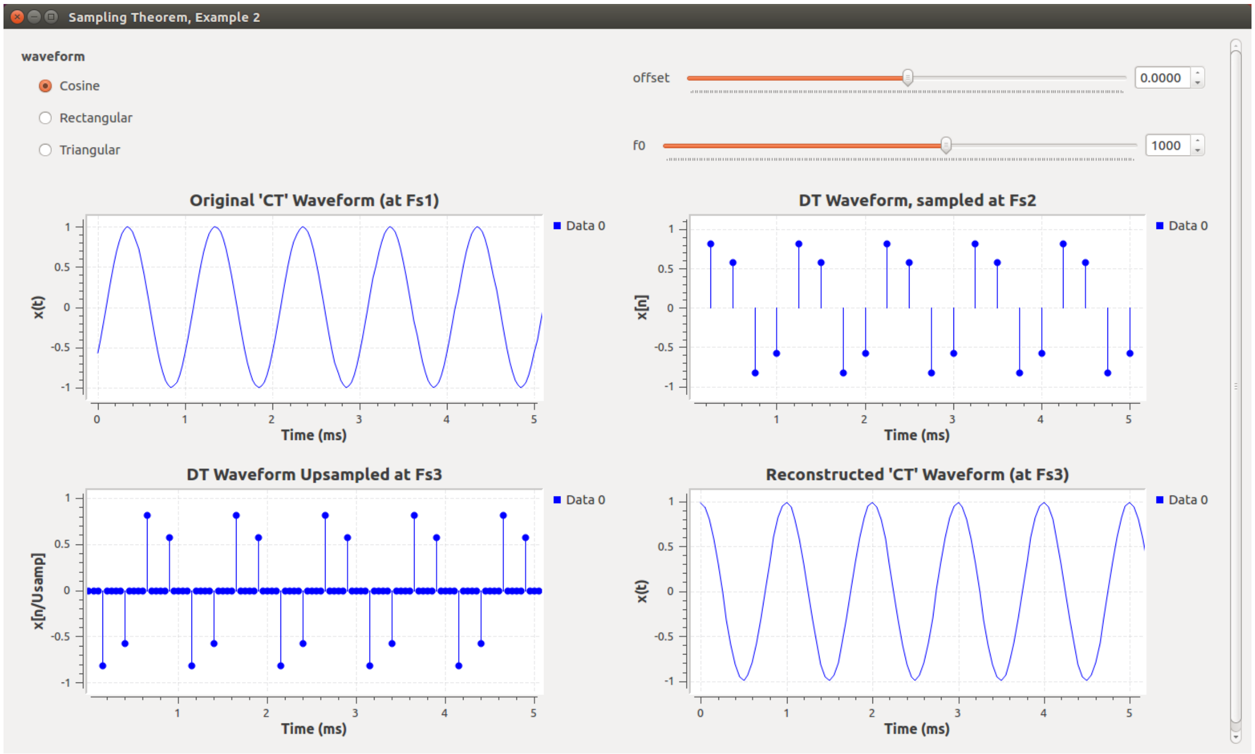
```
>>> Done
```

The “Signal Source” on the left is used to generate “Cosine”, “Square” or “Triangle” waveforms at the ‘CT’ sampling rate of `Fs1=32` kHz. This ‘original CT waveform’ is then sampled at rate `Fs2=4` kHz using a “Decimating FIR Filter” with decimation factor `Dsamp=8`. The result is a DT sequence that is subsequently upsampled by a factor of `Usamp=5` to produce a ‘reconstructed CT waveform’ at rate `Fs3=20` kHz. The upsampling process takes place in two steps. First, an “Interpolating FIR Filter” is used to insert `Usamp-1=4` zeros between the samples of the DT sequence at `Fs2`. The resulting sequence, that now has sampling rate `Fs3=Usamp*Fs2=20` kHz, is then passed through an “Interpolating FIR Filter” with interpolation factor 1 that acts as a lowpass filter at `Fs2/2=2` kHz. The filter taps for this filter are generated by the function `pampt` from Python module `ptfun.py` which is imported as `pf`. Here is a detailed look at the Properties of this “Interpolating FIR Filter”.



ID: interp_fir_filter_xxx_1
Type: Float->Float (Real Taps)
Interpolation: 1
Taps: pf.pampt(Usamp,'sinc',[5, 2.6])
Sample Delay: 0

Note that the specified interpolation waveform is the ‘sinc’ function, in accordance with the statement of the sampling theorem. Running the flowgraph and zooming in on the Time Sinks produces the following graphs.


Sampling Theorem, Example 2
Original 'CT' Waveform (at Fs1)
x(t)
Time (ms)
Data 0
DT Waveform, sampled at Fs2
x[n]
Time (ms)
Data 0
DT Waveform Upsampled at Fs3
x[n/Usamp]
Time (ms)
Data 0
Reconstructed 'CT' Waveform (at Fs3)
x(t)
Time (ms)
Data 0
1
0.5
0
-0.5
-1
0
1
2
3
4
5

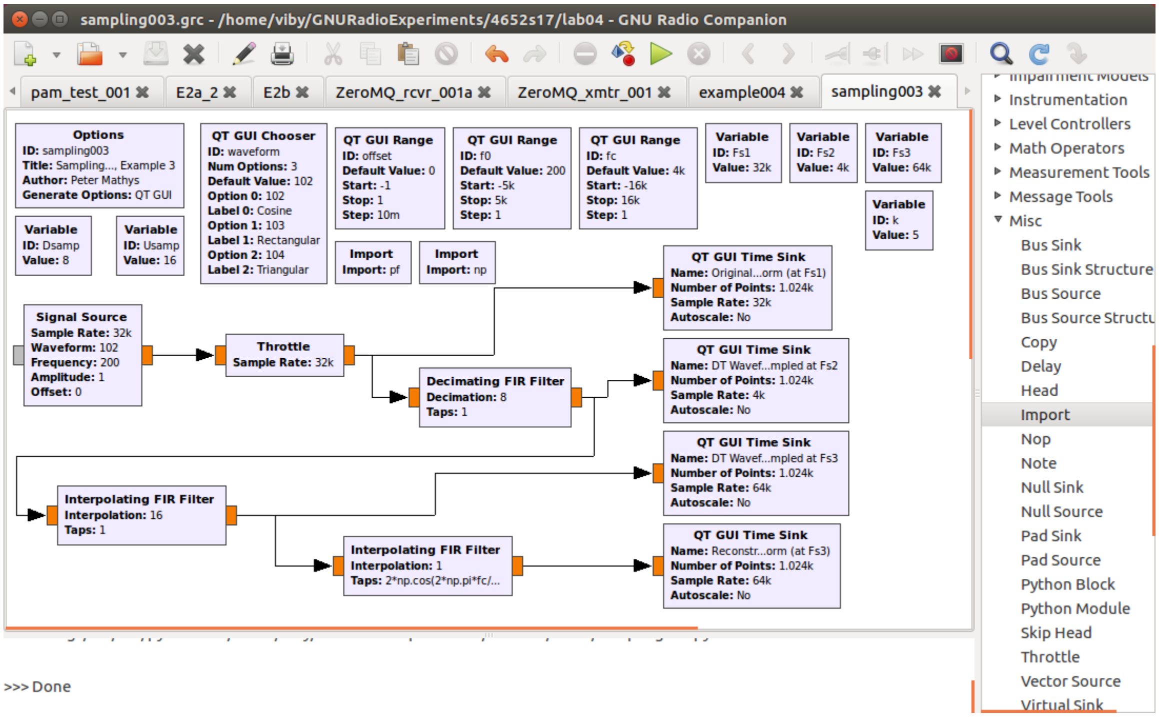
waveform
● Cosine
○ Rectangular
○ Triangular

offset: 0.0000
f0: 1000

The first graph on the top left shows the original ‘CT’ waveform (sampled at rate `Fs1=32` kHz). The graph on the top right shows the DT sequence after sampling at rate `Fs2=4` kHz.

The graph at the bottom left shows the DT sequence at rate `Fs3=20` kHz after inserting `Usamp-1` zeros between the samples of the DT sequence at `Fs2=4` kHz. The graph at the bottom right finall shows the reconstructed 'CT' waveform (sampled at rate `Fs3=20` kHz. Note that this flowgraph and the corresponding graphs not only illustrates the sampling theorem, but it also shows how to convert from one sampling rate (`Fs1`) to another (`Fs3`), even if the two rates are not integer multiples of each other. In general, rational resampling (when `Fs3/Fs1` is a rational number) requires (possibly repeated) up and down sampling stages.

So far we have assumed that the CT waveform $x(t)$ under consideration is a lowpass waveform, bandlimited to some highest frequency $f_L < F_s/2$. This dictated the use of a lowpass filter for the reconstruction of the original CT waveform. But since the spectrum of the sampled sequence x_n is periodic with period F_s, we could also use a bandpass filter for interpolation and filter out the frequency band from $F_s/2$ to $3F_s/2$ for a unique CT waveform reconstruction. Then we could argue that if this reconstruction had been the original waveform, we have a new sampling theorem for bandlimited bandpass signals. To see what the reconstructed waveform looks like, let's build the following flowgraph in the GRC.


sampling003.grc - /home/viby/GNURadioExperiments/4652s17/lab04 - GNU Radio Companion

pam_test_001 ✖ | E2a_2 ✖ | E2b ✖ | ZeroMQ_rcvr_001a ✖ | ZeroMQ_xmtr_001 ✖ | example004 ✖ | sampling003 ✖

Options
ID: sampling003
Title: Sampling..., Example 3
Author: Peter Mathys
Generate Options: QT GUI

Variable
ID: Dsamp
Value: 8

Variable
ID: Usamp
Value: 16

QT GUI Chooser
ID: waveform
Num Options: 3
Default Value: 102
Option 0: 102
Label 0: Cosine
Option 1: 103
Label 1: Rectangular
Option 2: 104
Label 2: Triangular

QT GUI Range
ID: offset
Default Value: 0
Start: -1
Stop: 1
Step: 10m

QT GUI Range
ID: f0
Default Value: 200
Start: -5k
Stop: 5k
Step: 1

QT GUI Range
ID: fc
Default Value: 4k
Start: -16k
Stop: 16k
Step: 1

Variable
ID: Fs1
Value: 32k

Variable
ID: Fs2
Value: 4k

Variable
ID: Fs3
Value: 64k

Variable
ID: k
Value: 5

Import
Import: pf

Import
Import: np

Signal Source
Sample Rate: 32k
Waveform: 102
Frequency: 200
Amplitude: 1
Offset: 0

Throttle
Sample Rate: 32k

Decimating FIR Filter
Decimation: 8
Taps: 1

Interpolating FIR Filter
Interpolation: 16
Taps: 1

Interpolating FIR Filter
Interpolation: 1
Taps: 2*np.cos(2*np.pi*fc/...

