INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Report TP “Protocols for Connected Objects”

Isabella Mi Hyun Kim, Hoang Diep Vu
5th year ISS

January 2020

1 - Introduction

Software-Defined Radio (SDR) is an RF communication system that allows us to perform signal processing tasks, that are typically done by hardware, using software. This is very different to the traditional RF communication methods that are full hardware based. Usually, all the signal processing is done via hardware and we can choose - through software/firmware - which type of modulation we want to perform (this is the Software-Controlled Radio). The SDR gives us a wireless and digital approach to signal processing.

During these practical works, we will use a National Instruments USRP-2900 software defined radio transceiver. With this transceiver, we'll be able to make the transposition of the received signals around the zero frequency and the analog to digital conversion. This way we'll be able to retrieve real radio signals from real radio sources operating at different frequencies.

In the TP sessions, we'll be using a software called GNU Radio which provides us with a free toolkit that can be used to develop software-defined radios. Firstly, we'll demonstrate that, in the case of narrowband signals, the IQ transceivers (In-phase/Quadrature) allow transmissions with frequency transportation without altering data (First part). After that, we worked on a real time demodulation of an FM broadcasting signal (Second part) and a VOLMET AM signal (Third part).

2 - First part: The In-phase / Quadrature Software-Defined Radio transceiver

In this session, we studied how our acquisition device, the National Instruments USRP-2900 transceiver, makes the demodulation of the signal. For that we did a mathematical study of the core of SDR: the I/Q signals.

The demodulator (see Figure 1) has inputs for the radio signal ($r_{RF}(t)$) and the local oscillators, represented by the sinusoidal waves. And it has two outputs to represent our signal in quadrature: the $r_R\,(t)$, the real part of the signal on the x-axis, and the $r_I\,(t)$ part of the signal on the y-axis. Combined they form the IQ signal that can therefore be used by the GNU Radio software.

Note: For this part, we'll use the notations shown in the Annexe.

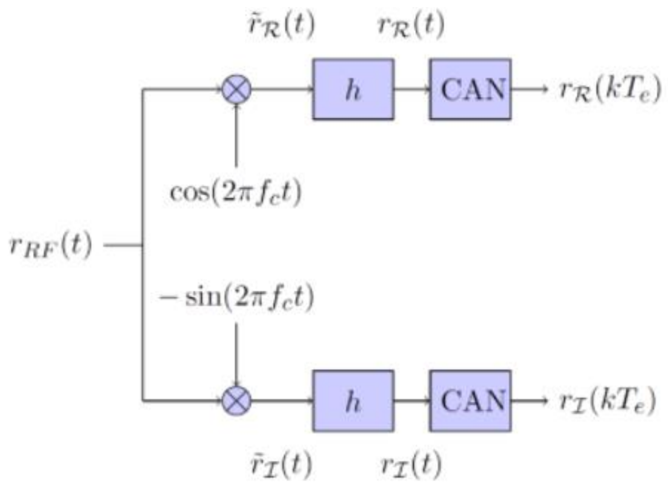

Figure 1: Diagram block of the transceiver.

2.1 - Expression of the received signal in the imaginary and real domain

First, we'll assume that we're making an effective transmission, that is: the received signal is similar to the transmitted one (equation 1).

$$r_{RF}(t) = s_{RF}(t) \tag{1}$$

So the received signal $r_{RF}(t)$ can also be represented by:

$$r_{RF}(t) = s_{RF}(t) = s_R\,(t)\,.\,cos(2\pi f_O t)\; - \; s_I\,(t)\,.\,sen(2\pi f_O t) \tag{2}$$

So, by multiplying our received RF signal by the frequency oscillators (as shown in Figure 1), we have:

$$\widetilde{r_R}(t) \;=\; r_{RF}(t)\,.\,cos(2\pi f_c t) \tag{3}$$

$$\widetilde{r_I}(t) \;=\; -\,r_{RF}(t)\,.\,sin(2\pi f_c t) \tag{4}$$

By substituting (2) in (3) and (4), we get the set of equations:

$$\begin{cases} \widetilde{r_R}(t) = \; s_R(t).\,cos(2\pi f_O t).\,cos(2\pi f_c t) - s_I(t).\,sen(2\pi f_O t)\,.\cos(2\pi f_c t) \\ \widetilde{r_I}(t) = -s_R(t).\,cos(2\pi f_O t).\,cos(2\pi f_c t) - s_I(t)\,.\,sen(2\pi f_O t)\,.\,sen(2\pi f_c t) \end{cases} \tag{5}$$

With the trigonometric relations, we can express the real and imaginary parts of the received signal as:

$$\begin{cases} \widetilde{r_R}(t) = \frac{s_R(t)}{2}\ [\cos(2\pi(f_O + f_c)\, t\ + \cos(2\pi(f_O - f_c)\, t\,] - \frac{s_I(t)}{2}\ [\, sen(2\pi(f_O + f_c)t\ + sen(2\pi(f_O - f_c)t\,] \\ \widetilde{r_I}(t) = -\frac{s_R(t)}{2}\ [\, sen(2\pi(f_O + f_c)t\ - sen(2\pi(f_O - f_c)t\,] + \frac{s_I(t)}{2}\ [\cos(2\pi(f_O - f_c)\, t - \cos(2\pi(f_O + f_c)\, t\,] \end{cases} \quad (6)$$

This result indicates that this is a baseband signal, whose range of frequencies is measured from close to 0Hz to the cut-off frequency defined by the carrier.

2.2 - Characterisation of the H filter

To determinate the characteristics of the H filter, we'll take $f_c = f_O$ (we consider a translation in the baseband by heterodyning). So we establish that:

$$f_c + f_O = 2.f_c = 2.f_O \quad (7)$$

With this hypothesis and by substituting (7) in the relations found at (6), we get:

$$\begin{cases} \widetilde{r_R}(t) = \frac{s_R(t)}{2}\ [cos(4\pi f_O t)\ + 1\,] - \frac{s_I(t)}{2}\ [\, sen(4\pi f_O t)] \\ \widetilde{r_I}(t) = -\frac{s_R(t)}{2}\ [\, sen(4\pi f_O t)\,]\ + \frac{s_I(t)}{2}\ [\, 1 - cos(4\pi f_O t)\,] \end{cases} \quad (8)$$

We see from that with the relations found in (8) we can reconstruct the transmitted signal ($s_R(t)$ and $s_I(t)$). For that we're only interested in the spectrum which is centered at DC, as shown in (8). The remaining spectrums have to be eliminated and we need a filter with a gain of 2 so we'll use a low pass filter, which is used for the elimination of unwanted higher frequency components.

