

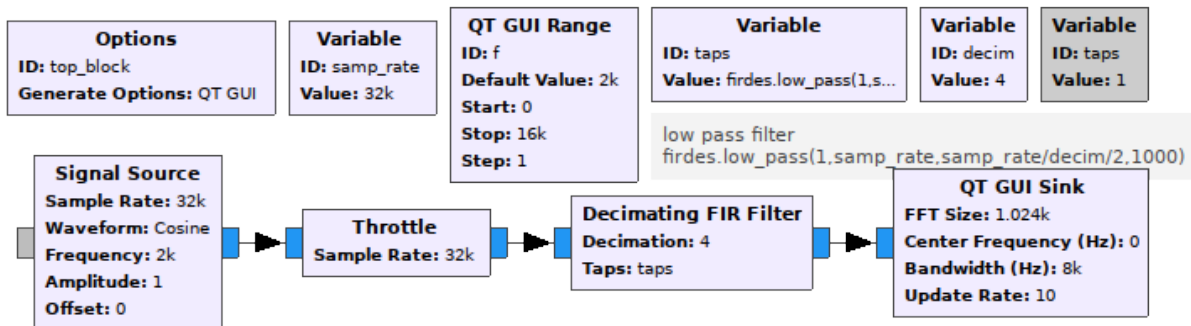
Basic GNU Radio Companion tutorial: having fun with discrete time digital signal processing

J.-M Friedt, July 1, 2018

Our objective in this tutorial is on the one hand to get the user to become familiar with some of the core concepts of discrete time signal processing, and on the other hand to exhibit the flexibility of GNURadio and its graphical frontend GNURadio Companion to prototype such concepts on synthetic and real signals. We address the issues of aliasing and filtering in the context of frequency transposition and cross correlation for detecting a known pattern in a noisy signal.

1 Frequency axis and decimation: the need for filtering

The abscissa of the Fourier transform is from $-f_s/2$ to $+f_s/2$ so that decimating by N will yield a Fourier axis from $-f_s/(2N)$ to $+f_s/(2N)$. Decimating aims at completing two tasks: zooming on the relevant signal in the frequency axis close to baseband (centered on 0-Hz frequency), and reducing the number of samples processed. However, simple decimation by N by taking one in N samples (in Matlab syntax, `x(1:N:end)`) might bring some unwanted spectral component close to baseband and prevent extracting the wanted information. Hence, decimating must always be prefixed with a low pass filter step cutting at less than $f_s/(2N)$. This low pass filtering is performed using a convolution with a Finite Impulse Response (FIR) filter, available as a GNURadio signal processing block. Care must be taken though to wisely select the transition width Δf , which dictates the length of the taps b defining the filter. Indeed, a Fourier transform on P samples generates bin sizes of f_s/P width. Selecting a transition width Δf smaller than f_s/P will not allow fitting the filter shape if P is too small. On the other hand, selecting a small Δf will require a large $P = f_s/\Delta f$ i.e. a long filter with many coefficient, requiring this extensive computational power to generate each output sample $y_n = \sum_k b_k \cdot x_{n-k}$.



2 Frequency transposition

Multiplication – yielding addition of the argument of the trigonometric functions – will lead to possible aliasing during the frequency transposition process if no care is taken to filter out the aliased images.

Indeed, the periodic signal $\exp(j\omega_1 t)$ multiplied by the periodic signal $\exp(j\omega_2 t)$ yields $\exp(j(\omega_1 + \omega_2)t)$: multiplication in the time domain is addition in the frequency domain. For a signal discretized in time, there is no reason for $(\omega_1 + \omega_2)/(2\pi)$ to comply with Shannon's theorem of being below half the sampling rate, making the sum frequency easily aliased, possibly close to baseband and preventing the demodulation of the information carried by the carrier.

Care must hence be taken to filter this aliased image: rather than filling the taps of the low pass filter of the Xlating FIR filter, selecting a low-pass filter to reject this aliased image will prevent such issues in most cases. Sometimes, the image will reach exactly back to baseband and cannot be filtered: eliminating the unnecessary spectral component *prior* to frequency shifting and decimation is mandatory, as shown in the next section.