QT GUI Time Sink
Name: Original...orm (at Fs1)
Number of Points: 1.024k
Sample Rate: 32k
Autoscale: No

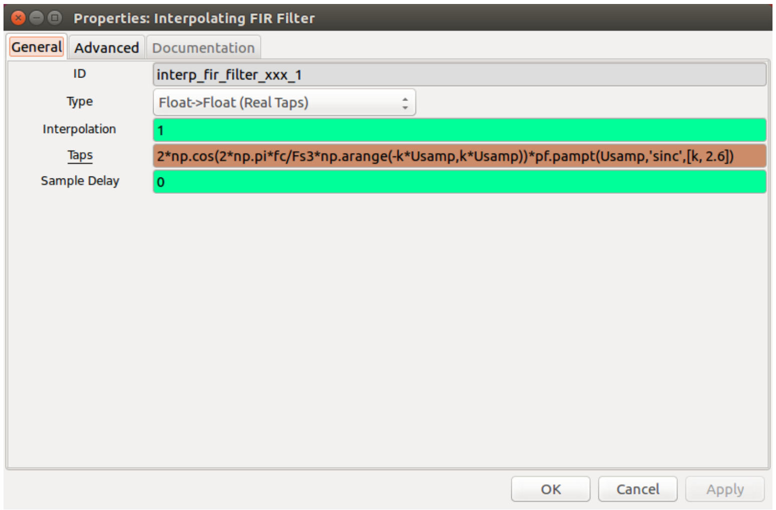
QT GUI Time Sink
Name: DT Wavef...mpled at Fs2
Number of Points: 1.024k
Sample Rate: 4k
Autoscale: No

QT GUI Time Sink
Name: DT Wavef...mpled at Fs3
Number of Points: 1.024k
Sample Rate: 64k
Autoscale: No

QT GUI Time Sink
Name: Reconstr...orm (at Fs3)
Number of Points: 1.024k
Sample Rate: 64k
Autoscale: No

```
>>> Done
```

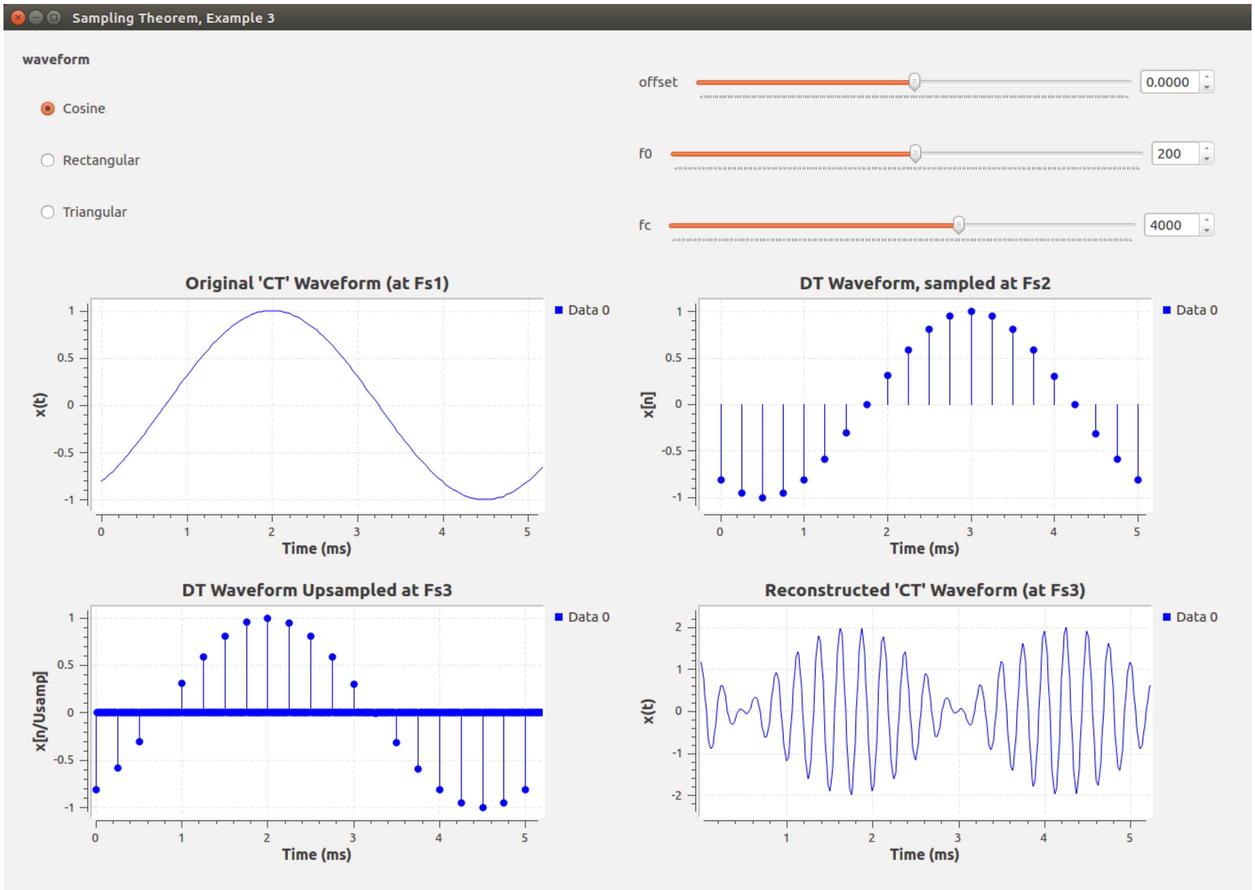
This is almost identical to the previous flowgraph to illustrate the sampling theorem in the GRC, except the the specification of the filter taps for the second "Interpolating FIR Filter" is now a bandpass filter with center frequency `fc`. The Properties of this block are shown below.



Properties: Interpolating FIR Filter

ID	interp_fir_filter_xxx_1
Type	Float->Float (Real Taps)
Interpolation	1
Taps	2*np.cos(2*np.pi*fc/Fs3*np.arange(-k*Usamp,k*Usamp))*pf.pampt(Usamp,'sinc',[k, 2.6])
Sample Delay	0

What makes this a bandpass filter (for real-valued signals) is the multiplication by `cos(2*np.pi*fc/Fs3*np` of the `pf.pampt(...)` filter taps. The graphs below show the resulting time signals for the case when `Fs1=32` kHz, `Fs2=4` kHz, `Fs3=64` kHz, `f0=200` Hz, and `fc=4` kHz.



Sampling Theorem, Example 3

waveform

● Cosine

○ Rectangular

○ Triangular

offset 0.0000

f0 200

fc 4000

Original 'CT' Waveform (at Fs1)
x(t)
Time (ms)
Data 0
DT Waveform, sampled at Fs2
x[n]
Time (ms)
Data 0
DT Waveform Upsampled at Fs3
x[n/Usamp]
Time (ms)
Data 0
Reconstructed 'CT' Waveform (at Fs3)
x(t)
Time (ms)
Data 0
1
0.5
0
-0.5
-1
2
-2
0
1
2
3
4
5

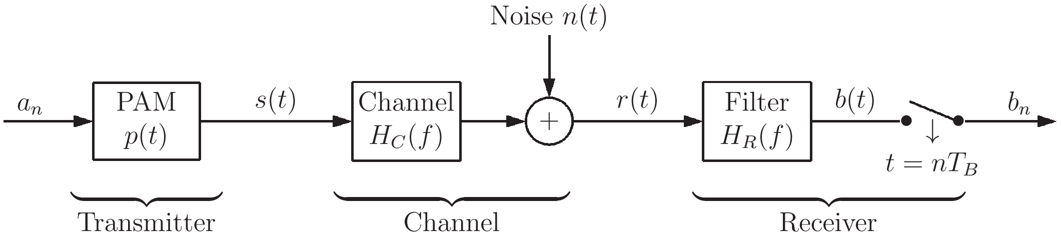
The reconstructed signal is

$$x_r(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) = 2\cos(\pi(f_2 - f_1)t)\cos(\pi(f_2 + f_1)t)\ ,$$

for $f_1 = 3800$ Hz and $f_2 = 4200$ Hz. This is an amplitude modulated (AM) signal with (suppressed) carrier frequency 4000 Hz and message frequency 200 Hz. Thus, the extension of the sampling theorem to bandpass signals is: If $x(t)$ is bandlimited to the range $(k-0.5)F_s \ldots (k+0.5)F_s$, k integer, then it can be recovered uniquely from its samples at rate F_s.

1.3 Nyquist's First Criterion

The blockdiagram of a general PAM communication system with baud rate $F_B = 1/T_B$ is shown in the following figure.



Without the noise $n(t)$, the Fourier transform of the PAM signal $b(t)$ at the receiver just before sampling is

$$B(f) = A(fT_B)\,Q(f) \qquad \Longleftrightarrow \qquad b(t) = \sum_{m=-\infty}^{\infty} a_m\, q(t - mT_B)$$

where

$$Q(f) = P(f)\,H_C(f)\,H_R(f) \qquad \Longleftrightarrow \qquad q(t) = p(t) * h_C(t) * h_R(t)$$

To keep the circuitry at the receiver simple, it is desirable to keep the effects of intersymbol interference (ISI) at the sampling times $t = nT_B$ small. The ideal situation is to have no ISI and thus $b_n = K\,a_n$ where K is some nonzero constant. In the time domain

$$b_n = b(nT_B) = \sum_{m=-\infty}^{\infty} a_m\, q\big((n-m)T_B\big) = \sum_{m=-\infty}^{\infty} a_m\, q_{n-m}\ ,$$

where $q_n = q(nT_B)$. Therefore, the condition in the time domain for no ISI is

$$q_n = q(nT_B) = \begin{cases} K\ , & n = 0\ , \\ 0\ , & \text{otherwise}\ . \end{cases}$$

This is known as **Nyquist's first criterion** in the time domain.

To derive a similar condition in the frequency domain, define the sampling function

$$c(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_B) \qquad \Longleftrightarrow \qquad C(f) = F_B \sum_{k=-\infty}^{\infty} \delta(f - kF_B)$$

Then $b_n = b(t)\,c(t)$ which translates into

$$\begin{aligned} B(fT_B) &= B(f) * C(f) = B(f) * \Big(F_B \sum_{k=-\infty}^{\infty} \delta(f - kF_B)\Big) = F_B \sum_{k=-\infty}^{\infty} B(f - kF_B) \\ &= F_B \sum_{k=-\infty}^{\infty} \underbrace{A(fT_B - kF_B T_B)}_{= A(fT_B)} Q(f - kF_B) = F_B\, A(fT_B) \sum_{k=-\infty}^{\infty} Q(f - kF_B)\,. \end{aligned}$$

The frequency domain condition for no ISI is thus

$$F_B \sum_{k=-\infty}^{\infty} Q(f - kF_B) = K\,,$$

where K is a nonzero constant. This statement is known as **Nyquist's first criterion** in the frequency domain.