By applying the Fourier transform, we get the following representation in the frequency domain:

$$\begin{cases} \widetilde{R_R}(f) = \frac{1}{4}\ [2S_R\,(f)\ + S_R\,(f + 2f_O)\ +\ S_R\,(f - 2f_O)\ - jS_I(f + 2f_O) + jS_R(f - 2f_O)] \\ \widetilde{R_I}(f) = \frac{1}{4}[2S_I\,(f)\ - S_I\,(f + 2f_O)\ -\ S_I\,(f - 2f_O)\ - jS_R(f + 2f_O) + jS_R(f - 2f_O)] \end{cases}$$

These equations are represented in the Figure 2 below:

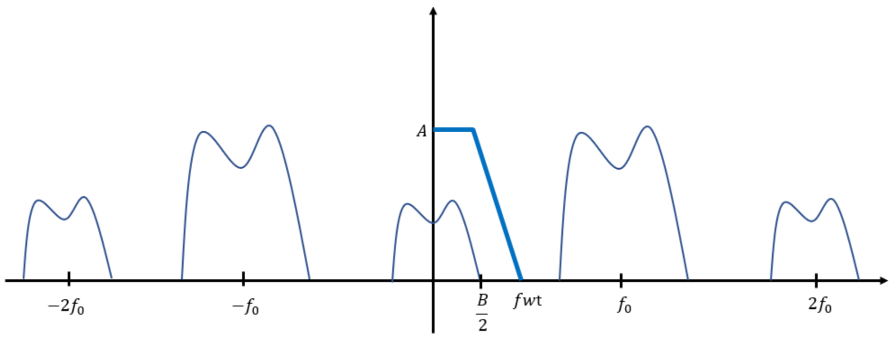

Figure 2: Representation of the received signal in the frequency domain and the filter (blue line).

By the figure we can see that to choose the cut-off frequency of our filter, we consider that the ideal filter would have a cut-off frequency of exactly the same size of half the bandwidth of the signal.

So we have an H filter with the following characteristics:

$$\frac{B}{2} < f_{cut-off} < f_O - \frac{B}{2} \tag{9}$$

$$A = 2$$

2.3 - Wide-band signal demodulation

The IQ receiver cannot work with wide-band signals, case where $f_O > \frac{B}{2}$(see Figure 4). Differently from the narrow-band case (see Figure 3), in the wide-band case the IQ receiver needs non-causal filters and that is impossible to do analogically.

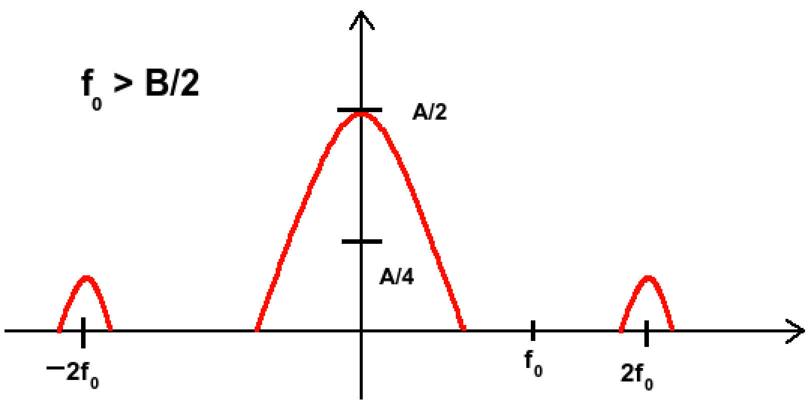

Figure 3: Narrow-Band f_0 > B/2.

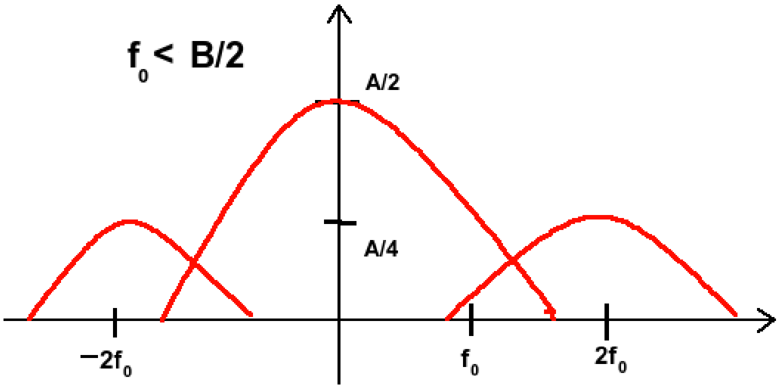

Figure 4: Wide-Band f_0 < B/2.

2.4 - Definition of the sampling period to recover the signal

The Shannon Theorem states that to recover any signal we need to have a sampling frequency rate (f_e) that is at least 2 times higher than the maximum frequency of the signal (f_{max}) that we want to recover. Putting this mathematically we have the following restriction:

$$f_e \geq 2.f_{max}$$

Considering $f_{max} = \frac{B}{2}$, we get:

$$f_e \geq B \rightarrow T_e \leq \frac{1}{B}$$

2.5 - Transceiver architecture

About the transceiver architecture chosen (see Figure 1), we could interchange the stages of frequency transposition and analog to digital conversion and still retrieve our desired signal back. The problem is that our system would have to operate at a higher frequency ($f_e > 2.(f_O + \frac{B}{2})$). In that case, the equipment would be much more expensive.
For example, the ADC12DJ5200RF RF-sampling analog-to-digital converter (ADC) from Texas Instruments can directly sample input frequencies from DC to above 10 GHz but it can cost up to $3000.

2.6 - Expressing the signal in the frequency domain

In its spectral representation (see Figure 5), the signal is expressed in positive and negative frequencies (symmetrically on the ordinate axis). The purely positive analytic signal associated to the real narrow-band signal, to conserve all its power, needs to have an amplitude that is twice as important. And the complex envelope is obtained after re-centring the analytic signal.

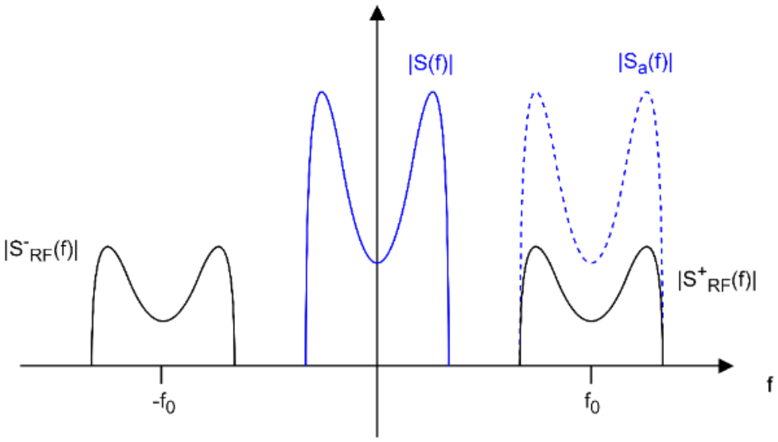

Figure 5: Spectral representations of a real narrow-band signal, of its analytic signal and of its complex envelope.

