INTRODUCTION TO **SOFTWARE-DEFINED RADIO**

ANALOG DEMODULATION OF SIGNALS USING GNURADIO

Elie Taillardat
Jonathan Malique

PTP ISS
23/12/2018

Summary

INTRODUCTION

During our last year of studying at INSA Toulouse, we focus on communication for connected objects. More precisely, we worked on a type of radio communication: the **software-defined radio** (SDR) where components that are traditionally implemented in hardware are implemented by means of software.

Then, we had three practical sessions about the reception of real communication signals, about processing these signals numerically using **GNURadio** software. The radio spectrum is divided in specific parts (one for broadcasting, one for aeronautical communication, one for mobile phones, …) and we can record almost all the radio spectrum with a **single receiver**.

The first practical session was about the **In-Phase/Quadrature transceivers**: we studied the transmission with frequency transportation without altering the data in the case of narrowband signals. The second was about processing FM broadcasting signal, demodulating it in real time signals to manage to listen FM radio on our computers. Finally, we worked on VOLMET AM broadcasting.

KEYWORDS

IQ	**In-Phase/Quadrature**: a sinusoid with angle modulation can be decomposed into two amplitude-modulated sinusoids that are offset in phase by one-quarter cycle ($\frac{\pi}{2}\ rad$). These modulated sinusoids are known as in-phase and quadrature components.
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
SDR	Software Defined Radio
USRP	Universal Software Radio Peripheral

TRIGONOMETRIC FORMULA

$$\cos(-x) = \cos(x)$$
$$\sin(-x) = -\sin(x)$$
$$\cos(A).\cos(B) = \frac{\cos(A-B) + \cos(A+B)}{2}$$
$$\sin(A).\sin(B) = \frac{\cos(A-B) - \cos(A+B)}{2}$$
$$\sin(A).\cos(B) = \frac{\sin(A+B) + \sin(A-B)}{2}$$
$$\cos(A).\sin(B) = \frac{\sin(B+A) + \sin(B-A)}{2}$$

NOMENCLATURE

$s_{RF}(t)$	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 in time domain after filtering and sampling
$r_I(k.T_e)$	imaginary part of the received signal in time domain after filtering and sampling
$s_a(t)$	analytic signal
$s(t)$	complex envelop
$S_R(t)$	Fourier transform of real part of the received signal
$S_I(t)$	Fourier transform of imaginary part of the received signal
$\widetilde{R_R}(t)$	Fourier transform of real part of the received signal, after filtering
$\widetilde{R_I}(t)$	Fourier transform of imaginary part of the received signal, after filtering

MAIN FOURIER TRANSFORMS

$s(t) \Rightarrow S(f)$

$1 \Rightarrow \delta(f)$

$\cos(2\pi f_0 t) \Rightarrow \frac{1}{2}.|\delta(f+f_0)+\delta(f-f_0)|$

$\sin(2\pi f_0 t) \Rightarrow \frac{j}{2}.|\delta(f+f_0)-\delta(f-f_0)|$

$multiply \Rightarrow convolve$

FIRST PART: IQ TRANSCEIVER

First, let's briefly define the "**Software Defined Radio**" notion. It is one type of radiofrequency transceiver, but its particularity is to have the most part of its processing done **digitally** – either in transmission or reception – which allows to develop a lot of different applications using the same hardware by changing only the software part. The possibility of updating the software without stopping the hardware is also a great advantage.

Indeed, with today technologies in our smartphones, the transmission channels are **independent** for each technology (GSM with 2G, UMTS with 3G, LTE with 4G, WIFI, Bluetooth, etc.). Therefore, the aim of SDR is to **mutualize** all these channels to be able to save space, energy and hardware replacement.

During these sessions, we used a National Instruments USRP-2900 SDR transceiver **[Fig. 1]**. As you can see on **[Fig. 2]** below, it is composed of **two parts**: one for the transposition of the signals around the zero frequency, and one to carry out their CAN. We will finally use the resulting flow of samples stored in the computer under **GNURadio**, a free software.

Fig. 1: National Instruments USRP-2900

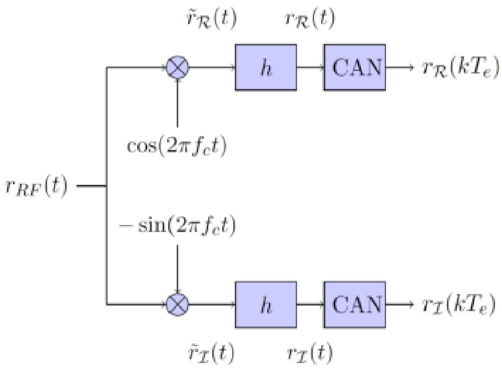

Fig. 2: Block diagram of the receiver

The objective is to demonstrate that this receiver will be able to reconstruct perfectly the in-phase and quadrature channels. Let's consider a communication signal transmitted around a carrier frequency f_0:

$$s_{RF}(t) = A(t).\cos\big(2\pi f_0 t + \varphi(t)\big) \qquad (1)$$

with $A(t)$ the envelope and $\varphi(t)$ the phase.

We can make here a small recap of the three common **modulation techniques** considering a message $m(t)$ to be transmitted:

- **Amplitude modulation** $A(t) \propto m(t)$ and $\varphi(t)$ constant
- **Phase modulation** $\varphi(t) \propto m(t)$ and $A(t)$ constant
- **Frequency modulation** $\varphi(t) \propto \int_{-\infty}^{u} m(u).du$ and $A(t)$ constant

We are also able to combine these three modulation types, with $A(t) \propto m(t)$ and $\varphi(t) \propto m(t)$.

If we note $s_R(t) = A(t).\cos(\varphi(t))$ and $s_I(t) = A(t).\sin(\varphi(t))$ the channels IQ, and supposing a bandwidth $\frac{B}{2} < f_0$, we re-write the previous expression (1) as follow:

$$s_{RF}(t) = s_R(t).\cos(2\pi f_0 t) - s_I(t).\sin(2\pi f_0 t) \qquad (2)$$

Here we can observe the Fourier representation of the signal $s_{RF}(t)$:

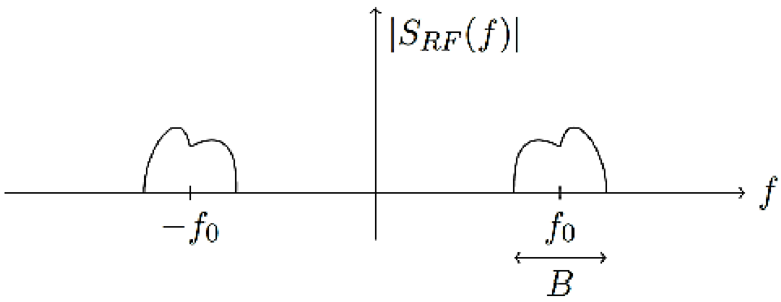

Fig. 3: Spectral representation of the signal $s_{RF}(t)$

According to Fig. 2, considering that the received signal is similar to the transmitted one and using trigonometric formulas, we can express the signal $\tilde{r}_R(t)$ and $\tilde{r}_I(t)$ in function of $s_{RF}(t)$, f_0 and f_c.

We have:

- $r_{RF}(t) = s_{RF}(t)$
- $s_{RF}(t) = s_R(t).\cos(2\pi f_0 t) - s_I(t).\sin(2\pi f_0 t)$

Replacing with above expressions, we can write:

$$\tilde{r}_R(t) = r_{RF}(t) \times \cos(2\pi f_C t) = s_R(t)\cos(2\pi f_0 t) \times \cos(2\pi f_C t) - s_I(t)\sin(2\pi f_0 t) \times \cos(2\pi f_C t)$$

$$\boxed{\begin{aligned}\tilde{r}_R(t) = &\frac{s_R(t)}{2}[\cos(2\pi(f_0 + f_C)t) + \cos(2\pi(f_0 - f_C)t)] \\ &-\frac{s_I(t)}{2}[\sin(2\pi(f_0 + f_C)t) + \sin(2\pi(f_0 - f_C)t)]\end{aligned}}$$

On the same way we demonstrate that:

$$\tilde{r}_I(t) = -r_{RF}(t) \times \sin(2\pi f_C t)$$

$$\tilde{r}_I(t) = -s_R(t)\cos(2\pi f_0 t) \times \sin(2\pi f_C t) + s_I(t)\sin(2\pi f_0 t) \times \sin(2\pi f_C t)$$

$$\boxed{\begin{aligned}\tilde{r}_I(t) = &\frac{s_I(t)}{2}[\cos(2\pi(f_0 - f_C)t) - \cos(2\pi(f_0 + f_C)t)] \\ &-\frac{s_R(t)}{2}[\sin(2\pi(f_0 + f_C)t) - \sin(2\pi(f_0 - f_C)t)]\end{aligned}}$$

If we take $f_0 = f_c$, we can evaluate the characteristics of the h filters to get $r_R(t) = s_R(t)$ and $r_I(t) = s_I(t)$.