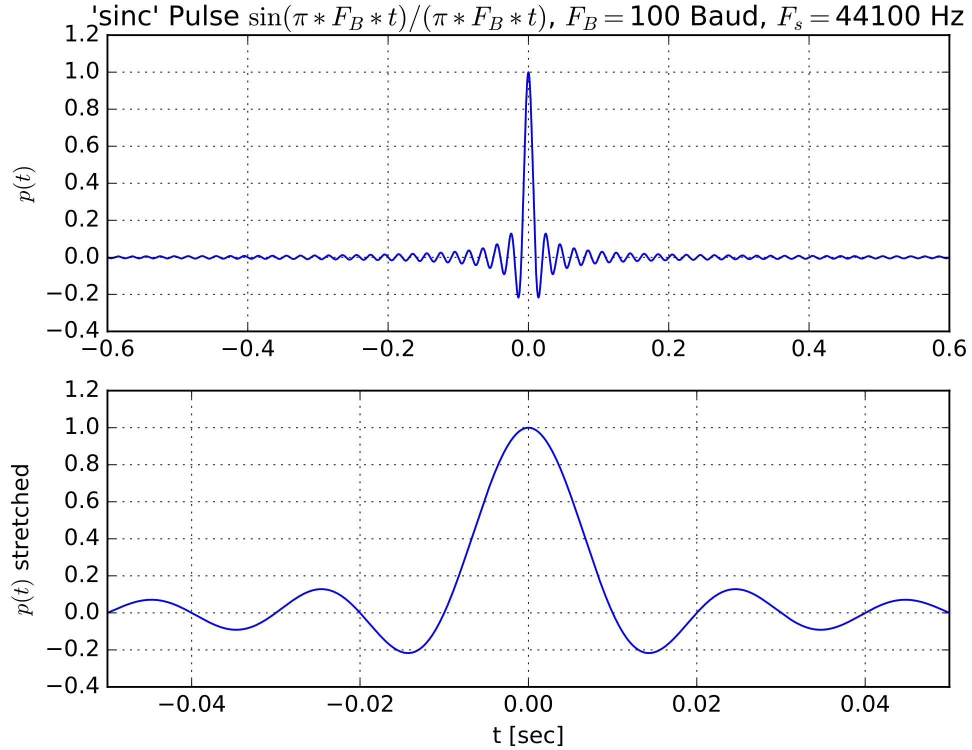
Using Nyquist's first criterion in the time and frequency domains it is readily seen that the simplest pulse in the time domain that avoids ISI is the rectangular pulse of width $T_B = 1/F_B$, and the pulse that uses the least amound of bandwidth and avoids ISI is the "sinc" pulse of bandwidth $F_B/2$.

1.4 Raised Cosine in Frequency Pulse

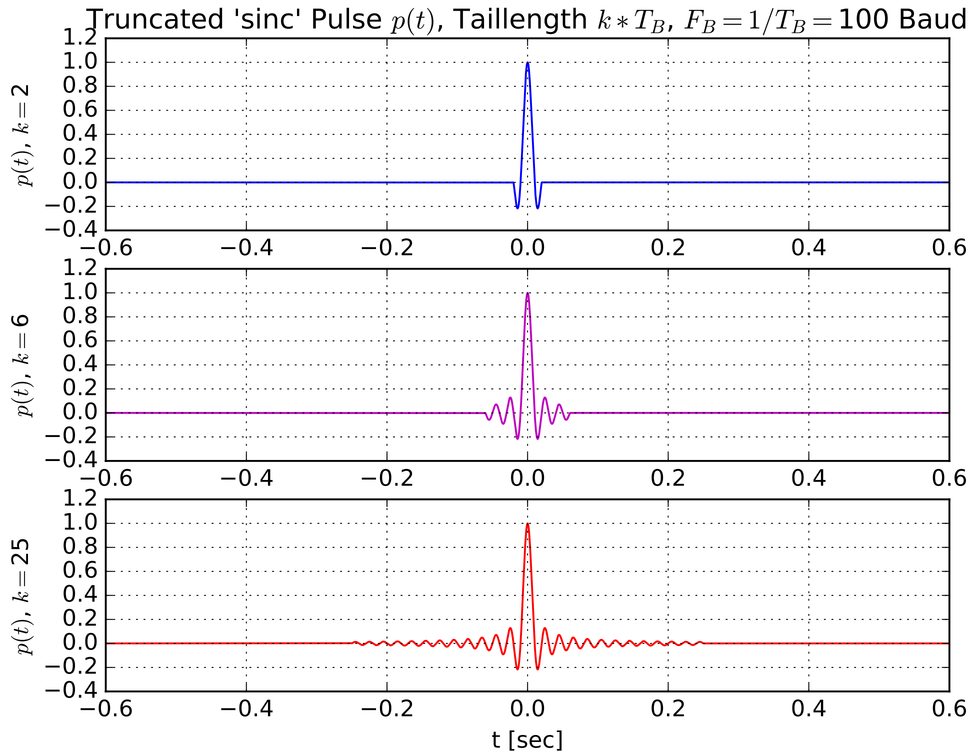
As mentioned previously, the "$\sin(x)/x$" or "sinc" pulse

$$p(t) = \frac{\sin(\pi t/T_B)}{\pi t/T_B} \qquad \Longleftrightarrow \qquad P(f) = \begin{cases} T_B\,, & -F_B/2 \le f < F_B/2\,, \\ 0\,, & \text{otherwise}\,, \end{cases}$$

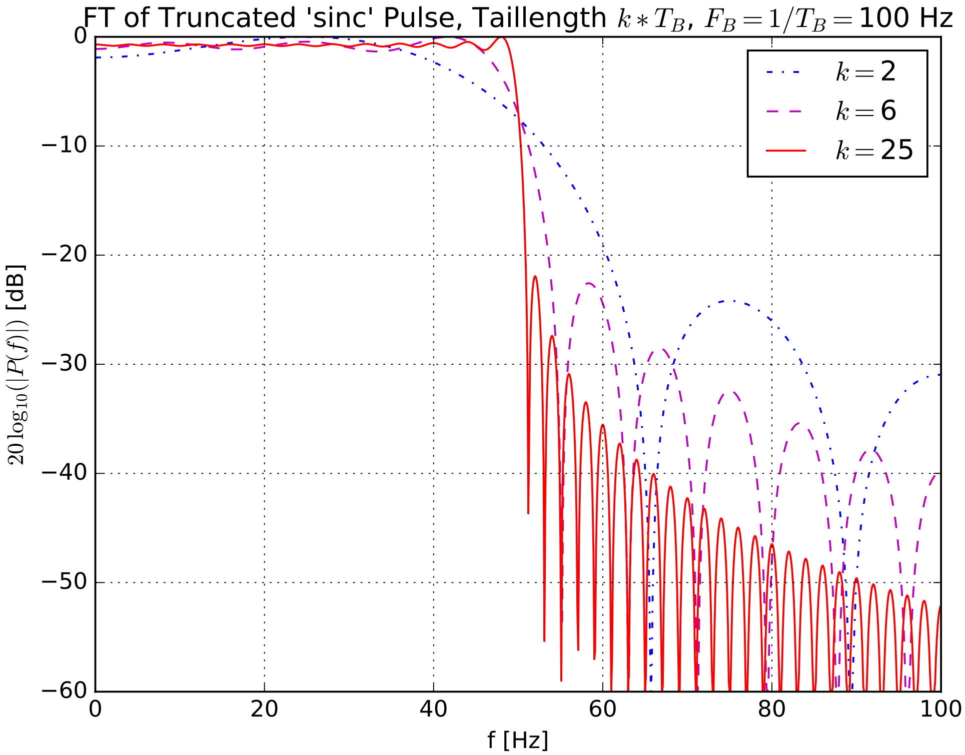
which is shown in the graph below, is the pulse of smallest bandwidth that satisfies Nyquist's first criterion for no ISI for a PAM system with baud rate $F_B = 1/T_B$.



The main shortcoming of this pulse, however, is that (as a result of the "brickwall" nature of $P(f)$) its "tails" in the time domain decrease only like $1/|t|$ in amplitude as $|t| \to \infty$. This leads to problems when the "tails" of the "sinc" pulse are truncated as shown next, e.g., to length kT_B for some integer k, for practical implementations.

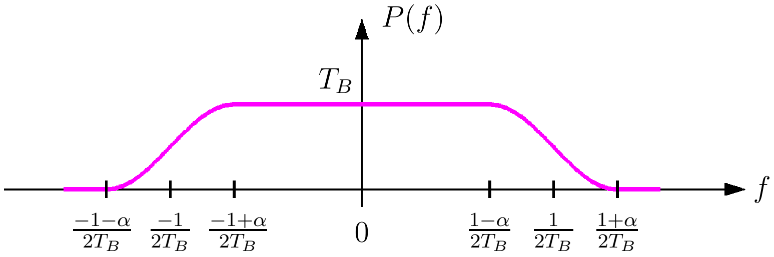


The following figure shows the resulting FT of a "sinc" pulse for $F_B = 100$ Hz and $k = 2, 6, 25$.



The truncation in the time domain to $t = \pm kT_B$ leads to sidelobes in the frequency domain which are spaced $F_B/(2k)$ apart. This is closely related to Gibbs phenomenon (trying to approximate a discontinuity in one domain with a finite number of basis functions in the other domain). As a consequence, the first sidelobe will always only be about 21 dB down from the main lobe, no matter how large k is chosen. Another problem is that, since $\int_\epsilon^\infty t^{-1}dt = \infty$, PAM signals based on the "sinc" pulse are very sensitive to timing errors of the sampling circuit at the receiver.

A solution to these problems is obtained by making the transition from T_B to 0 near $|f| = F_B/2 = 1/(2T_B)$ in the frequency domain less abrupt. The graph below shows the FT of a pulse $p(t)$ which uses an extra amount of $\alpha/(2T_B) = \alpha F_B/2$, with $0 \leq \alpha \leq 1$, of bandwidth to accomodate a sinusoidal or raised cosine rolloff.