Now, we consider the narrowband signal case:

$$s(t) = s_R(t) \,.\, cos(2\pi f_O t) \; - \; s_I(t) \,.\, sen(2\pi f_O t) \tag{10}$$

In this case, the representation in positive frequencies (or negative) is sufficient and it is possible to hide the value of the carrier frequency f_O.

By applying the transformation in frequency to the signal to (10) we have:

$$S(f) \;= F\{s_R\,(t) \,.\, cos(2\pi f_O t) \; - \; s_I\,(t) \,.\, sen(2\pi f_O t)\}$$

$$\Leftrightarrow \; S(f) \;= F\{s_R\,(t) \,.\, cos(2\pi f_O t)\} \; - F\{\, s_I\,(t) \,.\, sen(2\pi f_O t)\}$$

$$\Leftrightarrow \; S(f) \;= S_R\,(f) \, * \, \frac{1}{2}[\delta(f - f_O) \; + \; \delta(f + f_O)] + S_I(f) \; * \; \frac{1}{2}[\delta(f - f_O) \; - \; \delta(f + f_O)]$$

$$\Leftrightarrow \; S_{RF}(f) \;= \frac{1}{2}[S_R\,(f - f_O) \; + \; S_R\,(f + f_O)] \; + \; \frac{j}{2}[S_I\,(f - f_O) \; + \; S_I\,(f + f_O)]$$

The representation in positive frequencies (or negative) is sufficient and it is possible to hide the value of the carrier frequency f_O, so we have the received signal:

$$S_{RF}(f) \;= \frac{1}{2}[S_R\,(f - f_O) \; + \; \boldsymbol{j} S_I\,(f - f_O)]$$

So, for the analytic signal we have:

$$\begin{cases} S_a(f) = 2\,S_{RF}\,(f)^{+}, & if\ f > 0 \\ 0, & if\ f < 0 \end{cases}$$

Now switching to the temporal domain:

$$S_a(f)\ = [S_R\,(f)\ +\ \boldsymbol{j}S_I\,(f)] * \delta(f - f_O) \quad ,\forall f\ > O$$

$$s_a(t)\ = [s_R\,(t)\ +\ \boldsymbol{j}s_I\,(t)].\,e^{j2\pi f_O t}$$

For the complex envelope we have in the frequency and time domain:

$$S(f)\ =\ S_a(f + f_O)\ = S_R\,(f)\ +\ \boldsymbol{j}S_I\,(f)$$
$$s(t) = s_R(t)\ + j.\,s_I(t)$$

3 - Second part: Reception of frequency modulation (FM) broadcasting

In this part, we have a FM broadcasting recording file and the aim is to succeed in generating an audible signal and extracting appropriate information. In order to complete this mission we are going to use the GNURadio development environment.

First of all, we implement the frequency analysis processing chain on GRC as the following schema:

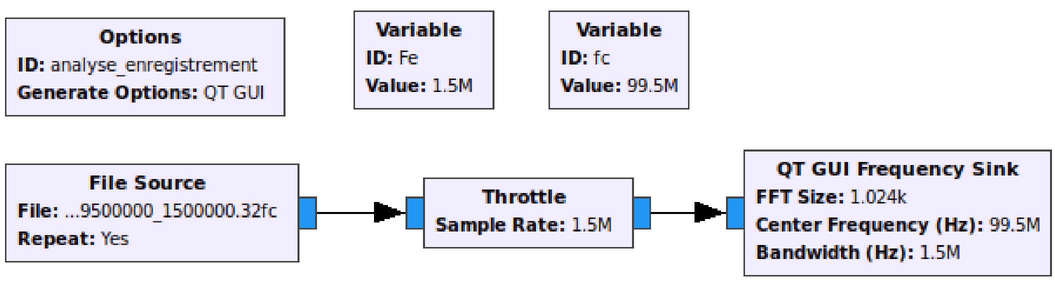

Figure 6 : Frequency analysis processing chain.

- The *Options* block sets special parameters for the flow graph, and the *Generate Options* controls the type of code generated : QT GUI in this case.
- The *Variable* block allows us to define a variable with an ID which can be called in other blocks later and the assigned value.
- The *File Source* block takes the original signal file in parameter in order to read stream from it and open it as a source of items into a flowgraph

- The *Throttle* block throttles flow of samples such that the average rate does not exceed the sample rate that we have defined.
- The *QT GUI Frequency Sink* block is a graphical sink to display multiple signals in frequency in which we can generally define the FFT Size, Center Frequency and Bandwidth.

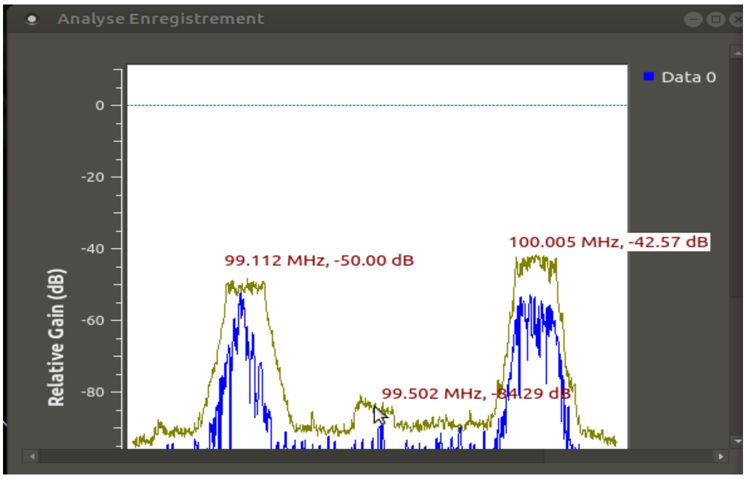

Figure 7: Spectrum analysis result.

Through the Figure 7 we can distinguish 3 radio stations: RFM Toulouse (99.1 MHz), Nostalgie Toulouse (99.5 MHz) and Skyrock (100.0 MHz).

