Software-Defined Radio

Lucie Bechtet, Vincent Erb

December 2019

1 Introduction

This report goes over our study of Software-defined radios, especially the software processsing of a signal coming from a **USRP**. A USRP, or Universal Software Radio Peripheral, is a radio receiver that moves the signal processing operations, like filtering, amplifying, demodulating, from the hardware to the software. During the three labs, we first looked at the theoretical aspects of the processing done by the USRP to an incoming signal, before trying to exploit samples coming from music radios an military plane transmissions using **gnuradio**.

1.1 Question 1

We have a complex signal coming in, and the objective is to separate the Real part and the Imaginary part. Assuming that the received signal is similar to the emitted signal, we have :

$$r_{RF}(t) = s_{RF}(t)$$

$$\tilde{r}_R(t) = [s_R * cos(2\pi f_0 t) - s_I(t) * sin(2\pi f_0 t)]cos((2\pi f_0 t)$$

$$\tilde{r}_I(t) = -[s_R * cos(2\pi f_0 t) - s_I(t) * sin(2\pi f_0 t)]sin((2\pi f_0 t)$$

$$\tilde{r}_R(t) = \frac{s_R(t)}{2}[cos(2\pi(f_0{+}f_c)t){+}cos(2\pi(f_0{-}f_c t)]{-}\frac{s_I(t)}{2}[sin(2\pi(f_0{+}f_c)t){+}sin(2\pi(f_0{-}f_c)t)]$$

$$\tilde{r}_I(t) = \frac{s_I(t)}{2}[cos(2\pi(f_0{-}f_c)t){-}cos(2\pi(f_0{+}f_c t)]{-}\frac{s_R(t)}{2}[sin(2\pi(f_0{+}f_c)t){-}sin(2\pi(f_0{-}f_c)t)]$$

$$iff_c = f_0\, then :$$

$$\tilde{r}_R(t) = \frac{s_R(t)}{2}[cos(4\pi f_0 t) + 1] - \frac{s_I(t)}{2} sin(4\pi f_0 t)$$

$$\tilde{r}_I(t) = \frac{s_I(t)}{2}[1 - cos(4\pi f_0 t] - \frac{s_R(t)}{2}[sin(4\pi f_0 t)]$$

1.2 Question 2

By stating $fc = f_0$, we bring the spectrum to a base. We use a fourier transform to compute the values of $\tilde{R}_I(f)$ and of $\tilde{R}_R(f)$:

$$\tilde{R}_R(f) = S_R * \frac{1}{2}\delta(f)*[\delta(f)+\frac{1}{2}[\delta(f+2f_0)+\delta(f-2f_0)]] - S_I(f)*\frac{1}{2}\delta(f)*[\frac{j}{2}[\delta(f+2f_0)-\delta(f-2f_0)]]$$

$$\tilde{R}_I(f) = S_I * \frac{1}{2}\delta(f)*[\delta(f)-\frac{1}{2}[\delta(f+2f_0)-\delta(f-2f_0)]] - S_R(f)*\frac{1}{2}\delta(f)*[\frac{j}{2}[\delta(f+2f_0)-\delta(f-2f_0)]]$$

$$\tilde{R}_R(f) = \frac{1}{4}[2S_R(f) + S_R(f-2f_0) + S_R(f+2f_0) + jS_I(f-2f_0) - jS_I(f+2f_0)$$

$$\tilde{R}_I(f) = \frac{1}{4}[2S_I(f) - S_I(f-2f_0) - S_I(f+2f_0) + jS_R(f-2f_0) - jS_R(f+2f_0)$$

This computation bring us to determine the filter characteristics. Gain : $|H(f)| = 2$
Bandwidth : $\frac{BW}{2} \leq f_{cut} \leq 2f_0 - \frac{BW}{2}$

1.3 Question 3

We need that $f_c > BW$ to e able to get the signal entirely and that $f_c < 2f_0$ to avoid having the small repeat of the signal that is center in $2f_0$ (in that case, we will face an overlapping). With a wide band, $f_0 < \frac{BW}{2}$, which is the cause of the overlapping. We therefore cannot work with a wide band signal.
Moreover, we can say here that we need a high band filter, that cannot be done analogically but can however be done numerically with a hilber filter.