We have:

$$\widetilde{r_R}(t) = \frac{s_R(t)}{2}[\cos(2 * 2\pi f_0 t) + 1] - \frac{s_I(t)}{2}[\sin(2 * 2\pi f_0 t) + 0]$$

If we apply the **Fourier transform** given in introduction in the main Fourier transform formulas, we obtain:

$$\widetilde{R_R}(f) = F(\widetilde{r_R}(t))$$
$$\widetilde{R_R}(f) = \frac{1}{2}\delta(f)\left(\frac{\widetilde{S_R}(f)}{2}\right) \circledast [\frac{1}{2}|\delta(f+2f_0) + \delta(f-2f_0)| + \delta(f)]$$
$$-\frac{1}{2}\delta(f)\left(\frac{\widetilde{S_I}(f)}{2}\right) \circledast [\frac{j}{2}|\delta(f+2f_0) - \delta(f-2f_0)|]$$

$$\boxed{\widetilde{R_R}(f) = \frac{1}{4}[2S_R(f) + S_R(f+2f_0) + S_R(f-2f_0) - jS_I(f+2f_0) + jS_I(f-2f_0)]}$$

The figure below represents the spectrum of this signal:

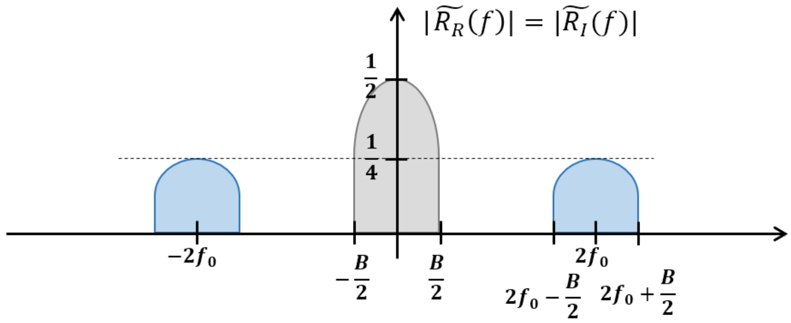

Fig. 4: Spectral representation of $|\widetilde{R_R}(f)| = |\widetilde{R_I}(f)|$

Therefore, in order to have $r_R(t) = s_R(t)$ ad $r_I(t) = s_I(t)$, we will choose a h **low-pass filter** with a cut frequency between $\frac{B}{2}$ and $2f_0 - \frac{B}{2}$ and a gain equal to 2:

$$|H(f)| = 2 \ for \ |f| \leq \frac{B}{2}$$
$$|H(f)| = 0 \ for \ |f| > \frac{B}{2}$$
$$R_R(f) = \widetilde{R_R}(f) \circledast H(f) = \frac{1}{4} \circledast [2S_R(f)] \times 2 = S_R(f)$$

> We can discuss now about the receiver presented in Fig. 1 and its capability to work with wide-band signals.

If we have a look at the previous figure, in the case of a bandwidth $\frac{B}{2} > f_0 \leftrightarrow B > 2f_0$, a **spectrum covering** will occur between the centred and translated ones, and so our filter action will not work properly. Therefore, we have:

$$\boxed{B < 2f_0}$$

Thus, it will **not work** for wide-band signals.

We now want to know how we must choose the sampling period T_e in order to recover $r_R(t)$, $t \in \mathbb{R}$ from $r_R(k.T_e)$, $k \in \mathbb{Z}$

We have to respect the **Shannon-Nyquist** famous criteria:

$$f_e > 2f_{max} \Leftrightarrow f_e > \frac{2B}{2}$$

$$\boxed{T_e < \frac{1}{B}}$$

We will be able in this case de recover all the frequencies.

We can interrogate us about the possibility of interchanging the stages of frequency transposition and analog to digital conversion. Would it be possible?

If we do this operation, we can observe that the sampling frequency F_e will have to be very high, and so it would be very difficult and expensive to respect the Shannon criteria. Therefore, our model is well design as it is.

So, we have demonstrated that two real signals $s_R(t)$ and $s_I(t)$ can be transmitted on the carrier frequency f_0 and perfectly recovered thanks to IQ receiver system.

We will now define two equivalent models for $s_{RF}(t)$ based on **symmetrical Hermitian Fourier** transform principle. For narrowband signals **[Fig. 5]**, the representation in positive (or negative) frequencies is sufficient, and so it is possible to hide the value of f_0.

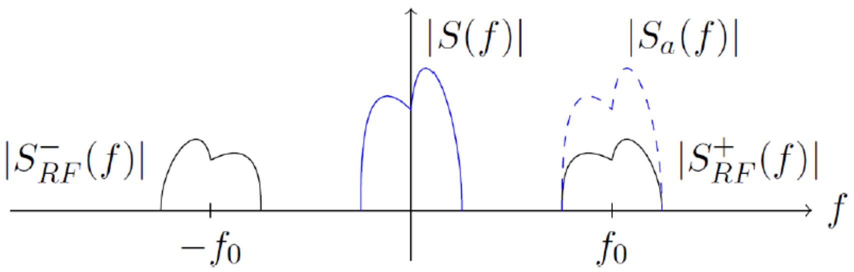

Fig. 5: Spectral representation of a real narrow-band signal S_{RF}, *its analytic signal* S_a *and its complex envelope* $S(f)$

To conserve all its power, the positive **analytic signal** associated to the real narrow-band signal needs to have an amplitude that is **twice** as important. The **complex envelope** is obtained after re-centring the analytic signal on the Y axis.

Supposing a real narrow-band signal $s_{RF}(t) = A(t).\cos(2.\pi.f_0.t + \varphi(t))$, $t \in \mathbb{R}$, we can express its analytic signal and its complex envelop in function of f_0, knowing that $S_{RF}(f) = S_{RF}^{*}(-f)$.

Remembering that we get:

- $s_{RF}(t) = s_R(t).\cos(2\pi f_0 t) - s_I(t).\sin(2\pi f_0 t)$

We can express $S_{RF}(f)$, find its analytic signal S_a and finally its complex envelop $S(f)$.

With $S_{RF}(f) = S_{RF}{}^*(-f)$ equality, we have:

$$S_{RF}(f) = \frac{1}{2}[S_R(f - f_0) + S_R(f + f_0) + jS_I(f - f_0) - jS_I(f + f_0)]$$

To obtain the analytic signal, we **only keep the positive frequencies** $(f - f_0)$, and we multiply by two:

$$S_a(f) = S_{RF}(f) + j[-j \times sgn(f) \times S_{RF}(f)] = 2 \times S_{RF}(f)$$

$$S_a(f) = S_R(f - f_0) + jS_I(f - f_0) = [S_R(f) + jS_I(f)] \circledast \delta(f - f_0)$$

$$\boxed{s_a(t) = [s_R(t) + js_I(t)]e^{-j2\pi f_0 t}}$$

And finally:

$$S(f) = S_a(f + f_0) = S_R(f) + jS_I(f)$$

$$\boxed{s(t) = s_R(t) + js_I(t)}$$

Next, we will implement some receivers thanks to **GNURadio** SDR software.

SECOND PART: RECEPTION OF FM BROADCASTING

After having studied theoretical elements, we have to demodulate FM broadcasting by implementing a receiver with **GNURadio**. For this, we use a recorded file (obtained thanks to the acquisition system introduced in the first part).

This signal is part of the **Very High Frequency band** (VHF) which extends from 30MHz to 300MHz, here in a sub-band between **87.5MHz and 108MHz**. It uses a center frequency of $f_c = \mathbf{99.5}\boldsymbol{MHz}$ and a sampling frequency of $\boldsymbol{F_e} = \mathbf{1.5}\boldsymbol{MHz}$. Theatrically, each different channel is separated with at least 100kHz giving us 203 simultaneous broadcasting stations. We will now try to restore the audio content of the recording.