Then, in order to know the transmission quality we have calculated the signal-to-noise ratio in decibel with the following method:

$$P_{dB} = 10\log_{10}(P) \Leftrightarrow P = 10^{\frac{P_{db}}{10}}$$

For the 1st signal (RFM Toulouse) we detect the noise at -90 dB: $P_{dB} = 10 \log_{10} (10^{-4+9}) = 40$ dB.

In the same way, we have 1 dB and 50 dB respectively for the 2nd and 3rd signal. For a good understanding of messages the signal-to-noise ratio must not be under a certain threshold, under 0 dB the listening becomes uncomfortable. So regarding the values found we can conclude that the signal can be demodulated.

Moreover the approximate bandwidth of a channel can be calculated at 3 dB of the peak. At 3 dB, inside of this bandwidth we have at least half of signal power. For example: RFM Toulouse, at -53dB, has a BW = 0,183 MHz. So for the other stations we found:

$$BW_{Nostalgie} = 0,1 \text{ MHz}$$
$$BW_{Skyrock} = 0,136 \text{ MHz}$$

In the next step, we are going to use frequency transposition and low-pass filtering to realize channel extraction because we would like to receive each RF broadcasting station separately by using a new processing chain. The aim is to center the wished signal:

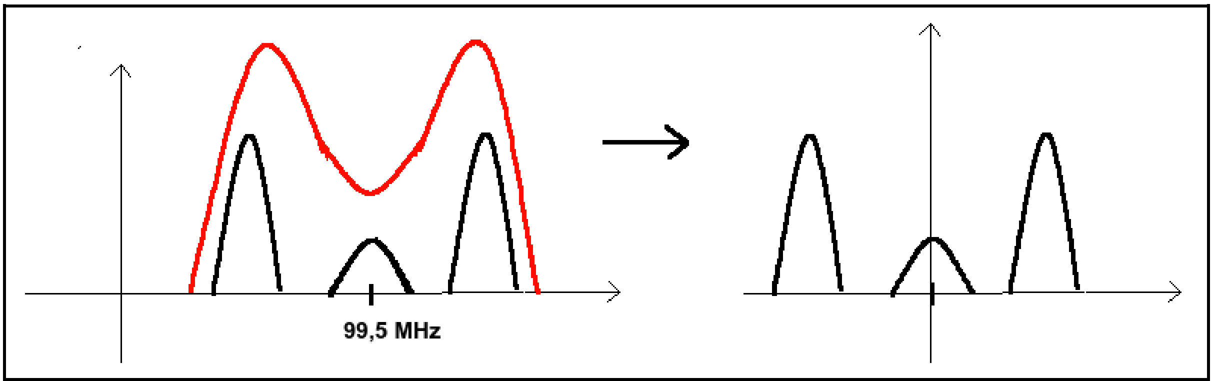

Figure 8: Centering the signal with exponential envelope.

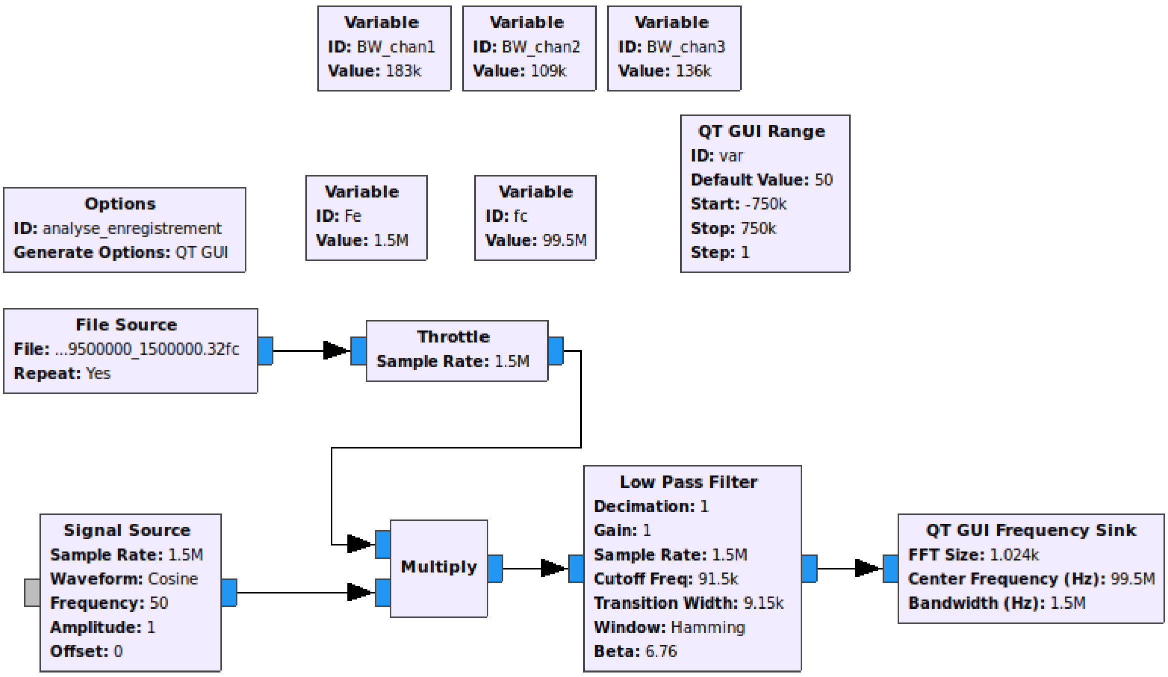

Figure 9 : Channel extraction graph flow.

In this graph flow we added a Multiply block which multiplies across all input streams, a Signal Source block to generate the complex exponential envelope, and the QT GUI Range block in which we can define the frequency offsets which are -750 kHz and 750 kHz because the sampling frequency F_e = 1.5 MHz.

During this step we must pay attention to frequency offset value which should not be greater than the sampling frequency F_e otherwise the signal will be repeated periodically.

Afterwards, we introduce the Low Pass Filter block which have the following characteristics:

- Sample rate: F_e
- Cut-off frequency: BW / 2
- Transition width: 10% of the cutoff frequency

We found the bandwidth via the Carson rule: $B_{FM} = 2\ (\Delta f + f_m) \approx 256$ kHz

Now, we are showing that $y_l[k] \ = \ e^{j.k_f.\sum_{i=0} m[i]} \ + b[k]$

So we refer to the expressions **(1)** $S_{RF}(t) \ = \ A(t)\ cos(2\pi f_0 t \ + \ \varphi(t)), t \in \mathbb{R}$

And **(10)** $\boldsymbol{S_{RF}(t) \ = \ A(t)\ cos(\ 2\pi f_0 t \ + \ \frac{\Delta f}{max(|m(t)|)} \int_{-\infty}^{t} m(u).\, du\ )}$.