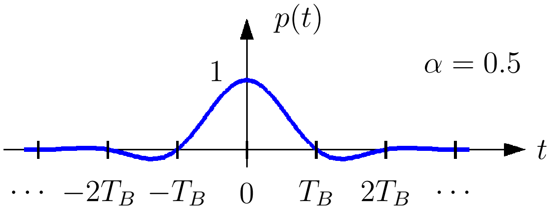
Analytically, this $P(f)$ can be expressed as

$$P(f) = \begin{cases} T_B\,, & |f| \le \frac{1-\alpha}{2T_B}\,, \\ \dfrac{T_B}{2}\left[1+\cos\left(\frac{\pi T_B}{\alpha}\left(|f|-\frac{1-\alpha}{2T_B}\right)\right)\right], & \frac{1-\alpha}{2T_B} \le |f| \le \frac{1+\alpha}{2T_B}\,, \\ 0\,, & |f| \ge \frac{1+\alpha}{2T_B}\,. \end{cases}$$

In the time domain this corresponds to

$$p(t) = \frac{\sin(\pi t/T_B)}{\pi t/T_B}\,\frac{\cos(\pi\alpha t/T_B)}{1-(2\alpha t/T_B)^2}$$

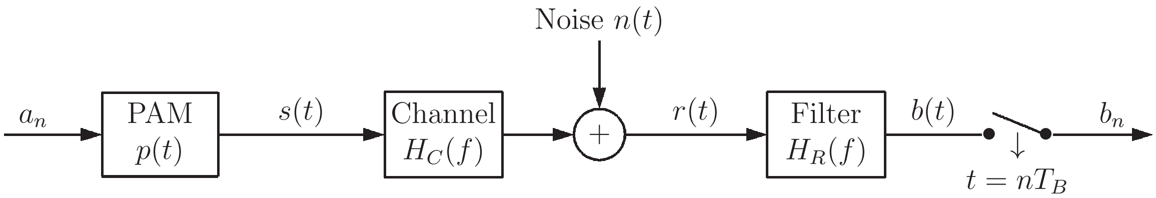
$$0 \le \alpha \le 1$$



Now, for any $\alpha > 0$, $p(t)$ decreases like $1/|t|^3$ as $|t| \to \infty$, which is a big improvement over the $1/|t|$ decrease of the"sinc" pulse. Typical values used in practice are $\alpha = 0.2\ldots0.5$. Because of its shape in the frequency domain, this pulse is called **raised cosine in frequency (RCf)** pulse. It is quite easy to see, either in the time or frequency domain, that the RCf pulse satisfies Nyquist's first criterion for no ISI.

1.5 Partial Response Signaling

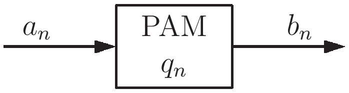
Consider the general blockdiagram of a communication system using pulse amplitude modulation (PAM) with baud rate $F_B = 1/T_B$.



Define

$$q(t) \iff Q(f) = P(f)\,H_C(f)\,H_R(f) \qquad \text{and} \qquad q_n = q(nT_B) \iff Q(z)\,.$$

The DT equivalent circuit of the PAM communication system is shown in the following figure.



If Nyquist's first criterion is satisfied then there is no intersymbol interference (ISI) at the sampling times and thus

$$q_n = \delta_n \quad \Longrightarrow \quad b_n = a_n * q_n = a_n * \delta_n = a_n \ .$$

Many practical channels have spectral nulls at certain frequencies, e.g., at dc if transmission media are transformer-coupled or magnetic recording and playback is used. Near the upper frequency limit, practical channels usually have some rolloff characteristic, as opposed to the "brickwall' cutoff of ideal filters. In magnetic recording the finite gap width of the read/write head actually leads to a spectral null which essentially defines the upper frequency limit of the channel. In all these cases it is not very convenient and efficient to try to satisfy Nyquist's first criterion. A better alternative is to allow a **controlled amount of ISI** between adjacent samples at the output of the DT equivalent PAM model $q_n \Leftrightarrow Q(z)$. This technique is called **partial response (PR) signaling** or correlative-level coding.

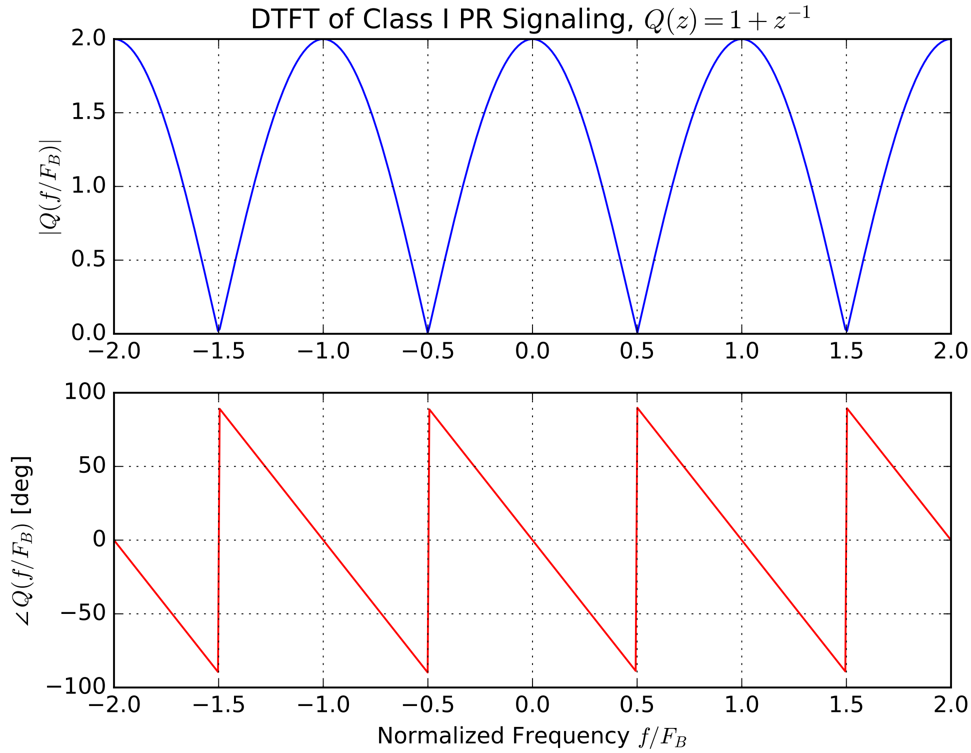
Example: A simple example is the DT equivalent PAM model with

$$Q(z) = 1 + z^{-1} \quad \Longleftrightarrow \quad q_n = \begin{cases} 1\,, & n = 0, 1\,, \\ 0\,, & \text{otherwise}\,, \end{cases}$$

which places a spectral null at $f = F_B/2$ and is called **class I PR**. The DTFT of this q_n is

$$\begin{aligned} Q(fT_B) &= \sum_{n=-\infty}^{\infty} q_n \, e^{-j2\pi f T_B n} = 1 + e^{-j2\pi f T_B} \\ &= (e^{j\pi f T_B} + e^{-j\pi f T_B}) \, e^{-j\pi f T_B} = 2 \cos \pi f T_B \, e^{-j\pi f T_B} \ . \end{aligned}$$

The magnitude and the phase of this DTFT are shown in the following figure.



To obtain a corresponding CT $q(t)$, make use of the fact that

$$Q(fT_B) = F_B \sum_{m=-\infty}^{\infty} Q(f - m\,F_B)\,.$$

To obtain a $q(t)$ of minimum bandwidth set

$$Q(f) = \begin{cases} T_B\,Q(fT_B)\,, & |f| < F_B/2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

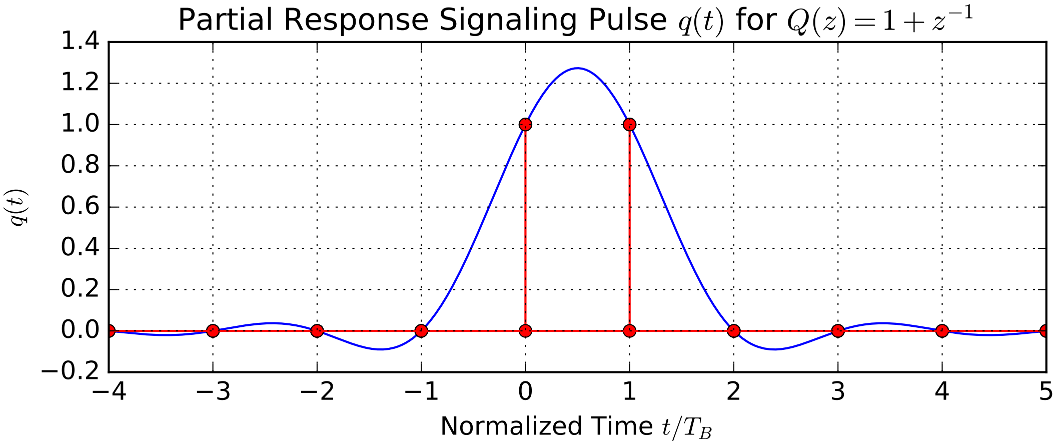
For class I PR this becomes

$$Q(f) = \begin{cases} 2T_B\,\cos\pi f T_B\,e^{-j\pi f T_B}\,, & |f| < F_B/2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

Taking the inverse FT yields

$$q(t) = \frac{\sin\pi t/T_B}{\pi t/T_B\,(1 - t/T_B)}\,, \qquad \text{with} \quad q(t) = 1 \ \text{ at } \ t = 0, T_B\,.$$

This pulse is shown in the following figure for the range$-4T_B \le t \le 5T_B$ (with $F_B = 100$ Hz and thus $T_B = 0.01$ s), and it is easy to see that $q_0 = q(0) = q_1 = q(T_B) = 1$, and all other $q_n = q(nT_B) = 0$.



More generally, if

$$Q(z) = \sum_n q_n\,z^{-n} \qquad \Longrightarrow \qquad Q(fT_B) = \sum_n q_n\,e^{-j2\pi f T_B n}\,,$$

then the $Q(f)$ of minimum bandwidth is

$$Q(f) = \begin{cases} T_B \sum_n q_n\,e^{-j2\pi f T_B n}\,, & |f| < F_B/2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

Taking the inverse FT yields $q(t)$ as

$$q(t) = \sum_n q_n \, \frac{\sin(\pi t/T_B - n\pi)}{\pi(t/T_B - n)} = \frac{\sin \pi t/T_B}{\pi t/T_B} \sum_n (-1)^n \, q_n \, \frac{t/T_B}{t/T_B - n} \, .$$

Different classes of PR signaling schemes and associated $Q(z)$ are shown in the following table. The most popular schemes are PR I and PR IV. Magnetic recording systems increasingly use the more advanced EPR4 and E2PR4 schemes.