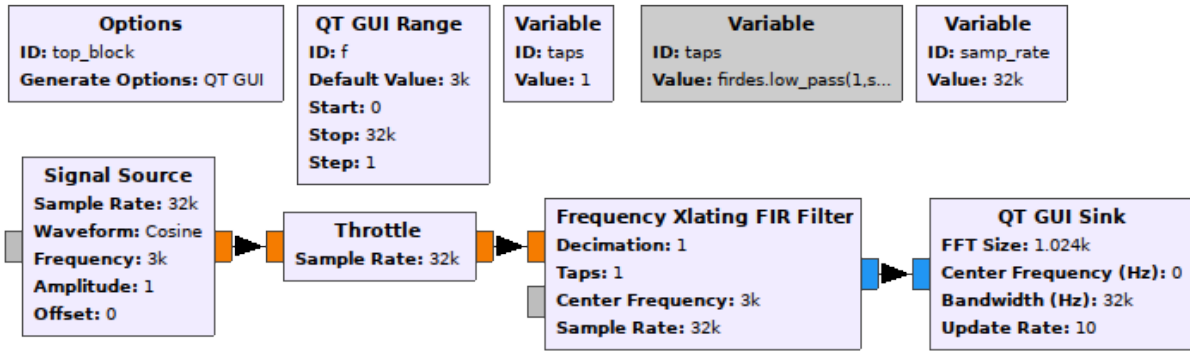


Figure 1: Flowchart demonstrating frequency transposition of a real signal characterized by an even spectrum.

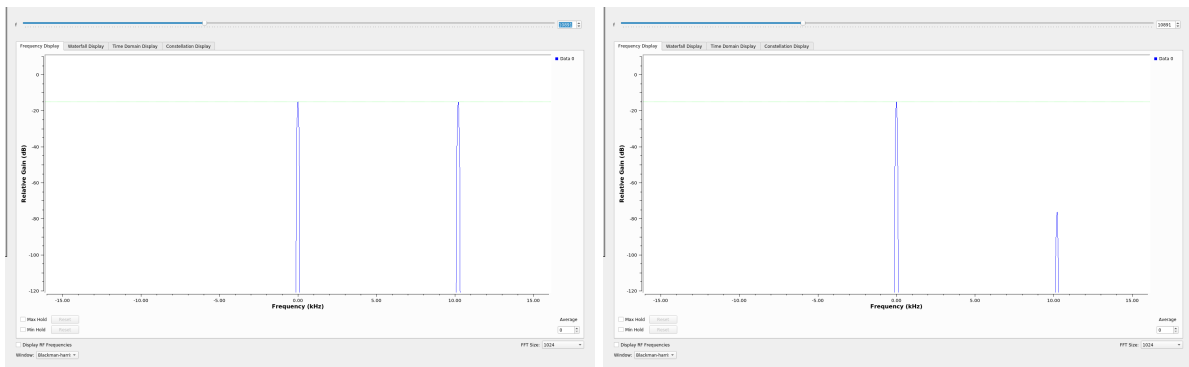
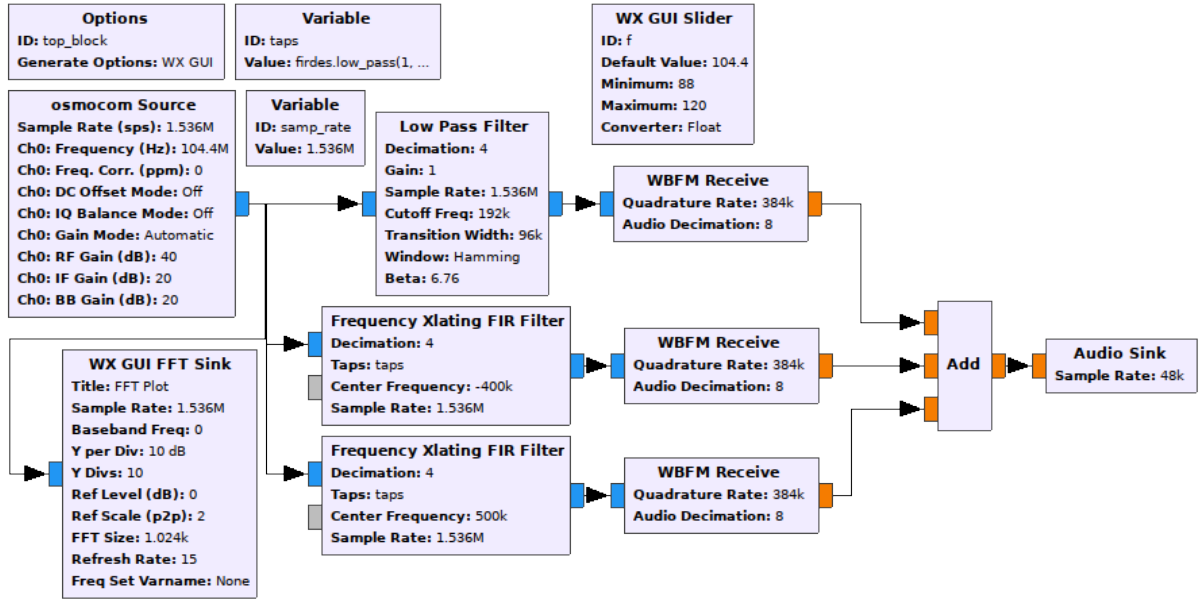