We identify: $\varphi(t) \ = \boldsymbol{\frac{\Delta f}{max(|m(t)|)} \int_{-\infty}^{t} m(u).\, du}$

With $s_R(t) \ = A(t)\ cos(\varphi(t))$
$s_I(t) \ = A(t)\ sin(\varphi(t))$

We remember from the previous part:

$$
\begin{aligned}
s(t) \ &= \ s_R(t) \ + \ j.s_I(t) \\
&= \ A(t)\ cos(\varphi(t)) \ + \ j.A(t)\ sin(\varphi(t)) \\
&= \ A(t) \ \ e^{j\varphi(t)} \\
&= \ A(t)\ e^{j\frac{\Delta f}{max(|m(t)|)} \int_{-\infty}^{t} m(u)\ du}
\end{aligned}
$$

Then we discretize: $s[k] \ = \ A[\frac{k}{F_e}]\ e^{j\frac{\Delta f}{max(|m(t)|)} \sum_{i=0}^{k} m(i)} \ + \ b[k]$

For FM, $A[\frac{k}{F_e}]$ is constant and will be noted A.

So, $y_l[k] \ = \ e^{j.k_f.\sum_{i=0} m[i]} \ + b[k]$ where $k_f \ = \ \frac{\Delta f}{max(|m(t)|)}$

Basically, we would like to center and obtain the demodulated signal as following:

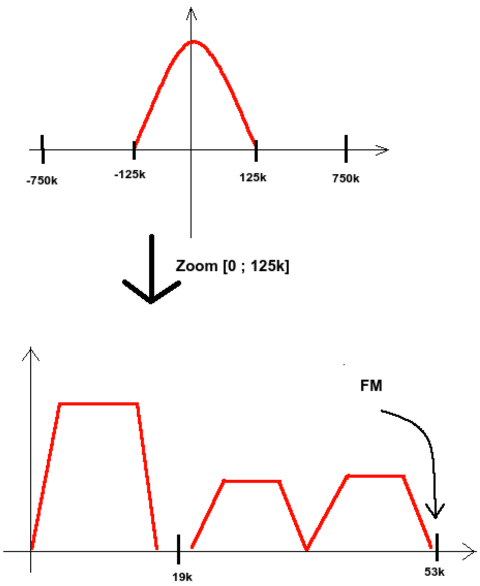

Figure 10: Theoretical spectrum of the demodulated channel.

Now, we want to frequency demodulate and play audio stream by the use of a new processing chain based on the previous one:

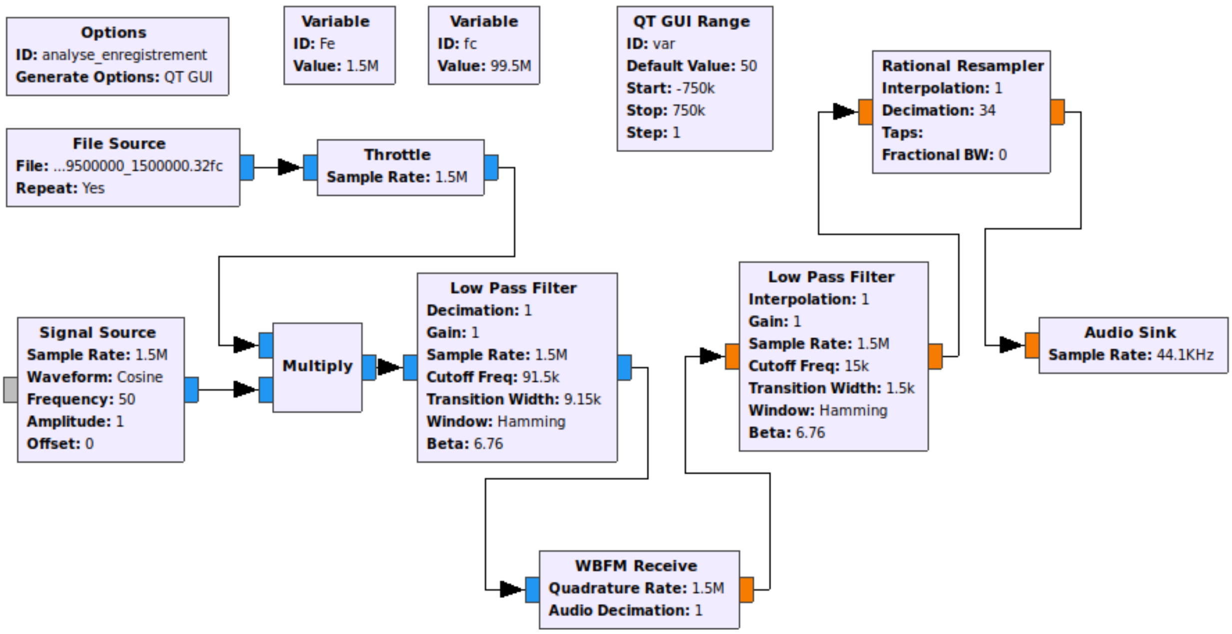

Figure 11: Frequency demodulator graph flow.

We pay attention to fill the correct decimation value of the Rational Resampler block here we want to have a sample rate of 44.1 kHz at the Audio Sink output so we should have $\frac{1.5MHz}{44.1kHz} = 34$, and the cutoff frequency of the low pass filter must be fixed reasonably in order to have an audible sound at the input. Here we have chosen 15 kHz.

Consequently, here is the result:

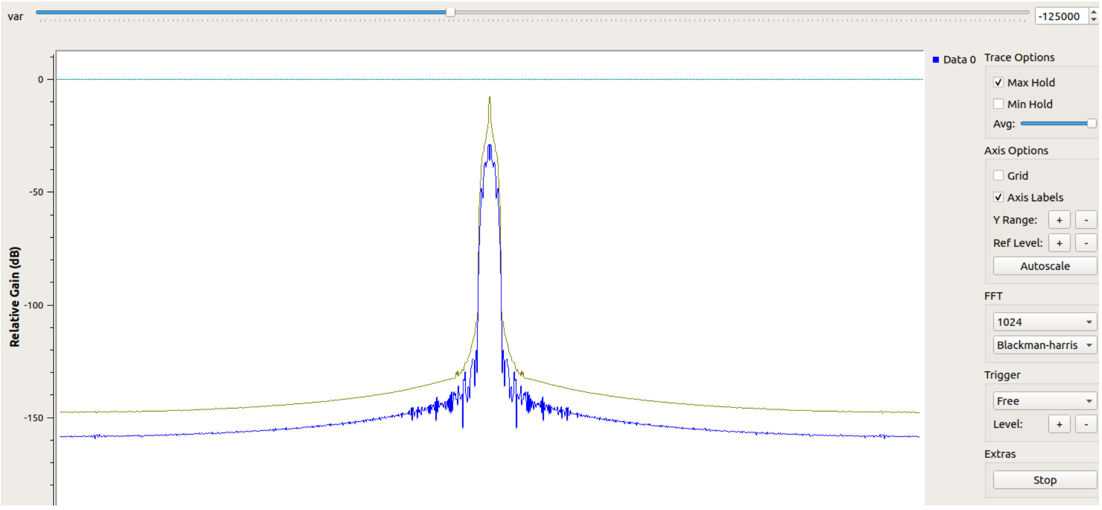

Figure 12: Simulated spectrum of the demodulated channel.