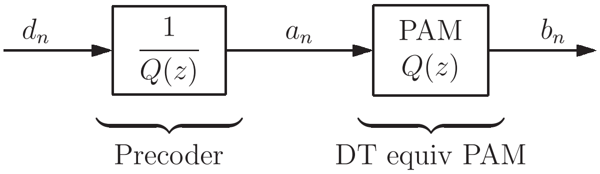
Classes of Partial Response Signaling Schemes

Class	$Q(z)$	Remarks
I	$1 + z^{-1}$	Duobinary coding
	$1 - z^{-1}$	"Dicode" channel
II	$(1 + z^{-1})^2$	
III	$(1 + z^{-1})(2 - z^{-1})$	
IV	$(1 + z^{-1})(1 - z^{-1})$	Modified duobinary
	$(1 + z^{-1})(1 + z^{-1} + z^{-2})$	(Optical storage)
EPR4	$(1 + z^{-1})^2(1 - z^{-1})$	Enhanced class IV
E2PR4	$(1 + z^{-1})^3(1 - z^{-1})$	Enhanced EPR4
V	$(1 + z^{-1})^2(1 - z^{-1})^2$	

In general, when PR signaling is used, $b_n \neq a_n$ in the expression $b_n = a_n * q_n$, even in the absence of noise because of the ISI that was deliberately introduced by using a $Q(z) \neq 1$. A simple solution to remove the effect of this ISI and recover the original data signal is to use a DT filter whose z-transform is the inverse of the DT PAM model $Q(z)$. For class I PR, for example, one would use

$$\text{PR I:} \quad Q(z) = 1 + z^{-1} \ \ \text{(FIR)} \quad \Longrightarrow \quad [Q(z)]^{-1} = \frac{1}{1 + z^{-1}} \ \ \text{(IIR)} \, .$$

At first sight it appears to be a good idea to use a DT filter with system function $[Q(z)]^{-1}$ at the receiver to remove the effects of ISI. However, in order to introduce only a well controlled amount of ISI, $Q(z)$ is generally chosen to be a FIR filter, which implies that its inverse is an IIR filter. But using an IIR filter at the receiver can lead to infinite error propagation if the received signal contains errors due to noise. Therefore, rather than compensating at the receiver for the ISI introduced by the PR signaling scheme, one generally uses **precoding** at the transmitter, as shown in the following block diagram.



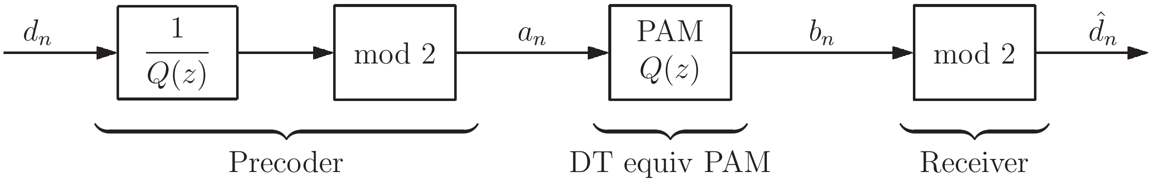
In the absence of noise, $b_n = d_n$ and the ISI introduced by the PR signaling scheme is compensated for at the transmitter. But suppose that

$$H(z) = [Q(z)]^{-1} = \frac{1}{1+z^{-1}} \quad \Longleftrightarrow \quad h_n = 1, -1, 1, -1, 1, -1, 1, -1, \dots .$$

Then, if $d_n = 1, 0, 1, 0, 1, 0, 1, 0, \dots$, the output of the precoder becomes

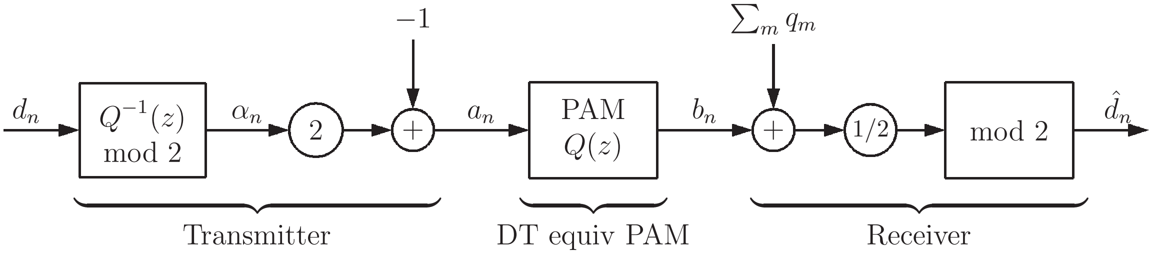
$$a_n = d_n * h_n = 1, -1, 2, -2, 3, -3, 4, -4, 5, -5, \dots ,$$

as can be easily verified by looking at $q_n * a_n = \{1, 1\} * a_n$. Thus, unless some additional measures are taken, the output of the precoder can grow without bound. Assuming that the input is binary and $d_n \in \{0, 1\}$, the output of the encoder can be taken **modulo 2** to prevent it from becoming unbounded. Then, in the absence of noise other than ISI, the original binary sequence can be recovered at the receiver by taking the channel output modulo 2. This is shown in the block diagram below.



Note that in the more realistic case when the channel attenuates the signal and there is additive white Gaussian noise in addition to ISI from PR signaling, the received sequence needs to be properly scaled and quantized to integers before the modulo 2 operation.

Another issue that needs to be addressed is that polar binary signaling ($a_n \in \{+1, -1\}$) is more efficient than unipolar binary signaling ($a_n \in \{0, 1\}$). The conversion from unipolar to polar is achieved using level translation and scaling at the transmitter and receiver as shown in the next blockdiagram.



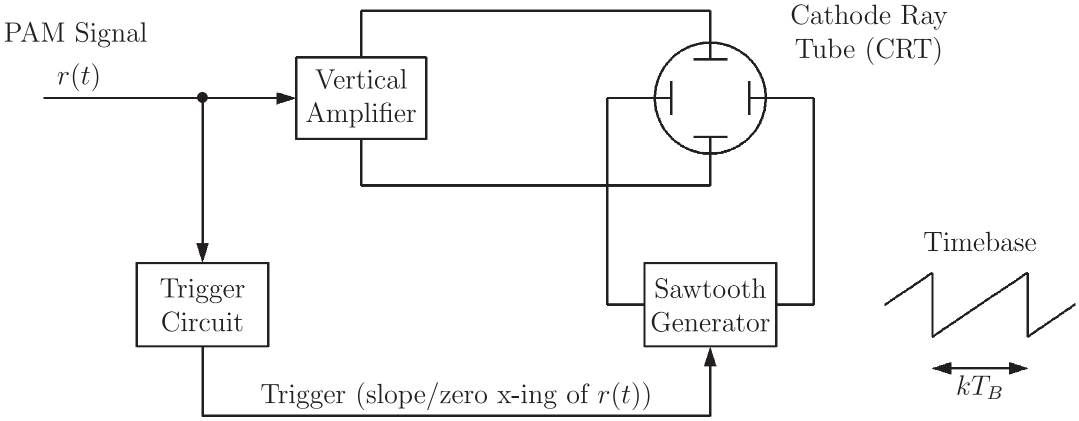
The precoded bits α_n are converted to $a_n = 2\alpha_n - 1$, i.e., $0 \to -1$ and $1 \to +1$, before transmission. If the channel is ideal (i.e., no attenuation and no noise except for ISI from PR signaling), the receiver computes $(b_n + \sum_m q_m)$ and takes the result modulo 2 to recover the transmitted data d_n. If there is channel attenuation, it needs to be compensated for before adding $\sum_m q_m$. If the channel adds Gaussian noise, then quantization to integers needs to be used before the modulo 2 operation.

Example: Binary Class I PR Signaling with Precoding. In this case $Q(z) = 1 + z^{-1}$. Assume that the initial condition of the precoder is zero and the initial condition of the channel is -1. Then, using the binary data sequence d_n given below and assuming no additive noise and no channel attenuation,

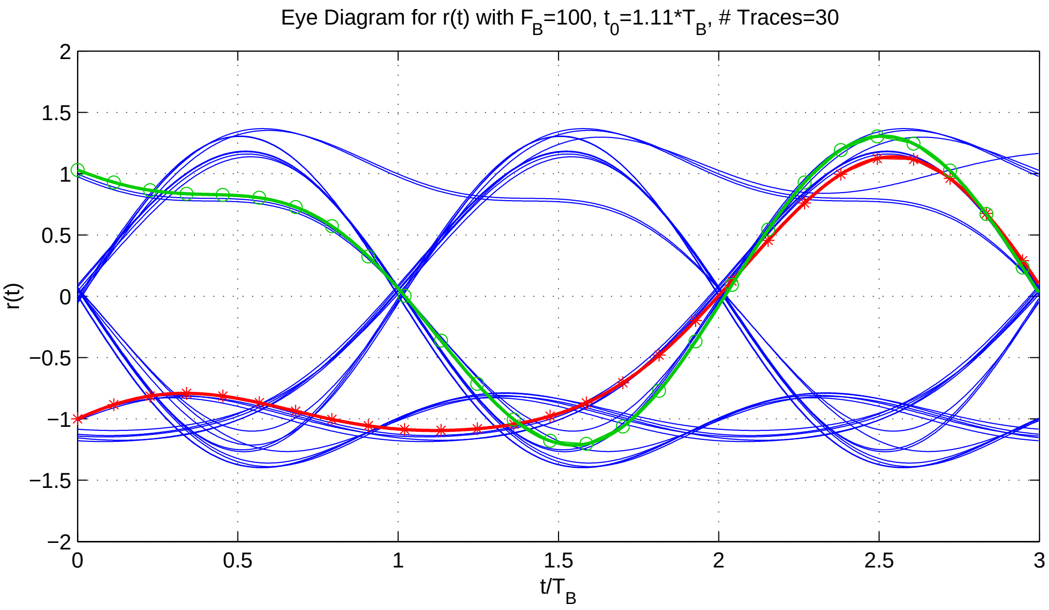
$$\begin{aligned}
d_n &= \{\ \ \ \texttt{0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1}\} \\
\alpha_n &= \{\ \texttt{0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0}\} \\
a_n &= \{\texttt{-1,-1,-1,-1,+1,+1,+1,-1,+1,+1,-1,-1,+1,-1,+1,-1}\} \\
b_n &= \{\ \ \ \texttt{-2,-2,-2, 0,+2,+2, 0, 0,+2, 0,-2, 0, 0, 0, 0}\} \\
\hat{d}_n &= \{\ \ \ \texttt{0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1}\}
\end{aligned}$$

1.6 Eye Diagrams

Let $r(t)$ be a received digital PAM signal with baud rate F_B that originated from a DT sequence a_n with L discrete amplitudes. A common method to analyze the quality of such a PAM signal is to use an oscilloscope, set the timebase to a multiple kT_B of the symbol interval $T_B = 1/F_B$, and trigger either on the slopes or the zero crossings of the incoming signal. This is shown schematically in the following blockdiagram.