1.4 Question 4

We apply the Shannon Nyquist theorem to chose the sampling period :

$$F_e \geq 2f_{max}$$

$$F_e \geq 2 * \frac{BW}{2}$$

$$F_e \geq BW$$

The sampling frequency have to be greater than the bandwidth.

1.5 Question 5

If filter after ADC : more expensive (cause you need to convert high frequencies) and possible errors after the ADC (due to number of levels for digital conversion)

For example, you can find the price of a CAN working at 1GHz here: https://www.digikey.com/product-detail/en/analog-devices-inc/AD9172BBPZ/AD9172BBPZ-ND/7606636

It would work to begin by the analog to digital conversion, but it would raise some issues. The price of this device would be much more expensive, since the conversion would take place in the range of the GHz, a high frequencies conversion costs a much higher price. The filter being after, it also mean that the signal is converted before being filtered and so ome conversion are using power to be done and are not especially necessary. On the number of conversion done, we also multiply the risk of errors.

1.6 Question 6

From $S_{RF}(t) = s_R(t)cos(2\pi f_0 t) - s_I(t)sin(2\pi f_0 t), \forall t \in R$ we get in frequential :

$$S_{RF}(t) = \frac{S_R(f)}{2}[\delta(f-f_0)+\delta(f+f_0)] - \frac{S_I(f)}{2j}[...])$$

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

Passing to temporal , the expression of the analytic signal and the envelop is:

$$S_a(f) = \begin{cases} 2S_{RF}(f), \forall f \geq 0, \\ 0, \forall f < 0 \end{cases}$$

$$\iff S_a(f) = S_R(f-f_0) + jS_I(f-f_0) = [S_R(f)+jS_I(f)] * \delta(f-f_0), \forall f > 0$$

$$s_A(t) = (s_R(t) + j\, S_I(t))\, e^{j2\pi f_0 t}$$

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

$$s(t) = s_R(t) + j\, s_I(t)$$

2 Frequency analysis of the recording

2.1 Question 7

In order to do a frequency analysis on the recording, we need the following blocks :

- Variable : holds a numerical value to input in other blocks simply, here we store the Fe and fc values.
- File source : takes a file as an input and returns a raw signal.
- Throttle : processes an input signal at a specific sample rate.
- QT GUI Sink : computes the FFT of a signal at a given center frequency and bandwidth and plots it. In the plotting menu, you can get mean values, hold maximum values and other options.

2.2 Question 8

On the figure 1, one can see the chain we implemented with the missing variables.

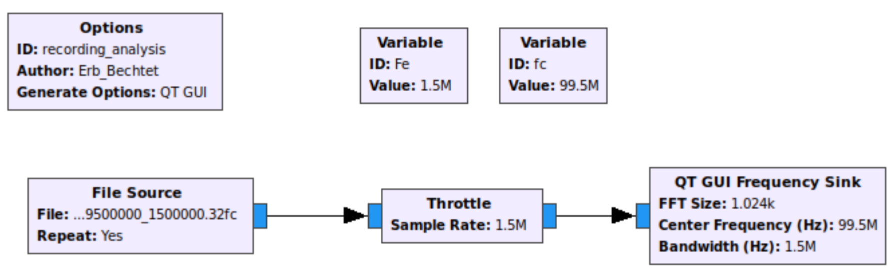

Figure 1: Chain firstly realized to process the signal

Two values were given to us, we therefore put 1.5 MHz for the sampling frequency and 99.5 MHz for the sampling frequency. In this exercise, the bandwidth had to be the same than the sampling frequency.
With these values, we can already have a first analysis of the signal. On the figure 2 , we can see 3 pics corresponding to the 3 radio stations whom signal are received by our processing chain.