Finally, Jordi has won the Sam Smith album at the first radio station (-500 kHz), “YMCA” song at the second one and “Counting Stars” of OneRepublic at the last one.

4 - Third part: Reception of VOLMET messages in AM-SSB

So to summarize we have done in the previous parts a modulation in frequency and now we are going to experiment the modulation in amplitude. To do this, we have realized the following graph flow:

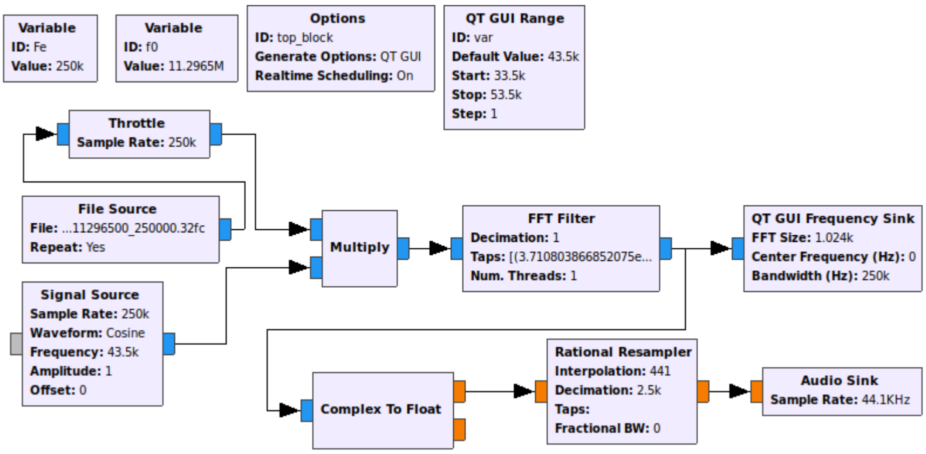

Figure 13: VOLMET graph flow.

If we Plot the modulus of the discrete Fourier transform in decibels, between $f_O - \frac{F_e}{2}$ and $f_O + \frac{F_e}{2}$, by using QT GUI Frequency Sink block:

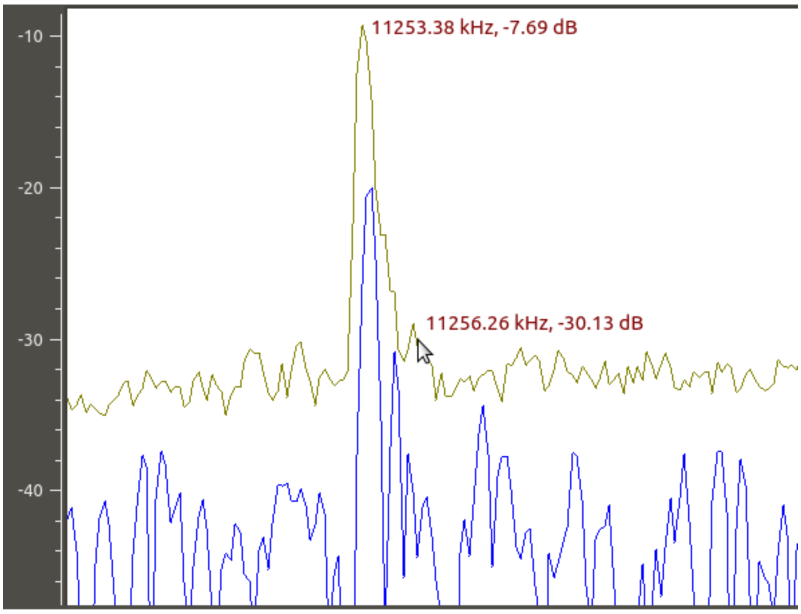

Figure 14: modulus of the discrete Fourier transform.

We can see a peek at **11.253 MHz** which matches the Military One station of Great Britain according to: http://www.dxinfocentre.com/volmet.htm. In addition, according to the following figure shown below:

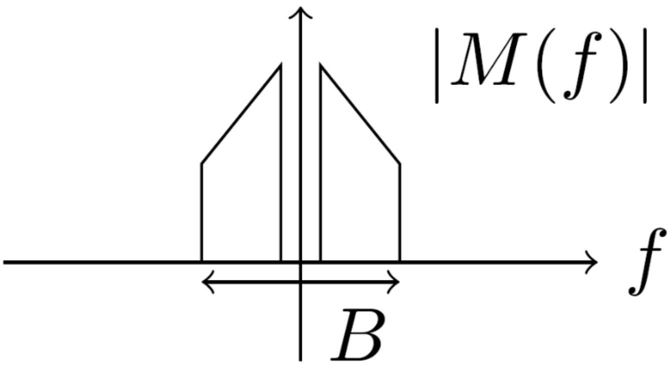

Figure 15: Spectral representation of the amplitude of the real signal to be transmitted.

We can see at Figure 14 that there is a second peek at **11.256MHz** which is equivalent to the right part of the spectral representation of the signal to be transmitted.

So the middle frequency is calculated by subtracting f_0 to **11.253 MHz** (the peek frequency we have found above: 11.2965 - 11.253 = **43.5 kHz**.