Figure 2: Left: no filtering, so that the aliased image of the negative component of the spectrum comes in the positive part of the spectrum close to baseband. Right: filtered image.

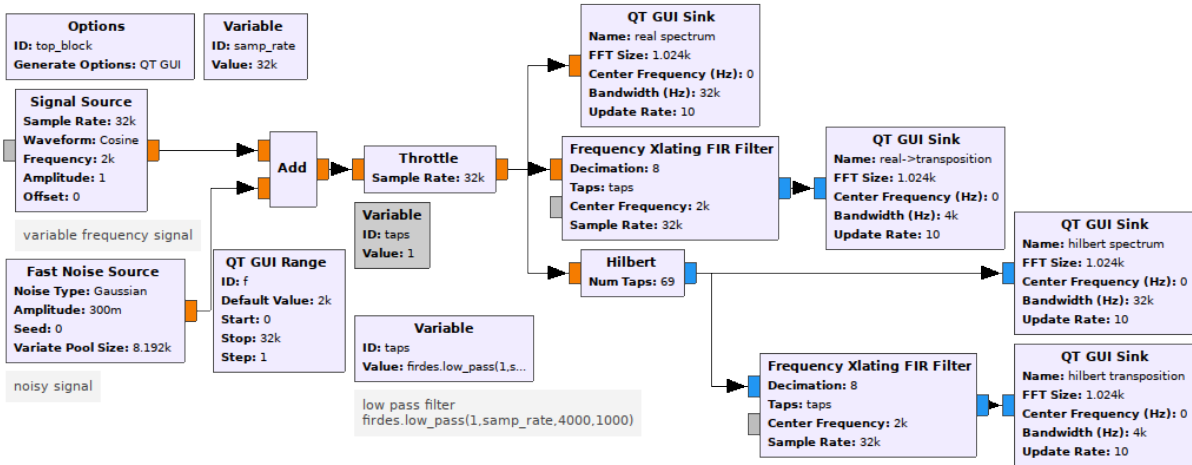
Demonstrate the concept by listening to two commercial broadcast FM stations simultaneously

The hardware DVB-T dongle brings, through mixing with a local oscillator generated by a Phase Locked Loop (PLL), the radiofrequency to baseband (centered on 0-frequency). The remote oscillator is never at the exact same frequency than the receiver oscillator: some tuning is needed to adjust the local oscillator to the remote carrier frequency. The problem is enhanced with direct frequency spectrum spreading in which the emitted carrier frequency jumps continuously from one value to another in a known sequence. The receiver PLL cannot track fast enough these carrier jumps, and a two-step mixing is performed by first applying a coarse transposition (from radiofrequency to within the sampling bandwidth) using the hardware PLL and a second transposition is applied using a software implementation of the Numerically Controlled Oscillator. Since $\exp(j\omega_{RF}t) \cdot \exp(j\omega_{NCO}t) = \exp(j(\omega_{RF} + \omega_{NCO})t)$, both oscillator frequencies are added to bring the received signal to baseband. This technique is so common that it is implemented in GNU Radio Companion as the Xlating FIR Filter. We will use this block to listen simultaneously to multiple FM broadcast stations: again the only characteristic of the receiver that matters is the bandwidth (and hence the sampling rate), while the carrier frequency is of little interest other than in defining antenna dimensions and compliance with regulations.



3 Hilbert transform when recording real signals

$\cos(\omega t) = \frac{1}{2}(\exp(j\omega t) + \exp(-j\omega t))$ so that the Fourier transform exhibits two peaks at $-\omega$ and $+\omega$. This information is redundant and the even Fourier distribution only duplicates the information from the positive frequency range in the negative range. If we wish to process the signal, we might want to get rid of the negative spectral component to avoid aliasing unwanted spectral components during the frequency transposition.