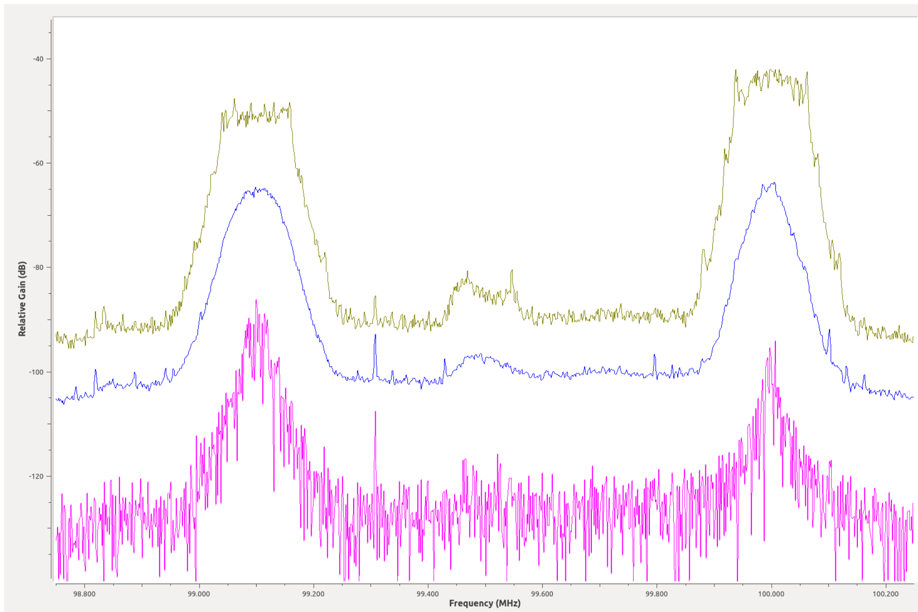

Figure 2: Graphic presenting pic of received signal

On this graphic we can see three lines. The blue one is the average line of the received signal, it is the signal we are going to work with for all the following values and computations. The green line is a drawing of the maximum of the signal received on the time we let the reception going on. The pink signal is acquired on the same principle but for the minimum values.

2.3 Question 9

As said previously, we can see 3 major signals : RFM at 99.1 KHz, Skyrock at 100 KHz and Nostalgie at 99.5 KHz. They are corresponding to the 3 pics in the figure 2. The medium pic at 99.5 KHz is very small, indicating a bad Signal-to-Noise Ratio and showing that this radio signal will be blurred at the place where the processing is done.

2.4 Question 10

In order to get the Signal to Noise Ratio (SnR), we need to read the power values in dB. We can read them directly on the graph (using the hold max value method). We have:
Skyrock:
$Signal = -64.88dB$
$Noise = -101.92dB$
$SnR = 10^{(-64.88-(-101.92))/10} = 10^3.7 = 5058$

RFM:
$Signal = -63.65dB$
$Noise = -102.11dB$
$SnR = 10^{(-63.65-(-102.11))/10} = 10^3.8 = 6309$

Nostalgie:
$Signal = -49dB$
$Noise = -89dB$
$SnR = 10^{(-92.29-(-101.82))/10} = 10^0.95 = 9.53$

It is important to note that the SnR depends on the chosen bandwidth, so the SnR we calculated is only valid for a bandwidth of 250 KHz

2.5 Question 11

To estimate the bandwidth, we looked at the plot values roughly at -3dB, a pretty standard and good way to estimate it:
$BW_RFM = [99.042, 99.159] = 0.117$
$BW_NOS = [99.444, 99.565] = 0.121$
$BW_SKY = [99.937, 100.066] = 0.129$

The average bandwidth is $> 100KHz$ (minimum imposed value)

3 Channel extraction by frequency transposition and low-pass filtering

3.1 Question 12

To center the pics at the frequency of 99.5 MHZ, we need to offset them of 400 MHZ for RFM and of -500 MHz for Skyrock. The following figure shows the pic of the Skyrock station, centred while it originally was situated at 100 MHz.