1. FREQUENCY ANALYSIS OF THE RECORDING

We first use the **GRC tool** (similar to Simulink), implementing a chain:

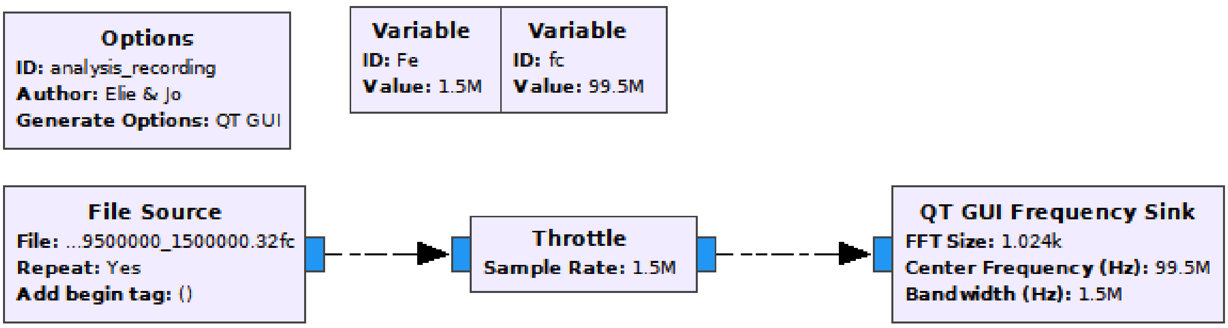

Fig. 6: Frequency analysis processing chain on GRC

| We use the online manual to present the role of each block used in the processing chain.

Here are the different roles of the blocks used in **[Fig. 6]**:

- *File source*: reading the file (but do not handle how to interpret it)
- *Throttle*: to give the sampling frequency of the source file
- *Variable*: to use variables (we have to think to use them)
- *QT GUI Frequency Sink*: a graphical window to display frequency representation
- f_c: center frequency (frequency of transposition)
- F_e: sampling frequency

| We choose the missing variables as we can see on **[Fig. 6]**:

- F_e: 1.5MHz
- f_c: 99.5MHz
- *Sample rate*: F_e $(1.5MHz)$
- *Center Frequency*: f_c $(99.5MHz)$
- *Bandwidth*: same as F_e $(1.5MHz)$ because we have already used the Shannon criteria

We observe several frequency channels and we use the allocation of frequencies in the FM band near Toulouse to know the stations that we are observing. (https://www.annuradio.fr/index.php?mode=searchville&choixville=TOULOUSE).

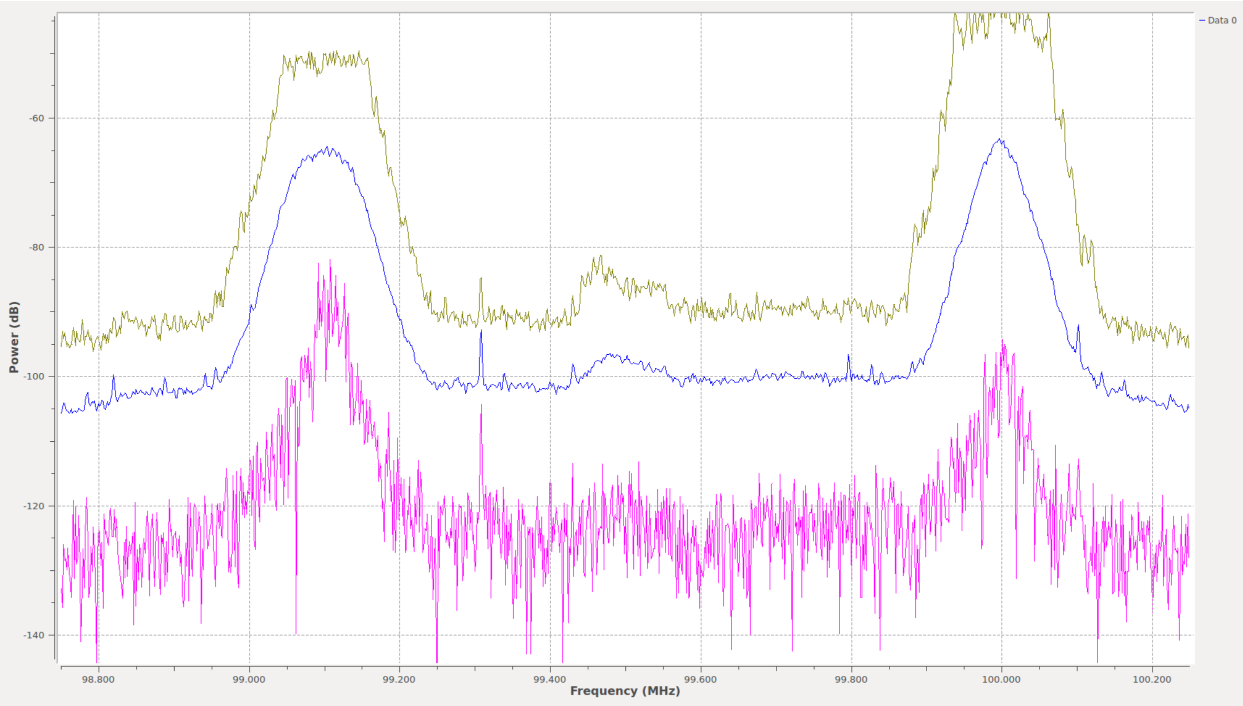

Fig. 7: FFT of the recorded signal

We can see two main frequencies on the **[Fig. 7]**: $\mathbf{99.1}\boldsymbol{MHz}$ and $\mathbf{100}\boldsymbol{MHz}$ which correspond, respectively, to **RFM** and **Skyrock** radios.

In the directory we can see that we should have a frequency for Nostalgie radio. In fact, we can see this peak (but very small) at $99.5MHz$, by holding the maximum.

Now, we will measure the signal-to-noise ratio (SNR) in decibel to see if it is enough to demodulate the signal.

To measure the **SNR**, it is better to use the **average** (blue line). We take it at $-\mathbf{3}\boldsymbol{dB}$ to try to not lose any signal (useful signal).

On **[Fig. 7]**, the maximum is $-65.8dB$, then we use $-68.8dB$. The minimum is $-103db$.

$$SNR(dB) = 103 - 68.8 = \mathbf{34.2}\boldsymbol{dB}$$

This value seems to be enough to manage to demodulate the signal, but the measure is very uncertain, so it is impossible to make a certain conclusion for the SNR.

To continue with transposition, filtering and demodulation, we have to approximate the bandwidth of a channel.

To have a great idea of the **bandwidth**, it is better to be at **mid-height** of each frequency peak. Then, at mid-height, the bandwidth of each radio is:

- RFM: $B = 99.184 - 99.027 = \mathbf{157}\boldsymbol{kHz}$
- Skyrock: $B = 100.064 - 99.934 = \mathbf{130}\boldsymbol{kHz}$

These two bandwidths are a little too small, we can use **250kHz**.

2. Channel extraction by frequency transposition and low-pass filtering

Now, we know that there are 3 different stations and we want to receive each one **separately**. We will first transpose the frequency to center the useful signal and then apply a **low-pass filter** to attenuate the noise (out of the band).

Noting $r[k]$ the complex envelope (regarding f_c) of the recorded signal, sampled at the frequency F_e, the frequency transposition for a quantity f_l, $l \in \{1; \dots; L\}$ can be wrote:

$$r_l[k] \; = r[k].e^{-j.2.\pi.\frac{f_l}{F_e}.k}$$

To do this frequency transposition through GRC, we use the next blocks:

- *Signal Source*: to generate the complex exponential, to transpose the signal
- *QT GUI Range*: to define dynamically the transposition frequency f_l during the software execution (choosing the step with 50 or 100kHz)
- *Multiply* to make the product

We use this new chain:

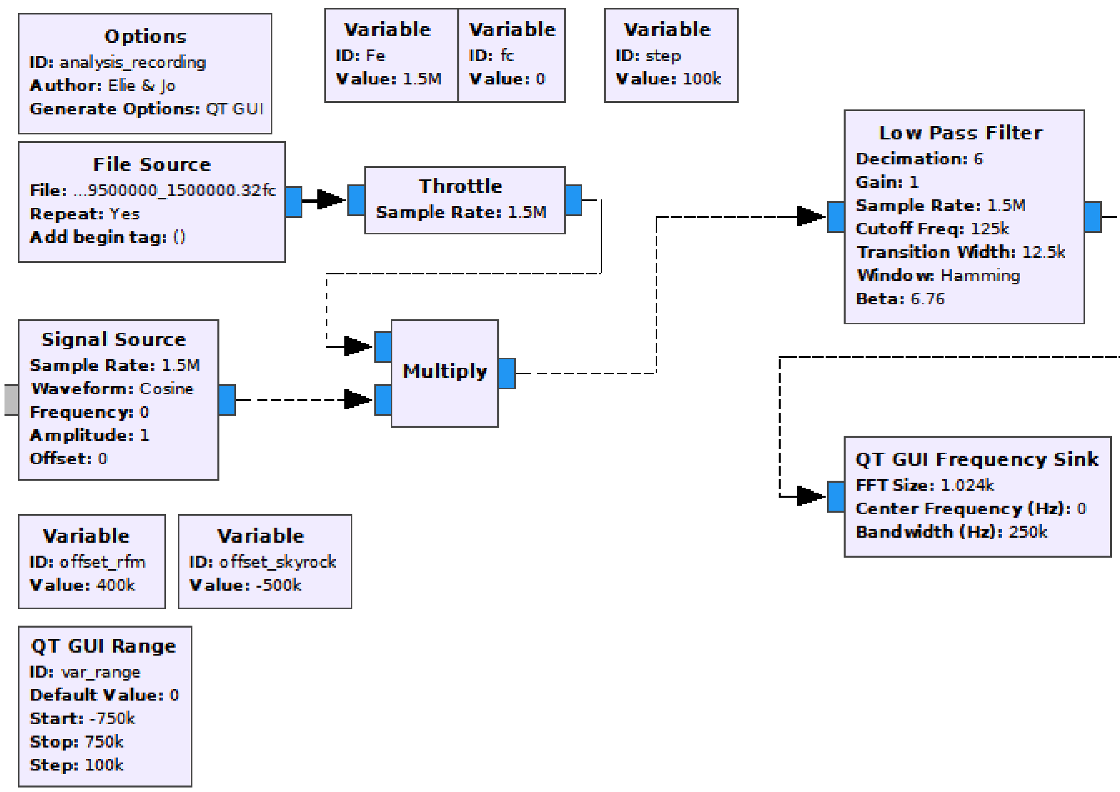

Fig. 8: Channel extraction chain on GRC

To be able to center each channel, we have to define the frequency offsets needed.

We want to transpose the signal because we have samples in **low frequency** (already useful signal) so we want to refocus the different channels on a zero frequency to facilitate the processing.

Regarding the range block, we only can move between $-\frac{F_e}{2}$ **and** $+\frac{F_e}{2}$ (respectively start and stop parameters, $-750Hz$ **and** $+750Hz$**).**

We can use the QT GUI Frequency Sink to check that the frequency transposition is effective (centering the visualization on the null frequency).

After launching the software, the frequency offsets needed to center each channel are:

- RFM: $\mathbf{400kHz}$ $(99.5 - 99.1 = 0.4MHz)$
- Skyrock: $\mathbf{-500kHz}$ $(99.5 - 100 = -0.5MHz)$

If the frequency offset is **higher** than the sampling frequency F_e, we **repeat** the spectrum, thanks to the periodicity of the complex exponential.

We define the low-pass filter parameters, and those of the frequency analyser.

- Cut-off frequency: $\frac{Bandwidth}{2} = \frac{250}{2} = 125kHz$
- Sample rate: F_e $(1.5MHz)$
- *Trans width*: **10%** of the cut-off frequency
- *Frequency analyser*: Center frequency = $0Hz$

Decimation: to keep only a fixed number of samples (example with 6: we keep 1 sample on 6 samples). It allows to respect the **Shannon** criteria and to use simple frequencies.

3. Frequency demodulation and restitution

We will now try to listen the radio station by using a **sound card**. We have to demodulate the signal (beginning by understanding the modulation method, using two stereophonic channels). The two channels are centered in frequency and have a maximum frequency of **15kHz**.

They are multiplexed to form the message:

$$m(t) = g(t) + d(t) + A_{sp}.\cos\left(2.\pi.f_{sp}.t\right) + [g(t) - d(t)].\cos(2.\pi.2.f_{sp}.t)$$

with $f_{sp} = 19kHz$ and amplitude $A_{sp} = 2$

The monophonic receiver consists of implementing a **low-pass filter at $15kHz$ [Fig. 9]**. The **stereophonic** receiver is required to amplitude demodulate the signal around $38kHz$ and recombine it with the baseband signal to reconstruct the left and right channels.

The composite signal $m(t)$ is frequency modulated and the radiofrequency signal centered on f_0 at the output of the transmitter is noted:

$$s_{RF}(t) = A.\cos\left(2.\pi.f_0.t + \frac{\Delta f}{max(|m(t)|)}.\int_{-\infty}^{t} m(u).du\right)$$

Δf is the maximum frequency excursion of the modulation (fixed at $75kHz$).

We show that the frequency-modulated signal is on an **infinite band**, but decreases rapidly, so that it can be approached via the **Carson** rule:

$B_{FM} \approx 2.(\Delta f + f_m)$ with f_m the maximum frequency of the composite signal $m(t)$.

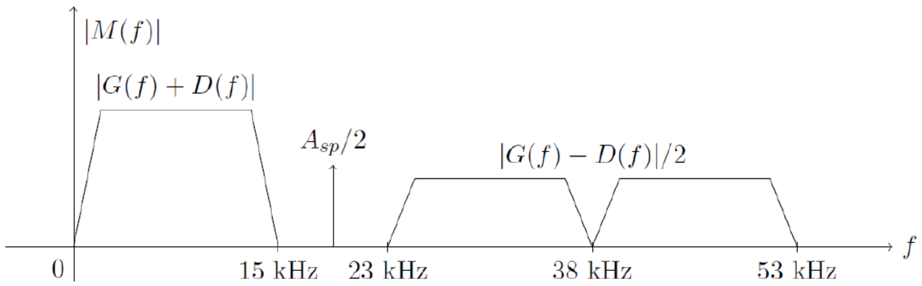

Fig. 9: Stereophonic composite signal before the frequency modulation

> We use the Carson rule to check that the bandwidth of the channel measured in the previous part confirms the theory.

We have $\boldsymbol{f_m = 53kHz}$ and we have $\boldsymbol{\Delta f = 75kHz}$.

Then $\boldsymbol{B_{FM} = 256kHz}$ in theory. This is consistent with the channel measured in the previous part ($250kHz$).

> From the expression of the transmitted signal and the affected processes until now (frequency transposition and low-pass filtering) we want to show that the signals $y_l[k]$ can be noted:
>
> $y_l[k] = A.e^{j.k_f\sum_{i=0}^{k} m(i)} + b[k]$ *(5)*
>
> We have to define the value of k_f, with $b[k]$ a complex noise term introduced by the propagation channel as well as by the transceiver itself.

We have: $S_{RF}(t) = A.\cos\left[2\pi.f_0.t + \frac{\Delta f}{\max(|m(t)|)}\int_{-\infty}^{t} m(u).du\right]$

$$S_{RF}(t) = S_R(t).\cos(2\pi.f_0.t) - S_I(t).\sin(2\pi.f_0.t)$$

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

$$S_{RF}(t) = A(t).\cos(\varphi(t)).\cos(2\pi.f_0.t) - A(t).\sin(\varphi(t)).\sin(2\pi.f_0.t)$$

Thanks to the first part, we know that:

$$S(t) = S_R(t) + j.S_I(t) = A(t).\cos(\varphi(t)) + j.A(t).\sin(\varphi(t))$$

$$S(t) = A(t).e^{j.\varphi(t)}$$

$$S(t) = A(t).e^{j.\frac{\Delta f}{\max(|m(t)|)}\int_{-\infty}^{t} m(u).du}$$

We put, to discretize: $$t = k.T'_e$$

$$u = i.T'_e$$

$$S[K] = A[k].e^{j.\frac{\Delta f}{\max(|m(t)|)}.\sum_{i=0}^{k} m(i)} + b[k]$$

$$A[k] = A$$

Then: $y_l[k] = A.e^{j.k_f \sum_{i=0}^{k} m(i)} + b[k]$ with $\boldsymbol{k_f} = \frac{\Delta \boldsymbol{f}}{\max(|\boldsymbol{m(t)}|)}$

From this formula, we can show that frequency demodulation can be performed numerically in the following way: $\widetilde{m_l}[k] = arg(y_l[k].y_l^*[k-1])$

with * the conjugate operator.