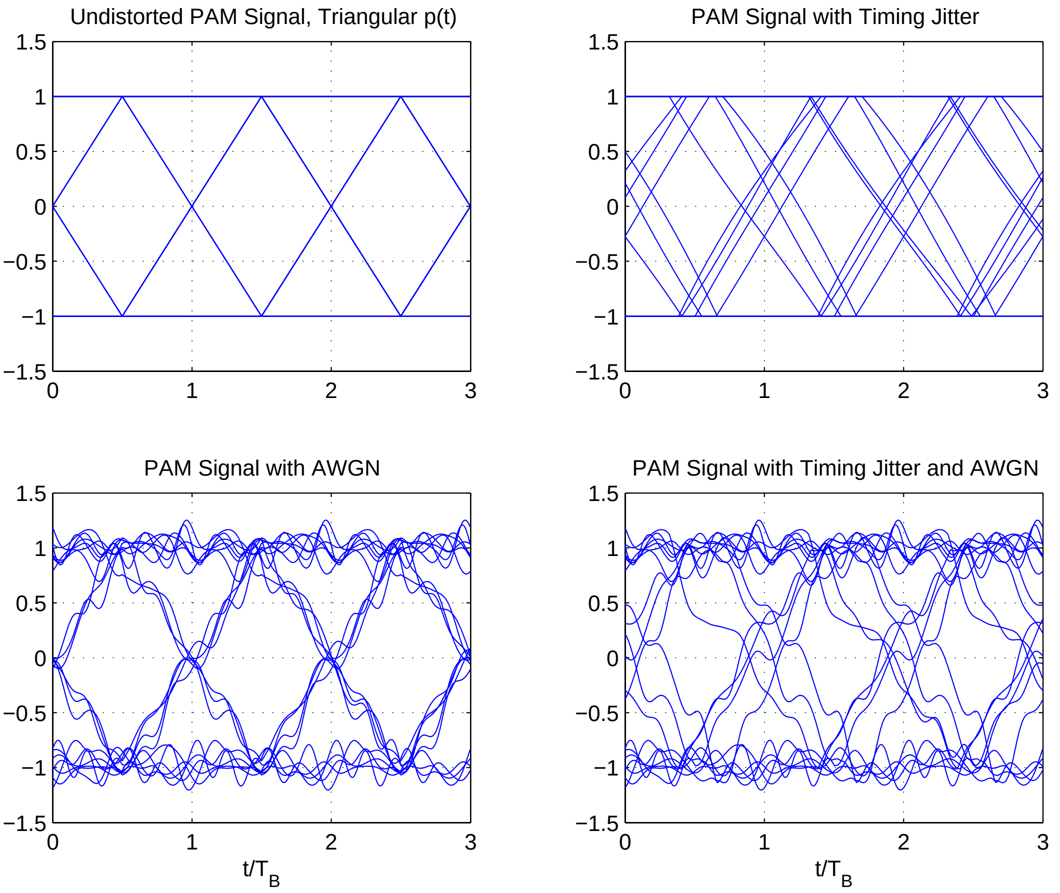


The result on the screen of the oscilloscope (the “CRT” or cathode ray tube) is the superposition of many signal segments of length kT_B. In $r(t)$ each segment starts at $t = nT_B + t_0$ for a different n, but on the oscilloscope all segments start at the left end of the display. Thus, if the PAM signal is sufficiently random and has L discrete amplitudes, then the oscilloscope screen will show all possible transitions between the L levels during one or more symbol intervals. An example that was obtained from a polar binary flat-top PAM signal with baud rate F_B Hz, after passing through a LPF channel with -3dB frequency near $0.7\,F_B$, is shown in the following graph.



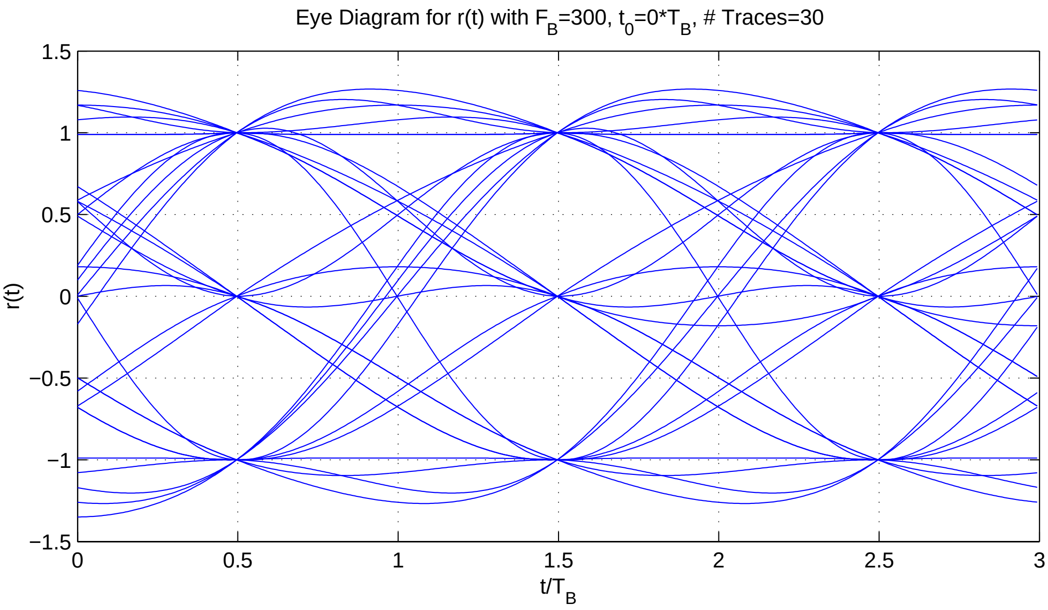
Such a graph is called an **eye diagram** or **eye pattern** because of its resemblance to the shape of an eye. The above plot extends over 3 symbol intervals and has been "synchronized" such that the optimum sampling times are approximately at $t/T_B = 0.5, 1.5, 2.5$. Looking at the graph, it is possible to trace all $2^3 = 8$ signal patterns consisting of ± 1 levels and extending over 3 symbol intervals. It can be seen, for instance, that the traces for $\{-1, -1, +1\}$ (red with `'*'` marks) and $\{+1, -1, +1\}$ (green with `'o'` marks) do not coincide for $1.5 \leq t/T_B \leq 2.5$ because of ISI.

The next figure shows eye patterns of width $3T_B$ for a polar binary PAM signal with triangular $p(t)$ and various transmission impairments.



The first graph shows the ideal, undistorted signal. The second graph shows a signal with timing jitter, but no amplitude impairments. The third graph shows a signal with additive white Gaussian noise (AWGN), but no timing jitter, whereas the last graph shows a signal that has been affected by both AWGN and timing jitter. In general, transmission impairments lead to a "closing" of the eye, either in horizontal, vertical, or both directions. Clearly, the more closed the eye is, the more difficult it becomes for the receiver to make the right decision, which leads to an increase in the error rate.

If the signal-to-noise ratio (SNR) on a given transmission channel is high enough, it is quite common to use PAM signals with more levels, e.g., in the range $L = 2 \ldots 128$. An example with $L = 3$ levels which uses an undistorted RCf pulse with $\alpha = 0.6$ is shown in the next graph.



In general, if there are three sampling time instants and L levels and the PAM signal is sufficiently random, then the eye pattern shows all possible L^3 transitions between the L levels.

2 Lab Experiments

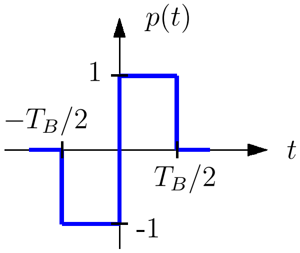
E1. Sampling Theorem, and PAM with Manchester and `'RCf'` $p(t)$. (a) Build the GRC flowgraph for the visualization of the sampling theorem given in the introduction. Instead of using a cosine waveform with frequency `f0=1000` Hz, use `f0=2700` Hz. Observe the reconstructed ‘CT’ waveform and explain why it does not have a frequency of 2700 Hz.

(b) Use the same flowgraph as in (a) but use a 1000 Hz square wave as input. Does the reconstructed ‘CT’ waveform at the output look as expected? Explain! Change the frequency of the square wave to 1200 Hz. Explain the reconstructed waveform that you see using explanations in both the time and frequency domains.

(c) Build the GRC flowgraph that uses a bandpass filter for the ‘CT’ waveform reconstruction. Display the reconstructed ‘CT’ waveform for a reconstruction filter with center frequency of `fc=Fs2` and `fc=2*Fs2`. Use the ’CT’ waveform that you see at the output for `fc=Fs2` as original ’CT’ input waveform (generate this waveform with sampling frequency `Fs1` using two signal sources) and verify that input and output ‘CT’ waveforms are identical.

(d) Extend the `pam10` function in the `pamfun` module to include the RCf pulse with rolloff parameter α and truncation index k (truncates to interval $-kT_B \leq t \leq kT_B$), and the manchester or diphase pulse shown in the following figure.