Moreover we would like to plot $|S_{RF}(f)|$ in function of the polarity of the second term in (16): $s(t) = m(t) \pm j\,\mathscr{H}\{m(t)\}$

So $\qquad S(f) = m(f) \pm sgn(f)m(f)$

(with $\mathscr{H}\{m(f)\} = -j\, sgn(f)m(f)$)

$$S_a(f) = m(f - f_O) \pm sgn(f - f_O)\, m(f - f_O)$$

- Case $\oplus$: $S_a(f) = m(f - f_O) + sgn(f - f_O)\, m(f - f_O)$

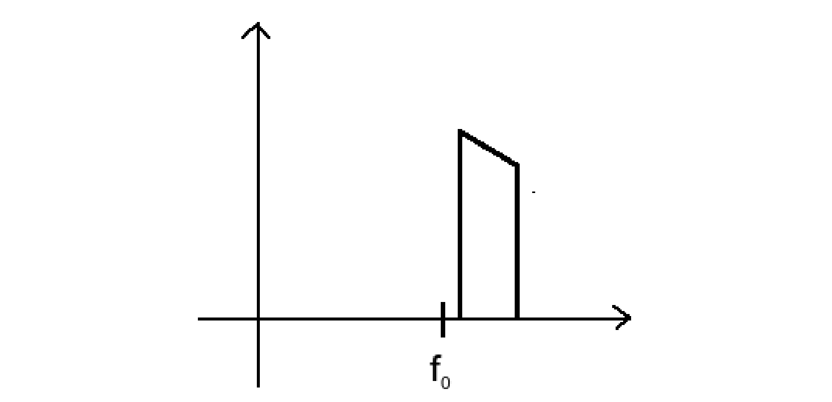

Figure 16: Positive case.

- Case $\ominus$: $S_a(f) = m(f - f_O) + sgn(-f + f_O)\, m(f - f_O)$

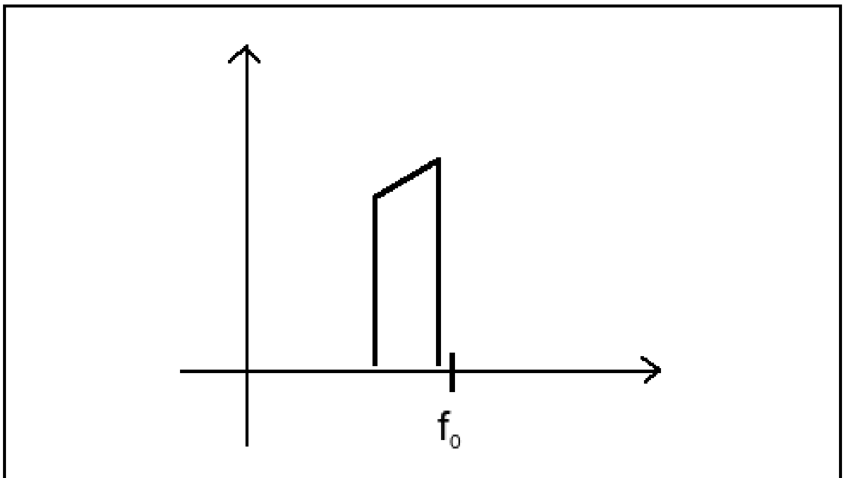

Figure 17: Negative case.

From the expression (15): $s_{RF}(t) = R_e(s(t)\, e^{\,j2\pi f_O t})$ we have:

$$|S_{RF}(f)| = \frac{1}{2}\,[|S_a(f)| + |S_a^*(f)|]$$

Consequently :

- Case $\oplus$:

$$|S_{RF}(f)| = \frac{1}{2}\,[m(f - f_O) + sgn(f - f_O)m(f - f_O) + m(-f + f_O) + sgn(-f - f_O)m(-f - f_O)]$$

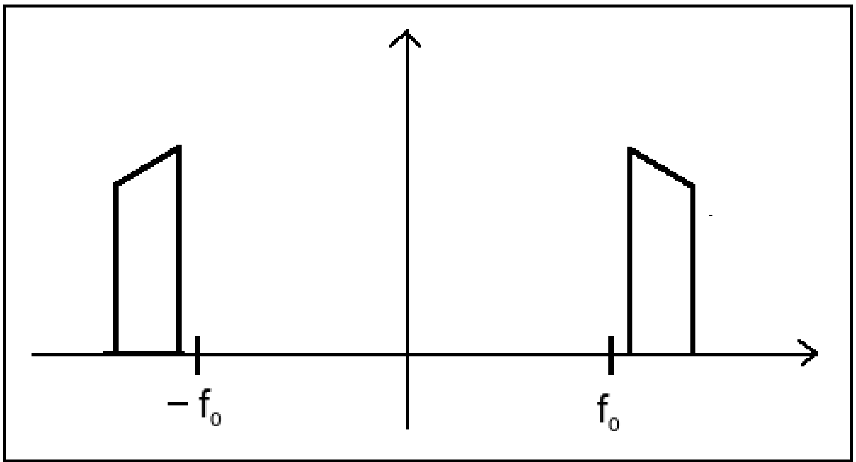

Figure 18 : Positive case.

- Case ⊖ :

$$|S_{RF}(f)| = \frac{1}{2}[m(f-f_O) + sgn(-f+f_O)m(f-f_O) + m(f-f_O) + sgn(f+f_O)m(-f-f_O)]$$

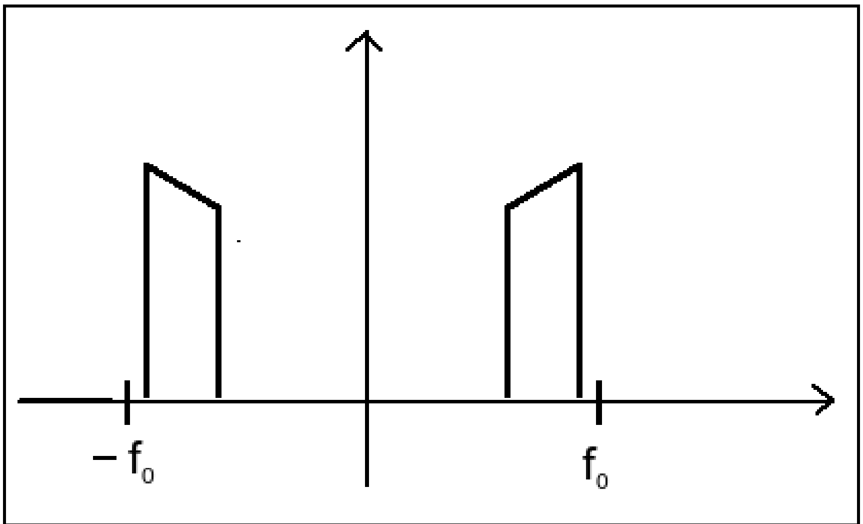

Figure 19: Negative case.

The FFT filter taps are generated by the *filter design tool.* In Figures 20 and 21 below we show the characteristics of the generated filter. We chose 0 Hz as the start of the pass band (actually, the software doesn't allow us to put exactly 0 as starting frequency, so we chose 1 Hz) and 3000 Hz as ending of the pass band. For the transition width, an ideal filter would have 0Hz (but the software doesn't accept that as input either), so we defined a transition of 10% the upper cut-off frequency.

After defining the specifications of the filter, the software generates a value that allows us to have a notion of the complexity of the filter by the number of taps. We could see that as the number of taps value changed, the time to computation time to implement the filter also changed.