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

- *WBFM Receive*: to process the frequency demodulation as defined by (6)
- *Low Pass Filter*: to conserve only the monophonic signal
- *Audio sink*: to play the audio stream.
- *Rational Resampler*: between the Low Pass Filter and the Audio Sink in order to adapt the sample rates with these available on the sound card

We use GRC to plot the spectrum of the demodulated channel:

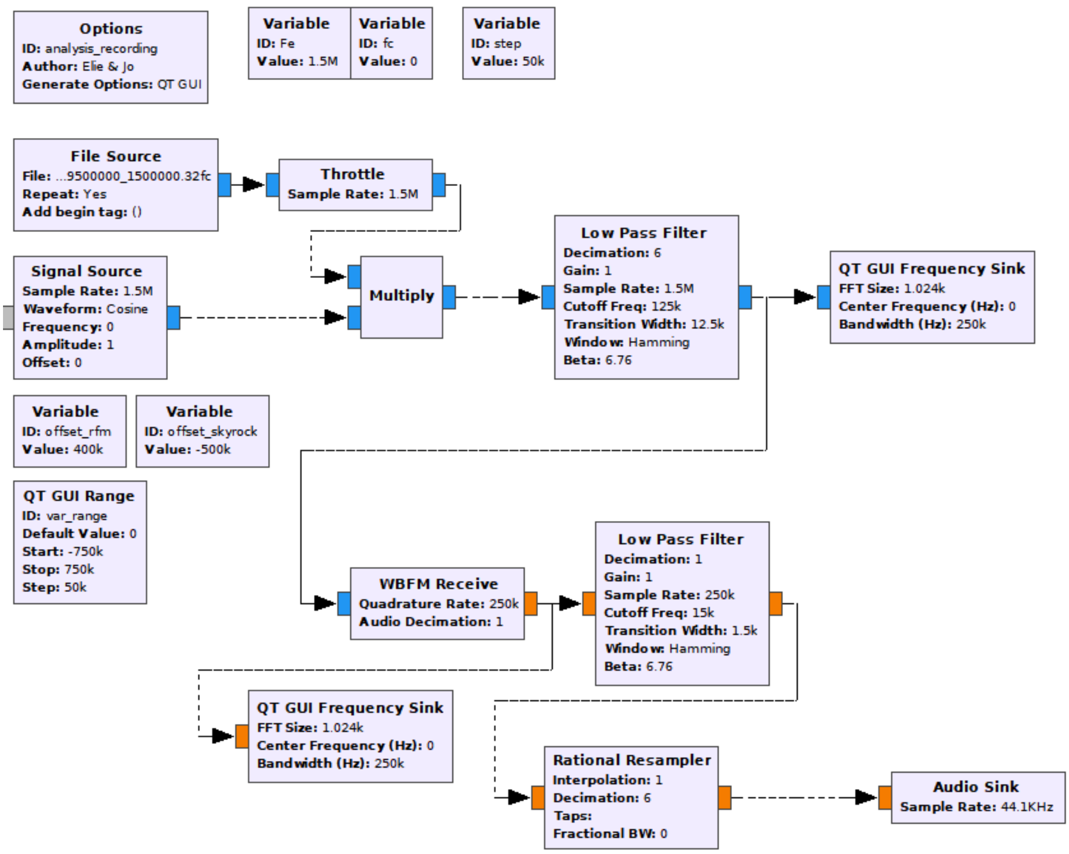

Fig. 10: Demodulation chain on GRC

Below are the 2 demodulated signals:

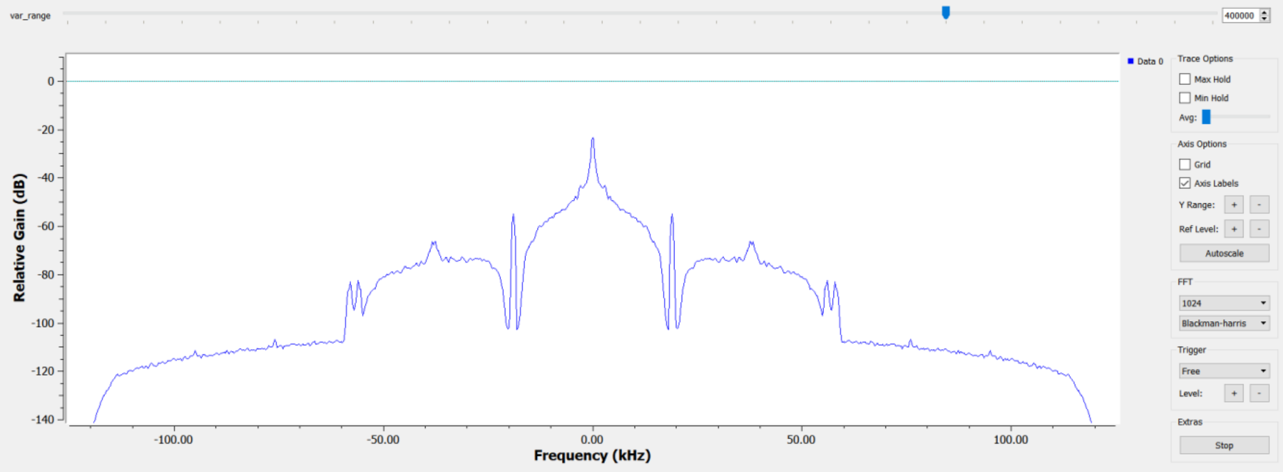

Fig. 11: Demodulated signal from RFM

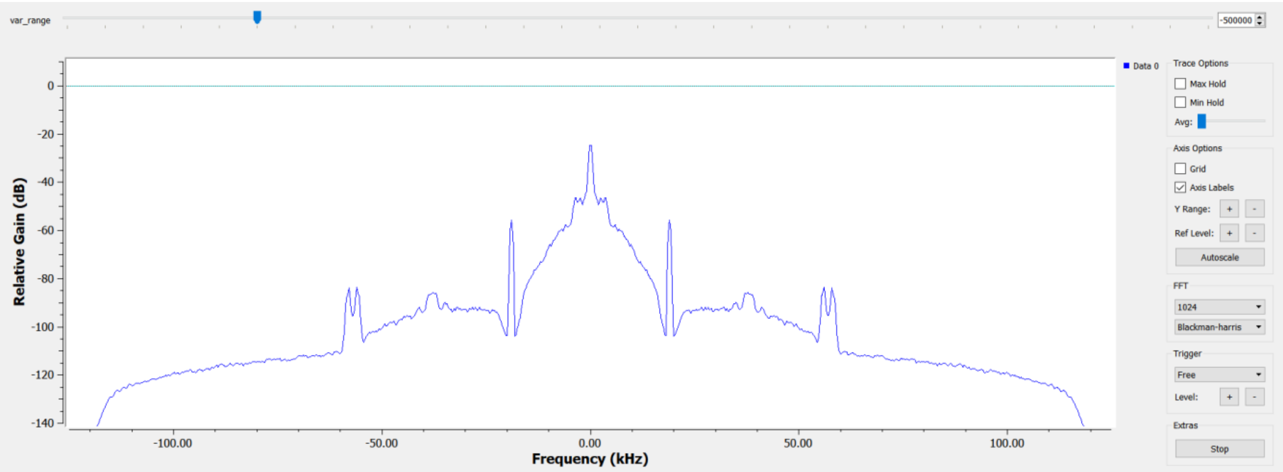

Fig. 12: Demodulated signal from Skyrock

> We can now listen the channels with the Audio Sink block.

Listening to RFM we learn that the winner of a Sam Smith album was **Jordy**. Also, we can hear the song **Counting Stars** (from One Republic) on Skyrock.

Finally, we try to listen to Nostalgie and we manage to hear a song (but with a lot of noise), YMCA from Village People.

4. Real time implementation with an USRP receiver

After having successfully listen the given sample, we installed an SDR transceiver to receive a signal in real time and to listen **the live radio**.

We just had to replace the source block by the **UHD USRP Source block** to interface GRC with the driver of the transceiver.