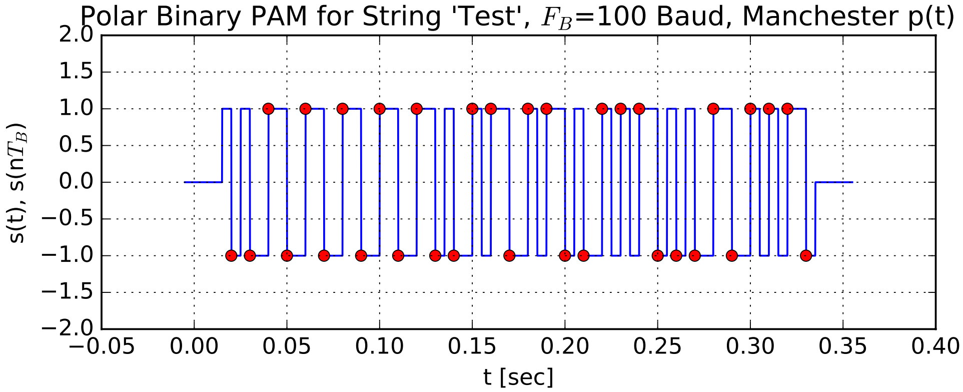
$$p(t) = \begin{cases} -1\,, & -T_B/2 \le t < 0\,, \\ 1\,, & 0 \le t < T_B/2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

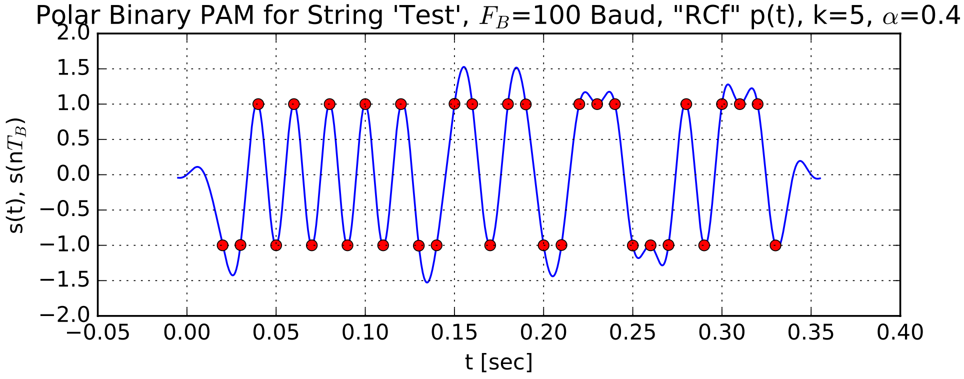


The header of the new `pam11` function in the `pamfun` module then looks as follows:

```
def pam11(sig_an, Fs, ptype, pparms=[]):
    """
    Pulse amplitude modulation: a_n -> s(t), -TB/2<=t<(N-1/2)*TB,
    V1.1 for 'man', 'rcf', 'rect', 'sinc', and 'tri' pulse types.
    >>>>> sig_st = pam11(sig_an, Fs, ptype, pparms) <<<<<
    where  sig_an: sequence from class sigSequ
           sig_an.signal():  N-symbol DT input sequence a_n, 0 <= n < N
           sig_an.get_FB():  Baud rate of a_n, TB=1/FB
           Fs:     sampling rate of s(t)
           ptype: pulse type from list
                  ('man','rcf','rect','sinc','tri')
           pparms not used for 'man','rect','tri'
           pparms = [k, alpha] for 'rcf'
           pparms = [k, beta]  for 'sinc'
           k:      "tail" truncation parameter for 'rcf','sinc'
                   (truncates p(t) to -k*TB <= t < k*TB)
           alpha: Rolloff parameter for 'rcf', 0<=alpha<=1
           beta:  Kaiser window parameter for 'sinc'
           sig_st: waveform from class sigWave
           sig_st.timeAxis():  time axis for s(t), starts at -TB/2
           sig_st.signal():    CT output signal s(t), -TB/2<=t<(N-1/2)*TB,
                               with sampling rate Fs
    """
```

An example of polar binary PAM using a manchester pulse, and an example using an RCf pulse with $\alpha = 0.4$ and $k = 4$ are shown in the next two graphs.





(e) Use the following Python commands to generate a PAM pulse $p(t)$ using your `pam11` function, and plot its FT $P(f)$ using `showft` function in the `showfun` module.

```
from pylab import *
import ecen4652 as ecen
import pamfun
import showfun
Fs = 44100                     #Sampling rate
FB = 350                       #Baud rate
N = FB                         #Number of symbols
an = hstack((zeros(round(N/2)), 1, zeros(round(N/2)-1)))
                               #Single pulse, padded with zeros
sig_an = ecen.sigSequ(an, FB, 0)
# ***** Set ptype, pparms here *****
sig_pt = pamfun.pam11(sig_an,Fs,ptype,pparms)  #Generate PAM pulse
sig_pt.set_t0(-(N/2)/float(FB))    #Place center of pulse at t=0
# ***** Set ff_parms here *****
showfun.showft(sig_pt,ff_parms) #Plot FT of pulse
```

Verify that $p(t)$ and $P(f)$ look right for the manchester and the RCf pulses that you added to the `pam11` function.

E2. PAM Analysis with Eye Diagrams. (a) Write a Python function, called `showeye` that can be used to display eye diagrams of digital PAM signals. The header of this function in the `showfun` module looks as follows.

```
def showeye(sig_rt, FB, NTd=50, dispparms=[]):
    """
    Display eye diagram of digital PAM signal r(t)
    >>>>> showeye(sig_rt, FB, NTd, dispparms) <<<<<
    where  sig_rt: waveform from class sigWave
           sig_rt.signal():  received PAM signal
                             r(t)=sum_n a_n*q(t-nTB)
           sig_rt.get_Fs():  sampling rate for r(t)
           FB:  Baud rate of DT sequence a_n, TB = 1/FB
           NTd: Number of traces displayed
           dispparms = [delay, width, ylim1, ylim2]
           delay:  trigger delay (in TB units, e.g., 0.5)
           width:  display width (in TB units, e.g., 3)
           ylim1:  lower display limit, vertical axis
           ylim2:  upper display limit, vertical axis
    """
```

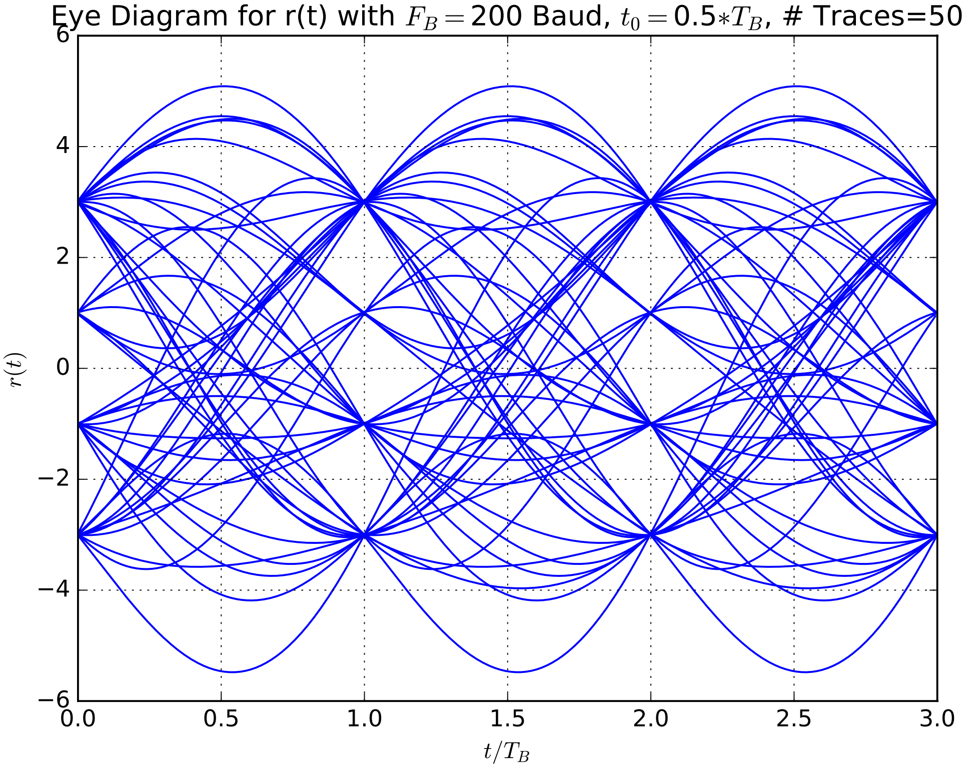
Here is some Python code to get you started with `showeye`:

```
    rt = sig_rt.signal()       # Get r(t)
    Fs = sig_rt.get_Fs()       # Sampling rate
    t0 = dispparms[0]/float(FB)  # Delay in sec
    tw = dispparms[1]/float(FB)  # Display width in sec
    dws = floor(Fs*tw)         # Display width in samples
    tteye = arange(dws)/float(Fs)  # Time axis for eye
    trix = around(Fs*(t0+arange(NTd)/float(FB)))
    ix = where(logical_and(trix>=0, trix<=len(rt)-dws))[0]
    trix = trix[ix]            # Trigger indexes within r(t)
    TM = rt[trix[0]:trix[0]+dws]   # First trace
    TM = vstack((TM, rt[trix[1]:trix[1]+dws]))
                                   # Second trace
    plot(FB*tteye, TM.T, ’-b’) # Plot transpose of TM
    grid()
    show()
```

To test your `showeye` function, use the following commands to generate a PAM signal and then display its eye diagram.

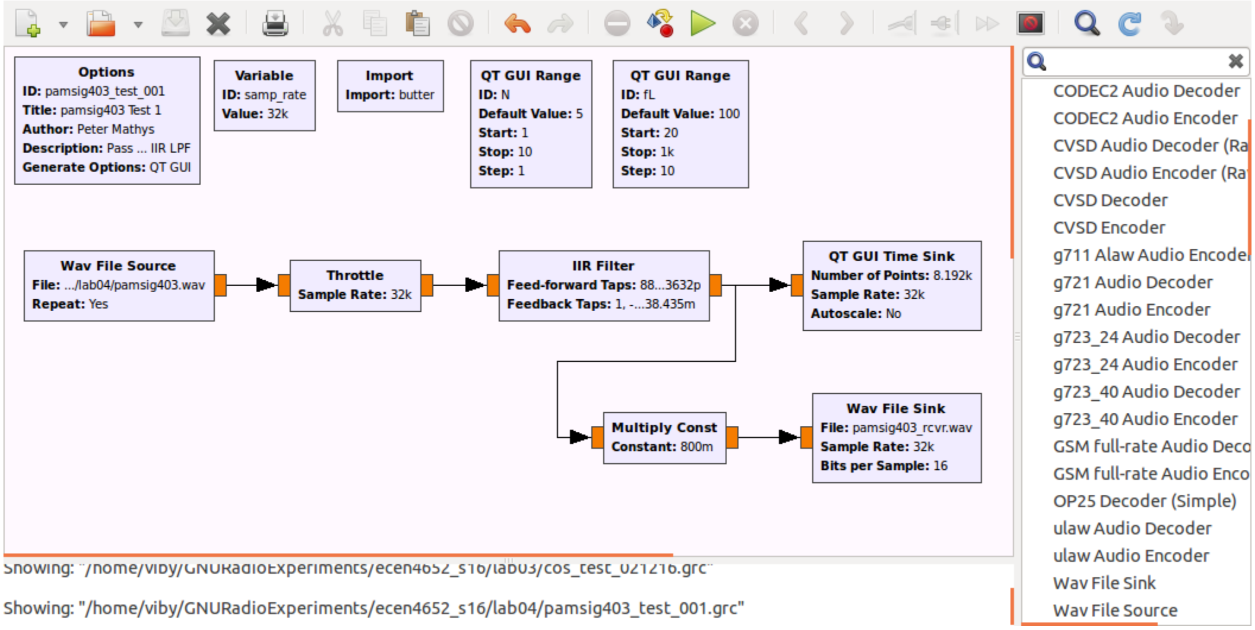
```
from pylab import *
import pamfun
import showfun
Fs = 44100                     # Sampling rate
FB = 200                       # Baud rate
NTd = 50                       # Number of traces displayed
N = NTd+10                     # Number of data symbols
L = 4                          # Number of data levels
dly = 0.5                      # Trigger delay TB/2
dn = floor(L*rand(N))          # Unipolar L-level random data
an = 2*dn - (L-1)              # Polar L-level DT sequence
tt, st = pamfun.pam11(an, FB, Fs, 'sinc', [20, 0])
                               # PAM signal, 'sinc' p(t)
showfun.showeye(st, Fs, FB, NTd, [dly, 3, -1.5*L, 1.5*L])
```

The result should look similar to the figure below.