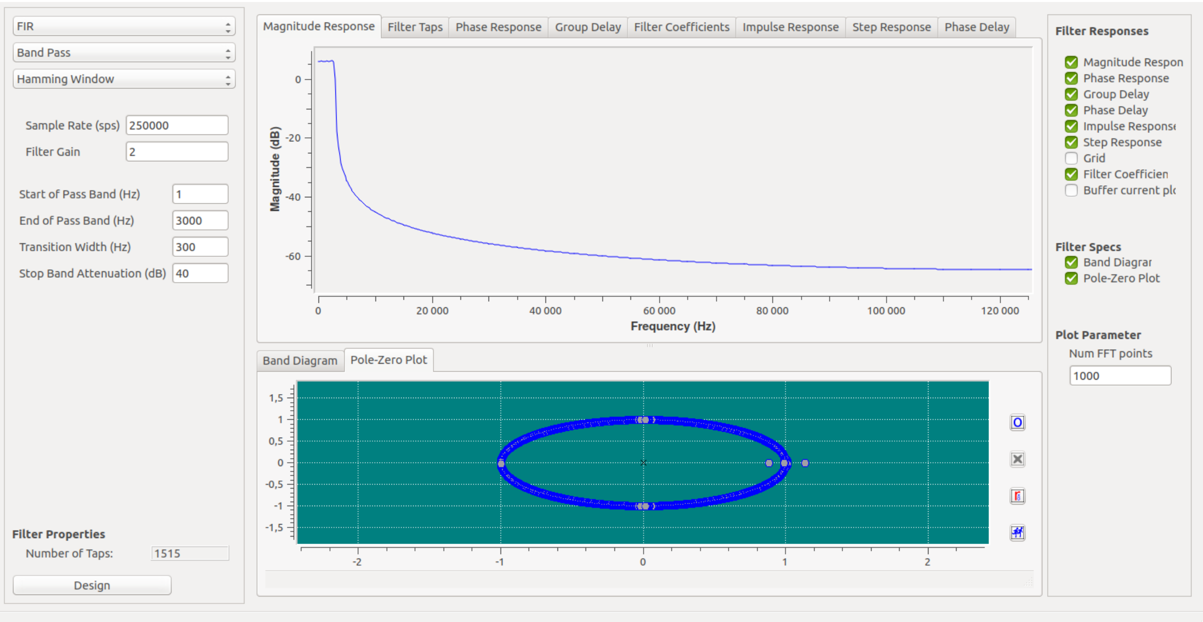

Figure 20: Magnitude response of the filter with pole-zero plot.

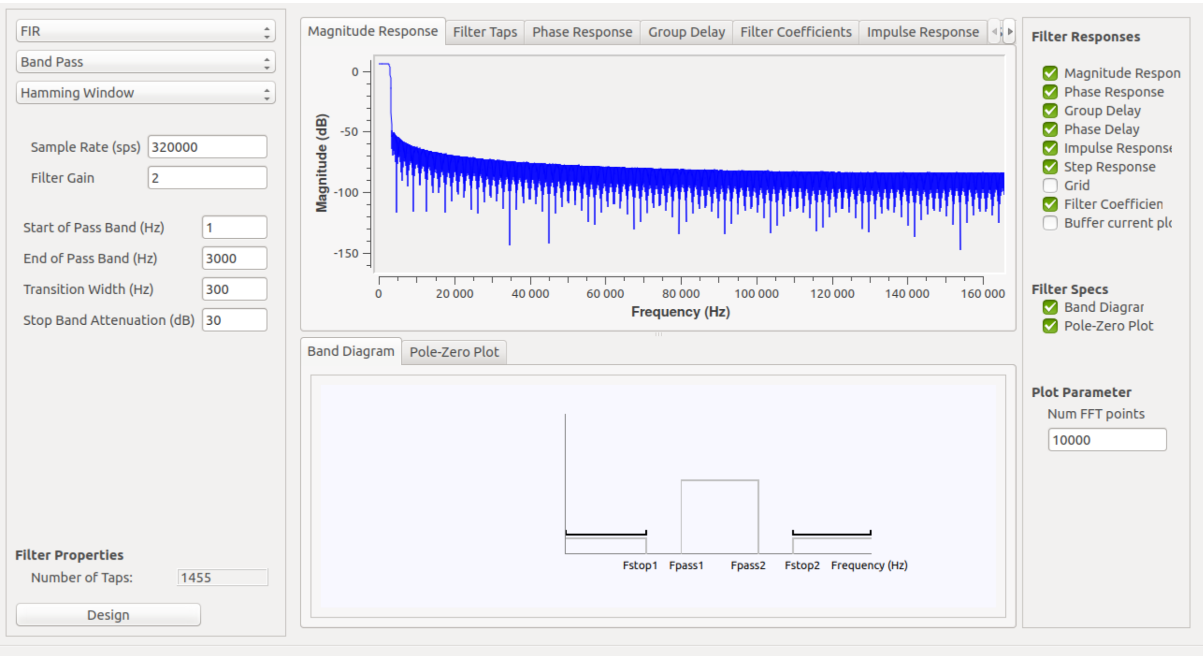

Figure 21: Filter with band diagram.

Finally, we have tried to listen to the audio stream but we could only hear noise and nothing was audible. We came to a conclusion that the amplitude modulation is easier to make but more susceptible to noises than the frequency modulation.

As a conclusion, we would like to give a special thanks to **Laurent Chasserat** for having guided and advised us all along the TPs.

Annexe

$s_{RF}(t)$	transmitted signal in time domain
$s_R(t)$	real part of the transmitted signal in time domain (in-phase)
$s_I(t)$	imaginary part of the transmitted signal in time domain (quadrature)
$r_{RF}(t)$	received signal in time domain
$\widetilde{r_R}(t)$	real part of the received signal in time domain
$\widetilde{r_I}(t)$	imaginary part of the received signal in time domain
$r_R(t)$	real part of the received signal in time domain after filtering
$r_I(t)$	imaginary part of the received signal in time domain after filtering
$r_R(k, T_e)$	real part of the received signal after filtering and sampling
$r_I(k, T_e)$	imaginary part of the received signal after filtering and sampling
$s_a(t)$	analytic signal
$s(t)$	complex envelope
$S_R(f)$	Fourier transform of real part of the received signal
$S_I(f)$	Fourier transform of imaginary part of the received signal
$\widetilde{R_R}(f)$	Fourier transform of real part of the received signal, after filtering
$\widetilde{R_I}(f)$	Fourier transform of imaginary part of the received signal, after filtering