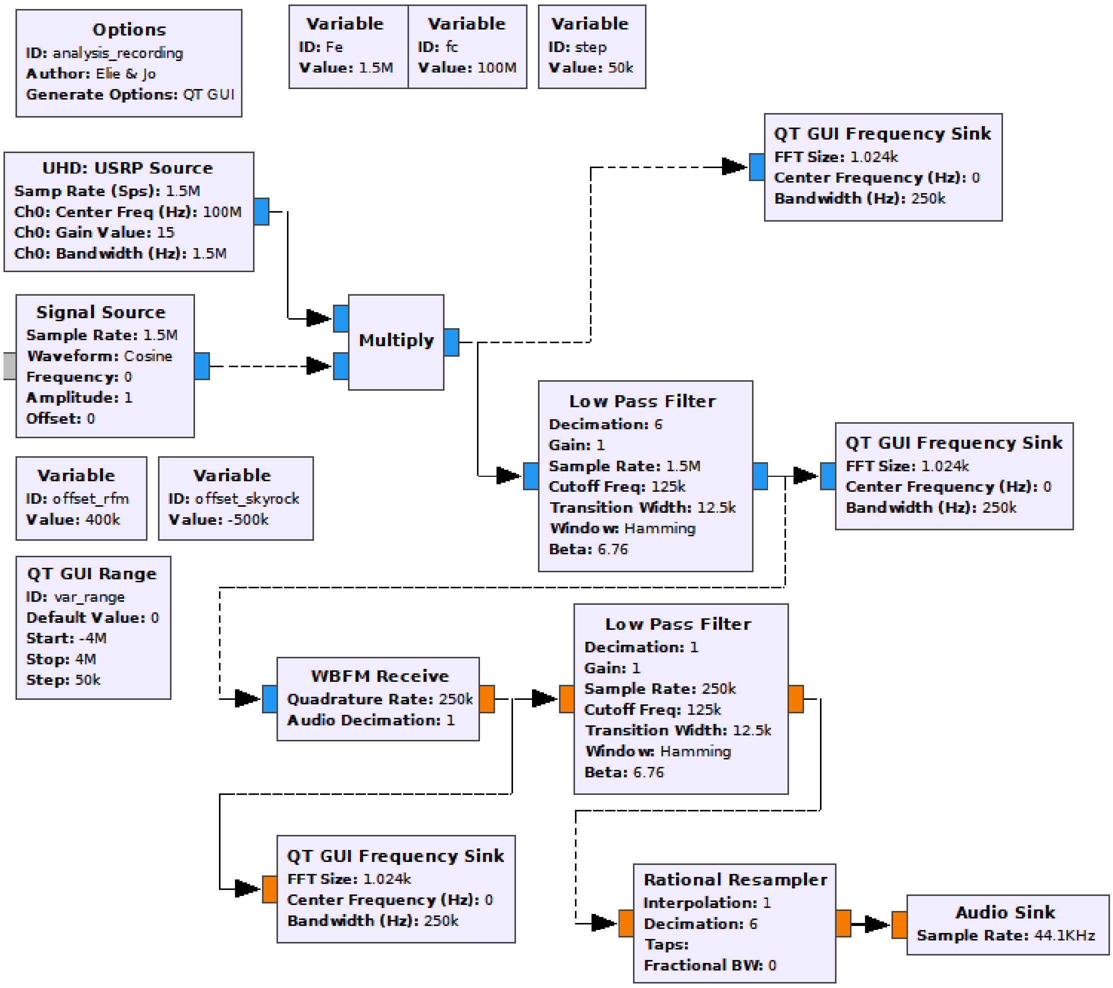

Fig. 13: Demodulation chain on GRC for real time signal

We finally managed to listen Skyrock, our configuration was working **very well**.

THIRD PART: RECEPTION OF VOLMET MESSAGES IN AM-SSB

Now, we will study the frequency sub-band between **11.175MHz and 11.4MHz** reserved to the **international aeronautic communications** (VOL METEO service, a meteorological information signal). This sub-band is part of the frequency band named **High Frequency** (HF), which ranges from **3MHz to 30MHz**. All the signal propagation is done there by successive reflections on the ionosphere and allow intercontinental links with a reasonable power budget.

To analyse this sub-band, we used a recorded file (obtained thanks to the acquisition system introduced in the first part). It used a center frequency of $\boldsymbol{f_c = 11.2965MHz}$ and a sampling frequency of $\boldsymbol{F_e = 250kHz}$. We will now try to restore the phonic content of the recording.

1. FREQUENCY ANALYSIS OF THE RECORDING

First, we are going to plot the modulus of the discrete Fourier transform in decibels, between $\mathrm{f_0} - \frac{\mathrm{F_e}}{2}$ and $\mathrm{f_0} + \frac{\mathrm{F_e}}{2}$, by using QT GUI Frequency Sink block, and second, check then with http://www.dxinfocentre.com/volmet.htm/ that the VOLMET station located in the Royal Air Force air-base of St-Eval, United Kingdom, is well observed at the expected frequency.

We use a simple chain to observe the modulus of the FFT:

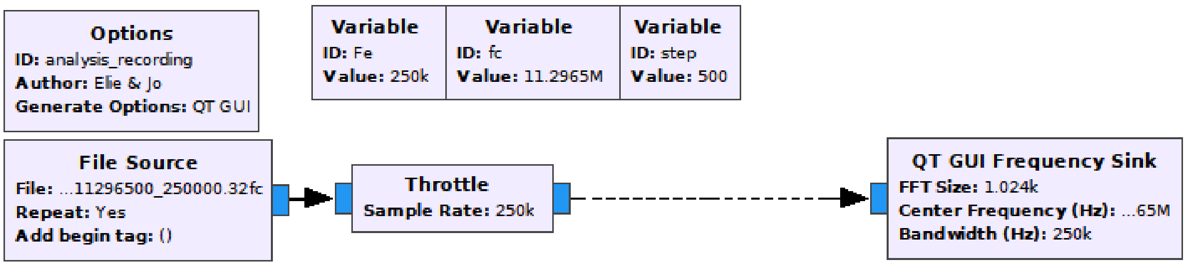

Fig. 14: FFT chain on GRC

Here is the figure we obtain running this chain:

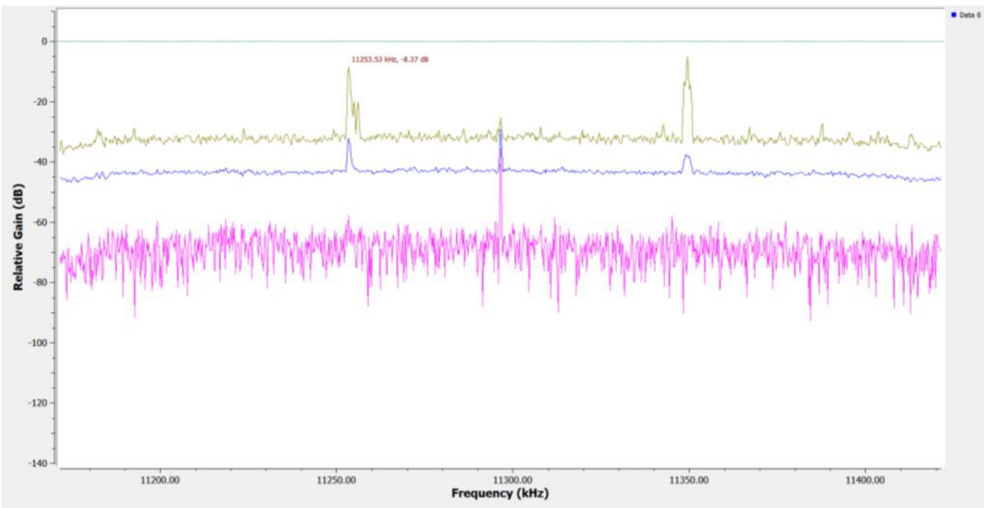

Fig. 15: FFT of the signal

We can observe on the figure above **[Fig. 15]** a power peak very precisely at **11 253.53 kHz** (and another peak at the center corresponding to f_c). When we go on the referred website, we see clearly that this radio frequency is the in fact the good one, dedicated to the military base of St Eval **[Fig. 16]**.

11 MHz									
11.247	, 35	MTS	FLK	MOUNT PLEASANT (VIPER)		4000	-51 50 14	-58 28 14	IRREGULAR HOURS
11.253	Cont	MKL	GBR	ST EVAL (MILITARY ONE)		4000	50 28 58	-5 00 00	
11.297	25 , 55	RLAP	RUS	ROSTOV	RR	2000	47 15 12	39 49 02	DAY
11.318	00 , 30	UBB-2	RUS	SIVKAR	RR	2000	61 38 17	50 31 49	DAY
	10 , 40	UNNN	RUS	NOVOSIBIRSK	RR	2000	55 00 16	82 33 44	DAY
	15 , 45	RQCI	RUS	SAMARA	RR	2000	53 11 00	49 46 00	DAY
11.369	01 ,	LWB	ARG	EZEIZA	SS	1000	-34 49 59	-58 31 55	
11.387	00 , 30	VKA-931	AUS	AUSTRALIAN		1000	-23 47 47	133 52 28	EX: AXQ-421 -27 04 06 153 03 17
	05 , 35	AWC	IND	KOLKATA		800	22 38 00	88 27 00	0305 - 1240 Z
	10 , 40	HSD	THA	BANGKOK		1000	13 44 00	100 30 00	2310 - 1145 Z
	15 , 45	ARA	PAK	KARACHI		5000	25 54 00	67 09 00	0145 - 1450 Z
	20 , 50	9VA-43	SNG	SINGAPORE		6000	1 20 11	103 41 10	2250 - 1225 Z
	25 , 55	AWB	IND	MUMBAI		800	19 05 15	72 51 09	0325 - 1300 Z

Fig. 16: Frequencies of VOLMET broadcast

2. Frequency transposition

To do the frequency transposition, we use the same method as the second part with the same blocks:

- *Signal Source* to generate the complex exponential
- *QT GUI Range* to define dynamically fl during the software execution
- *Multiply* to make the product

We first need to determine the frequency offset f_1 needed to center the channel with the maximum power at the null frequency. We can plot the modulus of the discrete Fourier transform in decibels, between $f_0 - \frac{F_e}{2}$ and $f_0 + \frac{F_e}{2}$, by using QT GUI Frequency Sink block.

As we can observe on **[Fig. 17]**, we need **43kHz** of offset in order to get the maximum peak, by moving our range variable.