A signal with only a positive spectral component is defined by $\exp(j\omega t) = \cos(\omega t) + j \sin(\omega t)$. We see that with respect to the previous expression $\cos(\omega t)$ when considering the real signal, a $j \sin(\omega t)$ was added. Getting rid of the negative part of the spectrum can be generalized hence by adding to a real signal an imaginary part in quadrature to the real signal. This transform is called the **Hilbert transform**, and is classically applied in RADAR signal processing when the real signal only is collected, e.g. with a hardware frequency transposition using a mixer for example. It is of course provided as a GNURadio signal processing block.

In this demonstration, we show how getting rid early in the processing chain of the negative part of the even spectrum by creating the quadrature imaginary component helps further processing steps. This processing chain was used when decoding DCF77 (77.5 kHz carrier) with a sound card (192 kHz sampling rate).

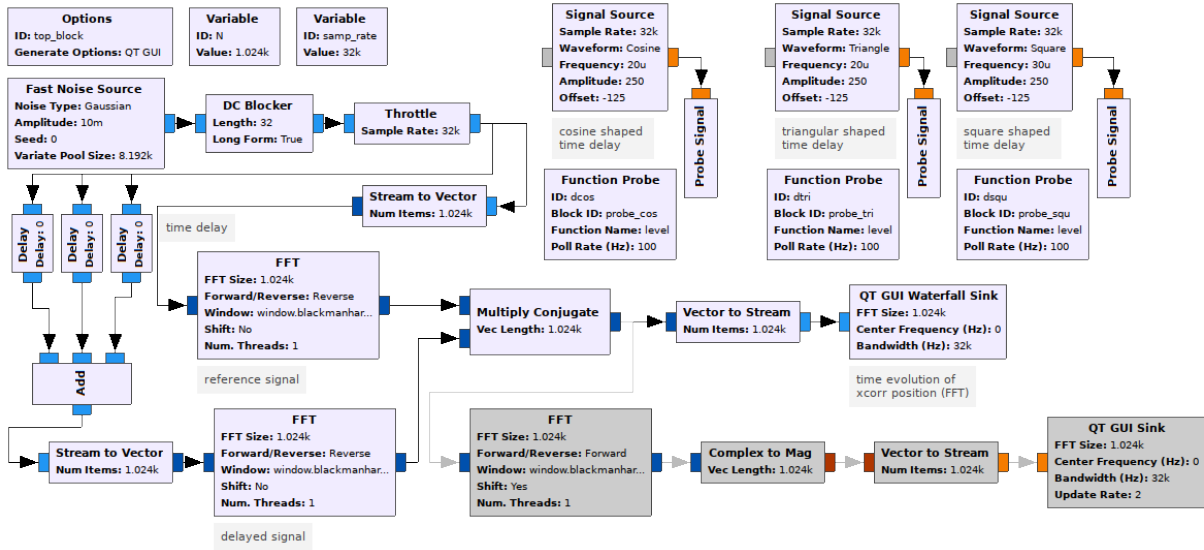
Identify the unwanted frequency components of the signal – to be eliminated by filtering prior to signal processing to extract the information content on the modulation of the carrier – when recording the DCF77 signal and transposing to baseband.

4 Correlation

All CDMA and most communication protocols will start with a header to be identified before decoding the message. Correlation is the mathematical tool for finding a pattern in a noisy signal: $corr(\tau) = \int p(t) \cdot s(t + \tau) dt$. This operation looks similar to the convolution, except for the lack of time reversal in the second term. As with the convolution, an N delay correlation on an N sample dataset requires N^2 multiplications. As in the convolution, computing a correlation is most efficiently performed in the Fourier domain. Indeed