(b) The signals in the files `pamsig401.wav` and `pamsig402.wav` are digital PAM signals with unknown parameters and impairments. Using plots in the time domain, `showft`, and `showeye`, find out as much as possible about these PAM signals. In particular, try to find F_B, $p(t)$, and the number of levels L. Also, try to characterize channel impairments (if any) like ISI, noise, and timing jitter.

(c) The file `pamsig403.wav` contains a binary unipolar PAM signal with random data and rectangular $p(t)$. The sampling rate is $F_s = 32000$ Hz and the symbol rate is $F_B = 100$ baud. Use this file as input in GNU Radio and build the following flowgraph.



Options
ID: pamsig403_test_001
Title: pamsig403 Test 1
Author: Peter Mathys
Description: Pass ... IIR LPF
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

Import
Import: butter

QT GUI Range
ID: N
Default Value: 5
Start: 1
Stop: 10
Step: 1

QT GUI Range
ID: fL
Default Value: 100
Start: 20
Stop: 1k
Step: 10

Wav File Source
File: .../lab04/pamsig403.wav
Repeat: Yes

Throttle
Sample Rate: 32k

IIR Filter
Feed-forward Taps: 88...3632p
Feedback Taps: 1, -...38.435m

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 32k
Autoscale: No

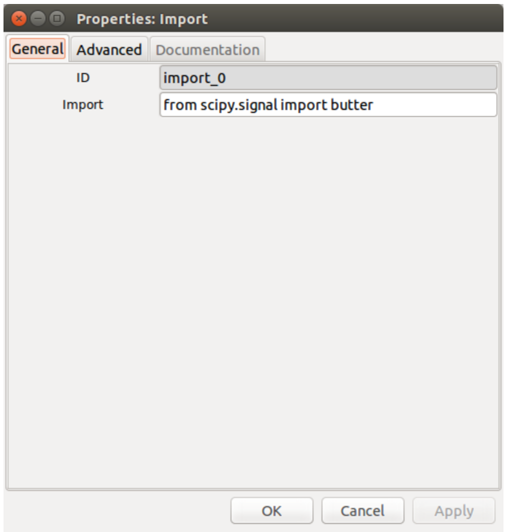
Multiply Const
Constant: 800m

Wav File Sink
File: pamsig403_rcvr.wav
Sample Rate: 32k
Bits per Sample: 16

Showing: "/home/viby/GNURadioExperiments/ecen4652_s16/lab03/cos_test_021216.grc"

Showing: "/home/viby/GNURadioExperiments/ecen4652_s16/lab04/pamsig403_test_001.grc"

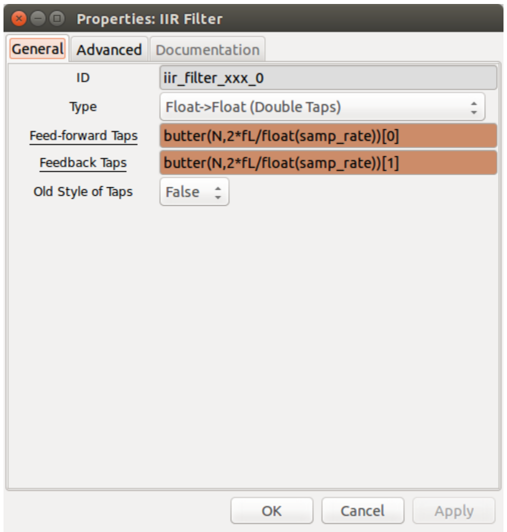
The filter is an IIR Butterworth lowpass filter with filter order N and -3dB frequency f_L in Hz. SciPy has a utility function `scipy.signal.butter` that can be used to compute the numerator and the denominator polynomial coefficients of the rational transfer function $H(z)$ of an N-th order DT Butterworth filter with critical frequency (or frequencies) W_n (as a fraction of the Nyquist frequency $F_s/2$). To import the butter function to the GNU Radio Companion, set the following properties in an import block (from the Misc category).



Properties: Import

ID: import_0
Import: from scipy.signal import butter

In the Properties of the IIR Filter block, specify the Feed-forward (numerator of $H(z)$) and the Feedback (denominator of $H(z)$) Taps as follows.



Properties: IIR Filter

ID: iir_filter_xxx_0

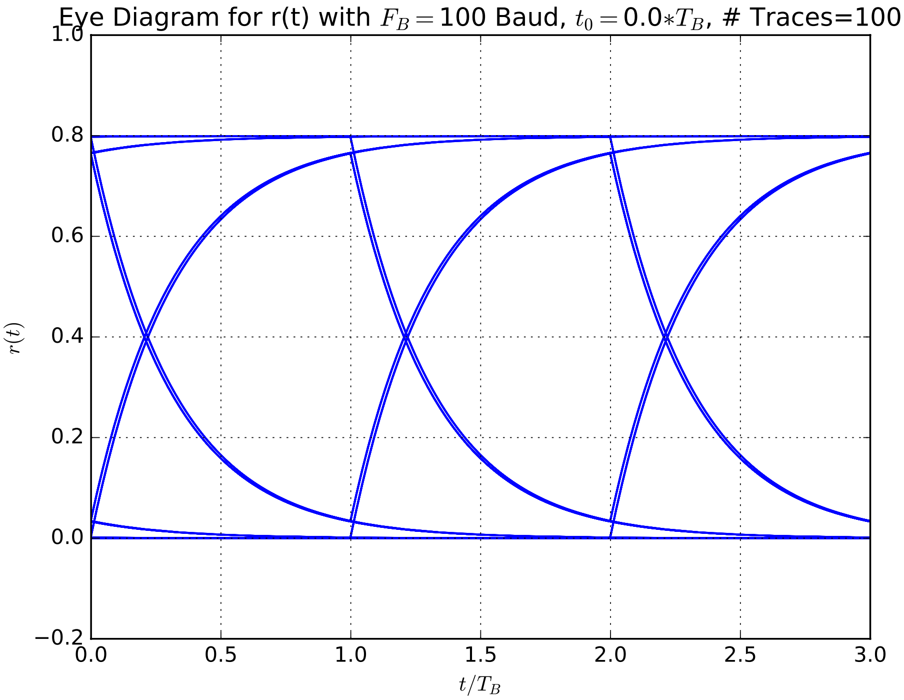
Type: Float->Float (Double Taps)

Feed-forward Taps: butter(N,2*fL/float(samp_rate))[0]

Feedback Taps: butter(N,2*fL/float(samp_rate))[1]

Old Style of Taps: False

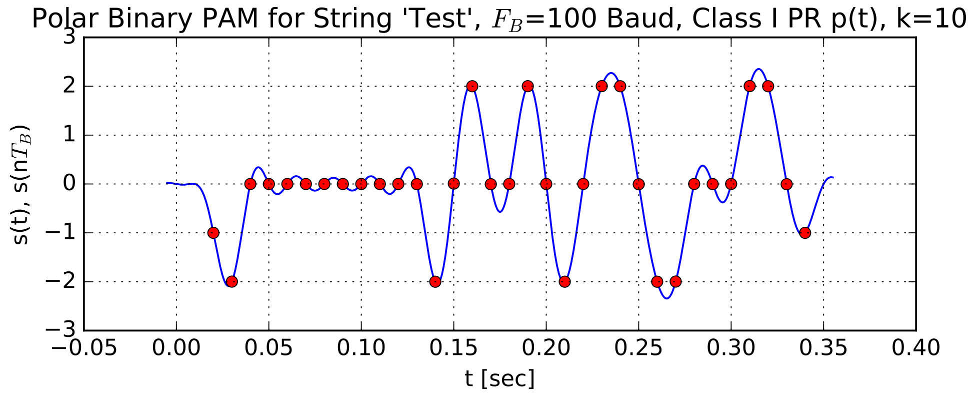
The goal of this experiment is to analyze the effect that a simple lowpass channel, as modeled by the Butterworth filter, has on the quality of the received PAM signal. Since GNU Radio (currently) does not come with an eye diagram display block, we will record the filter output in a wav-file and then display an eye diagram using the Python `showeye` function. Make wav-file recordings for $N = 5$ for $f_L = 0.25, 0.5, 1.0, 2.0 \times F_B$. Look at the corresponding eye diagrams and interpret the results. What can you conclude for the channel bandwidth that is needed for reliable detection of the transmitted data? Look at the eye diagram below that was obtained using the GNU Radio flowgraph of this experiment. Can you determine the N and the f_L that was used for the Butterworth filter? Hint: Remember from your circuits class what the charge and discharge graphs for a simple RC lowpass look like.



E3. Class I PR Signaling. (Experiment for ECEN 5002, optional for ECEN 4652) **(a)** Extend your `pam11` function so that the specification `'pr1'` for `ptype` will generate a PAM signal using the class I PR pulse

$$p(t) = \frac{\sin(\pi t/T_B)}{\pi t/T_B\,(1-t/T_B)}\,,$$

for `pt`. Similar to the `'sinc'` pulse, the "tails" of this $p(t)$ have to be truncated to length kT_B, where k is an integer specified in `pparms[0]`. A sample plot of a binary polar PR I signal (without precoding) is shown below.



(b) Compare the FT of a PR I pulse and the FT of a "sinc" pulse for $k = 10$ in the same way as described in E1c.

(c) The signal in `pr1sig401.wav` contains a polar binary PR I signal with $F_B = 100$ Hz that was precoded and level shifted as described in the introduction. The signal contains an ASCII message (8 bits/char, LSB first). See if you can recover the message.

(d) GNU Radio does not come with an eye diagram block. Propose and implement a flowchart in gnuradio companion, using only existing blocks, that displays a reasonable approximation to an eye diagram. Hint: The QT GUI Time Sink can display up to 10 input signals simultaneously.