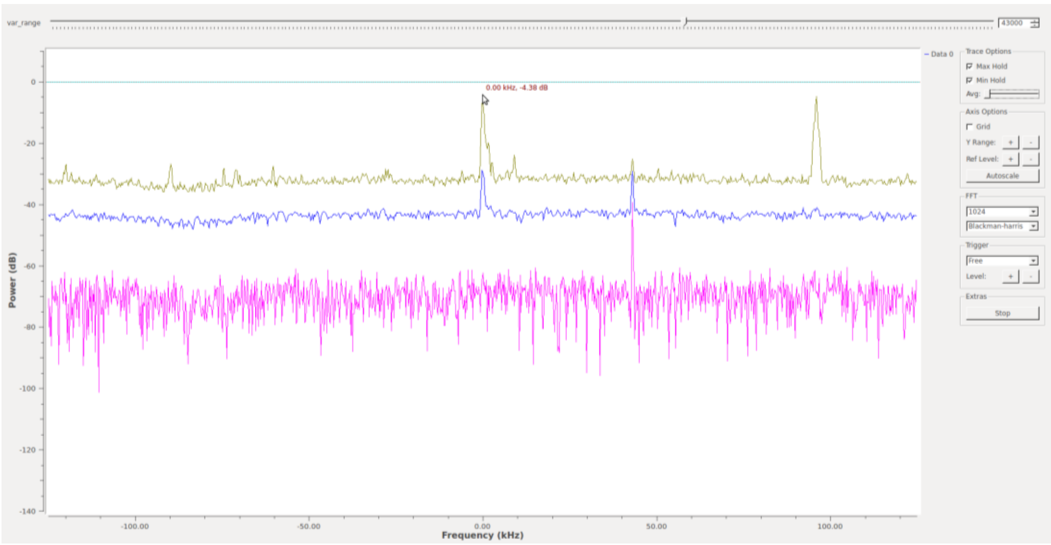

Fig. 17: Signal centered in f_0 and translated of 43kHz

3. SINGLE SIDEBAND AMPLITUDE DEMODULATION

The **VOLMET** service is a single sideband amplitude demodulation.

If we note $m(t)$ the vocal message to be transmitted, we need a low-pass filter in order to occupy a bilateral sideband as $B = 6kHz$, as presented below on **[Fig. 18]**.

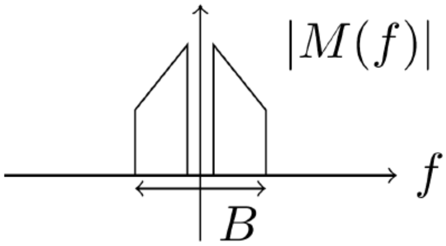

Fig. 18: Spectral representation of the amplitude of the real signal to be transmitted

This signal $m(t)$ can be used to modulate the amplitude of a sinusoid of frequency f_0 generally **much higher than B**. This is called amplitude modulation (**AM**) and the **real signal** is noted:

$$s_{RF}(t) = R\{s(t).e^{j.2.\pi.f_0.t}\}$$

with the **complex envelope** relative to f_0 defined as:

$$s(t) = m(t) \pm j.\mathcal{H}\{m(t)\}$$

> We need to plot $|S_{RF}(f)|$ in function of the polarity of the second term in s(t). We will then conclude about the denominations "single lower or upper sideband amplitude modulation".

We have $S_{RF}(f) = \mathcal{R}\{n(H) \pm j\mathcal{H}(n(H))e^{j2\pi f_0 t}\}$

With a previous demonstration in the first part, we know that $S_a(f) = S_{RF}(f) + sgn(f).S_{RF}(f)$

$$S_a(f) = [m(f) + sgn(f).S_{QR}(f)] * \delta(f - f_0) = m(f - f_0) + sgn(f - f_0).m(f - f_0)$$

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

First, with $\boldsymbol{f > f_0}$, we have:

$$S_{RF}(f) = \frac{1}{2}[m(f - f_0) + sgn(f - f_0).m(f - f_0) + m(-f - f_0) + sgn(-f - f_0).m(-f - f_0)]$$

And with $\boldsymbol{f < f_0}$:

$$S_{RF}(f) = \frac{1}{2}[m(f - f_0) + sgn(f - f_0).m(f - f_0) + m(-f - f_0) + sgn(-f + f_0).m(-f - f_0)]$$

We measured this result with the same idea taking in account the – sign at the beginning.

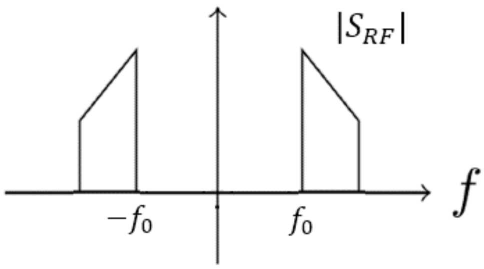

Fig. 19: Spectral representation of the amplitude of the modulated signal ($f > f_0$)

To do the link with the denominations "single lower or upper sideband amplitude modulation", we can say that, once the useful signal is **centred at the zero frequency**, we get a unique superior lateral band ($f > f_0$) and a unique inferior lateral band ($f < f_0$).

Using the spectrums plotted previously **[Fig. 15 and 17]**, we can find the sideband conserved for a VOLMET transmission:

The conserved sideband is the **upper band** (= 6kHz) **[Fig. 19]**. We can see on the figures that with the zero frequency at the center, there is the upper band 3kHz on the right side.

Considering the expression of a single upper sideband AM signal, we can associate the next model to our frequency centred signal:

$$r_1[k] = m[k] + j.\mathcal{H}\{m[k]\} + b[k]$$

with $b[k]$ a complex noise term affecting all the frequencies captured during the recording.

We note $y[k]$ the result of the filtered signal $r_1[k]$, so that we can delete the contribution of the noise outside of the band of interest, meaning $\left[0; \frac{B}{2}\right]$.

Using the tool Filter Design Tool, we are able to create a complex pass-band filter with the characteristics:

- complex pass-band filter with finite impulse response
- lower cut-off frequency at 0Hz
- upper cut-off frequency at $\frac{\text{B}}{2}$
- in-band gain of 0dB
- out-band gain of -30dB

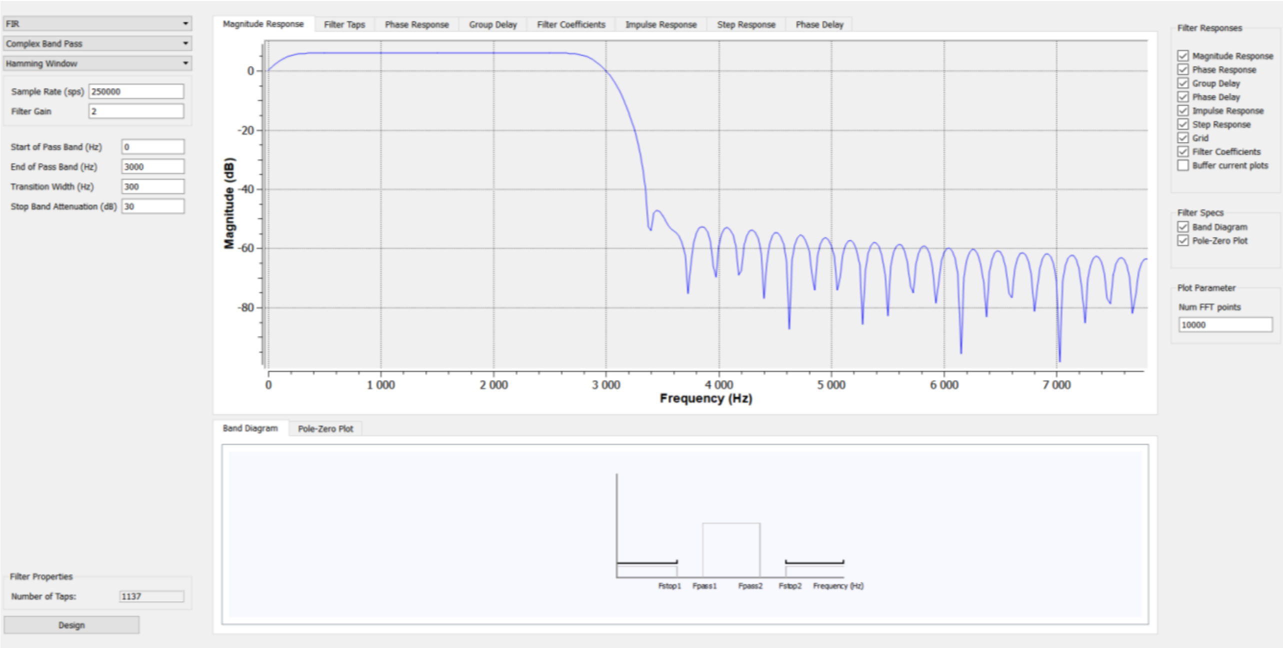

Fig. 20: Filter Design Tool

We can see in the above screenshot **[Fig. 20]** that we have a well a cut frequency equals to $\mathbf{3kHz} = \frac{B}{2}$.