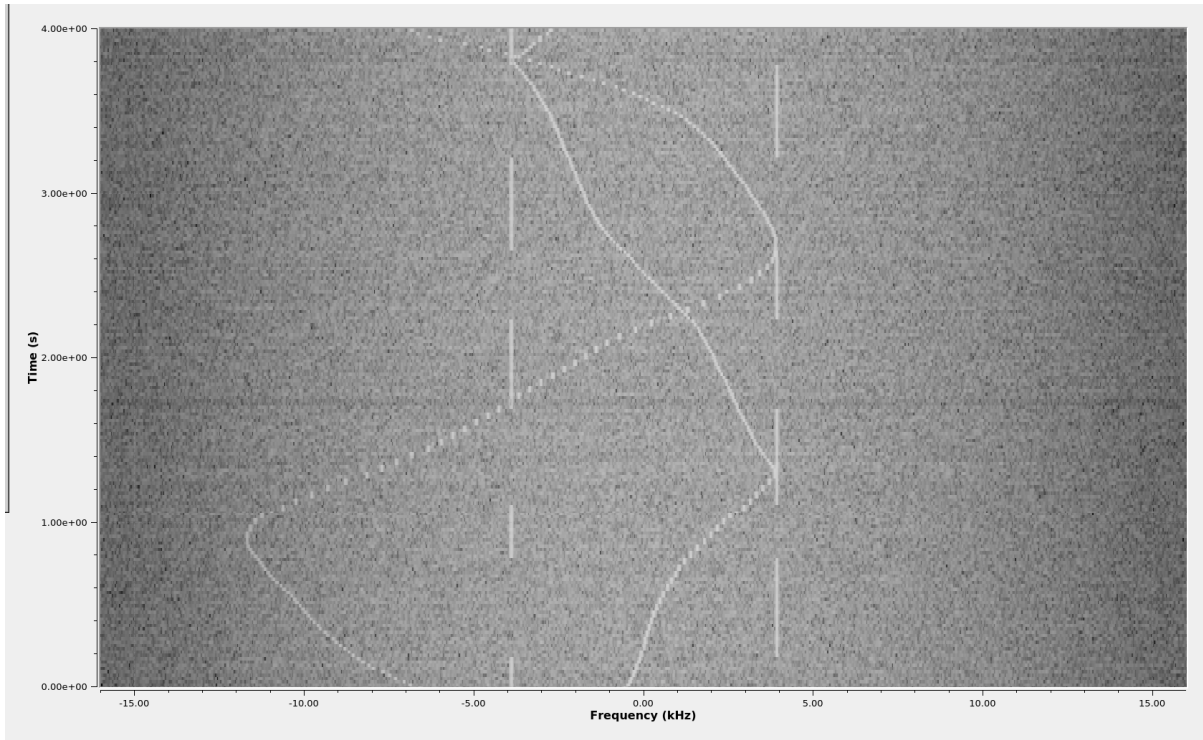
1. the convolution theorem tells us that $FT(\underbrace{conv(x, y)}(\tau)) = FT(x) \cdot (FT(y))$ with FT the Fourier transform
$$\int x(t)y(\tau-t)dt$$
2. since $\exp(j\omega t)^* = \exp(-j\omega t)$, with $*$ the complex conjugate, reverses the time of the argument, then
3. $corr(x, y)(\tau) = FT(x) \cdot FT^*(y)$

Since the Fourier transform is, through its Fast Fourier Transform, a $N \ln_2(N)$ complexity algorithm, computing the correlation through the product of the Fourier transform of the reference channel with the complex conjugate of the Fourier transform of the surveillance channel is most efficient. This concept can be demonstrated in GNU Radio Companion using the flowchart of Fig. 4.



In this example, three time delays are introduced in a random source whose autocorrelation function is ideally a Dirac peak centered on 0-delay. The sum of the three delayed copies, representing three targets reflecting the incoming random but known signal, are visible in the correlation map displayed as a waterfall (Fig. 4).

While the sum of the three signals is well visible on the waterfall chart, their shape does not match the expected cosine and triangle variations of the delays. The `throttle` block aims at preventing the scheduler from saturating the processing capabilities of the host computer by feeding the processing blocks with too many samples in the absence of hardware bandwidth limiter, but does not claim accurate timing. Hence, depending on processing load, the time will run at varying speed, leading to the fluctuating shape of the delays. Nevertheless, the capability of the cross-correlation function to extract multiple echoes as



defined by time delays of a known random pattern emitted by a radiofrequency source is demonstrated in an efficient way through Fourier transform to compute the cross-correlation. As usual with RADAR processing, the sampling rate defines the bandwidth and hence the range resolution (cross-correlation peak width), while the number of samples on which the cross-correlation is computed (i.e. number of samples of the Fourier transforms) is equivalent to the pulse repetition period (number of samples divided by the sampling rate), defining the furthest target to be detected before spatial aliasing occurs.