> Now, we have to filter $\mathrm{r}_1[\mathrm{k}]$ thanks to the previous defined pass-band filter in order to get $y[k]$. We can copy the generated filter coefficients to use these in an FFT Filter block.

We obtain the spectrum representation below:

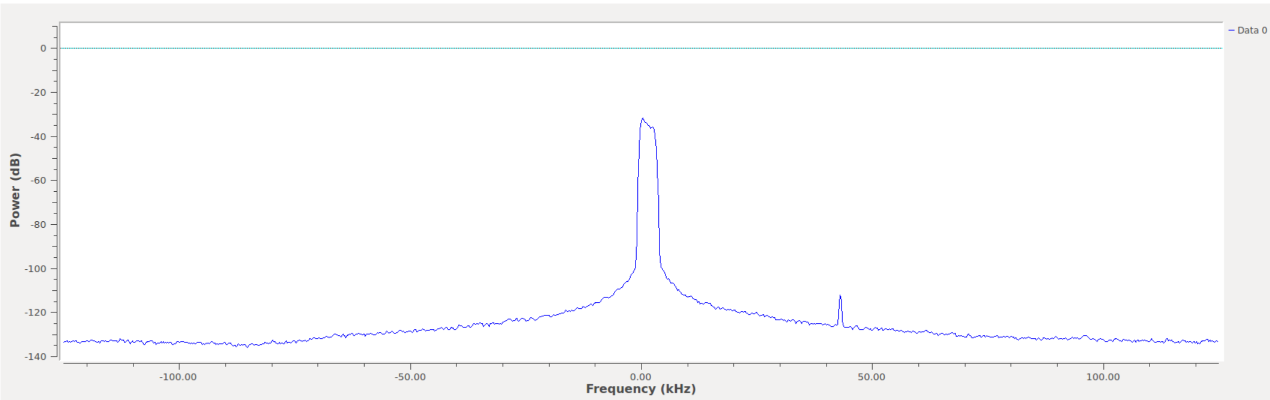

Fig. 21: Pass-band filter to get y[k]

> We finally implement the processes to recover a real signal in the form: $\widetilde{m}[k] = m[k] + \tilde{b}[k]$

We have $r_1[K] = m[K] + j\mathcal{H}\{m[K]\} + b[K]$, and we want under the following form $\widetilde{m}[K] = m[K] + b[K]$.

We get the following result **[Fig. 23]** under the software:

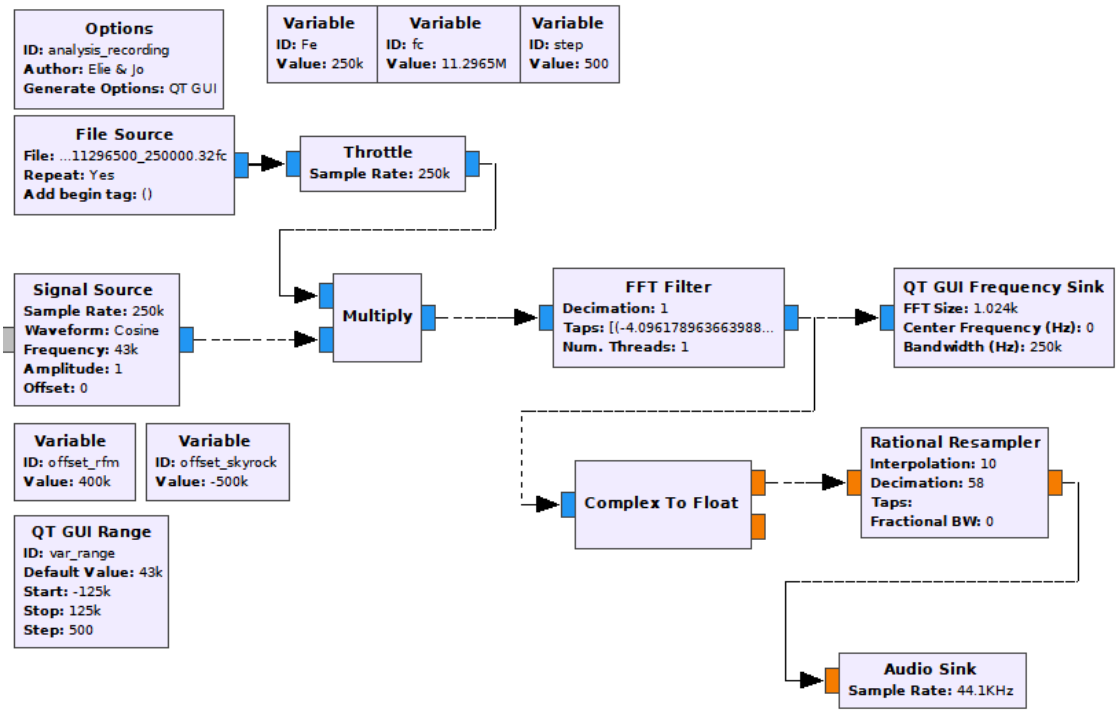

Fig. 22: Signal processing chain on GRC

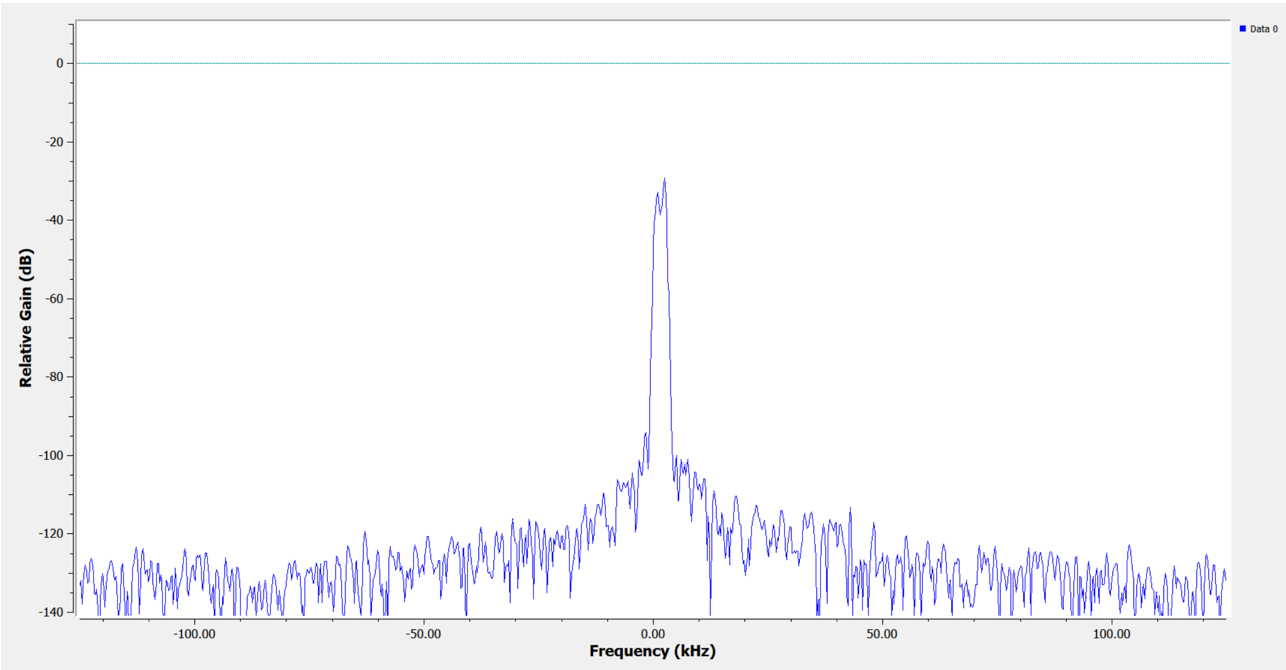

Fig. 23: Real signal demodulated

CONCLUSION

To conclude this report, we can say that these several sessions were very interesting in terms of first, understanding **theorical concepts** and then, applying them with real devices and **applications**.

The notion of “**Software-Defined Radio**” was unknow to us before these practical works, and we have discovered that is really something useful and which need to be used more and more in the future, particularly with the possibility of **updating software without changing the hardware**. In fact, we just had recently a conference about “Energy efficient future wireless communications” by Nuno Borges Carvalho, where SDR had a place we could not miss.

Finally, even if the theorical parts were **complicated** for some parts with new concepts, the practical part was much “funnier” with at the end the possibility to **hear radio** or specific channels we had no idea it existed.