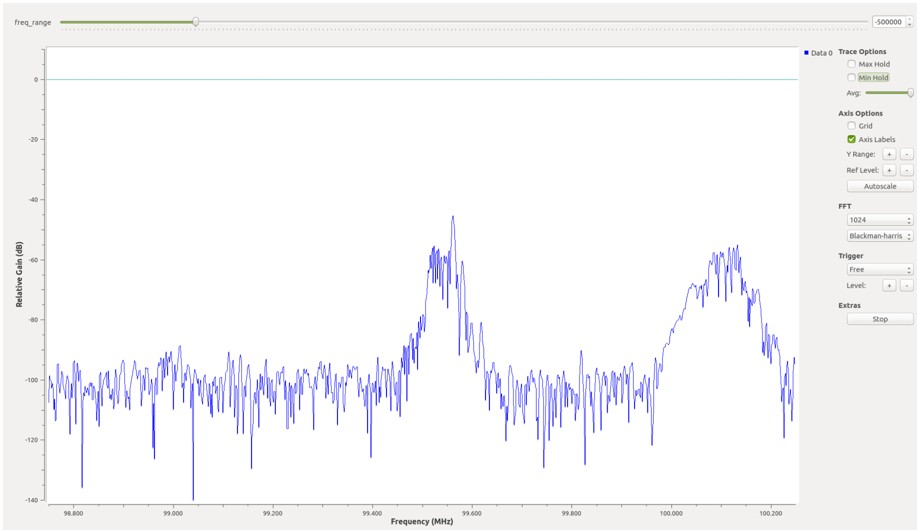

Figure 3: Graphic presenting the pic of Skyrock station centered

3.2 Question 13

If the frequency offset is higher than the sampling frequency, we do not respect the rule of the Shannon Nyquist theorem anymore. This will lead us to a mistake in the sampled signal. In that case, the signal that will appear on the screen won't be the real one but some point taken from it, showing most likely a more flat signal, that is not the original emitted one.

3.3 Question 14

From now on, we are going to use another scheme for signal processing. We added to the first protocol a "WBFM Receive" function to demodulate the frequency and a Low Pass filter to conserve the monophonic signal (which is a signal using only one channel). The final processing schematic is shown on figure 4 .

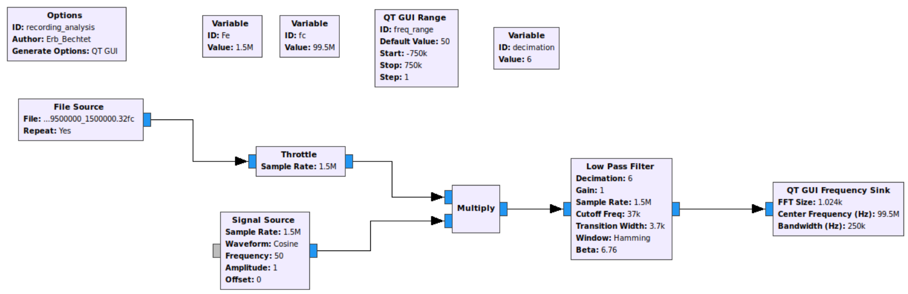

Figure 4: Chain for signal processing including the frequency demodulation, a low pass filter, and a decimation

The parameters are chose in coherence with the signal we analyse. For example, we know that the cut off frequency has to be half of the signal bandwidth. For RFM, the bandwidth is 34 KHz, therefore the cut off frequency is 37 KHz. Then , since the transition width is 10 % of the cut off frequency, the transition width for the analyse of RFM signal is 3.7kHz.
This diagram also takes into account a decimation parameter equal to 6. This means that we have enough points of data to take only one point out of 6.

3.4 Question 15

By applying Carsons's rule, we have:

$$\Delta_f = 75KHz$$

We also have $fm = 53KHz$
That makes $Bfm = 2 * (DeltaF + fm) = 256KHz$ This value confirms the theory.

3.5 Question 16

In order to actually listen to the music radio recording, we need to isolate the mono part, because gnuradio and our sound card don't allow for the listening of stereo sound.

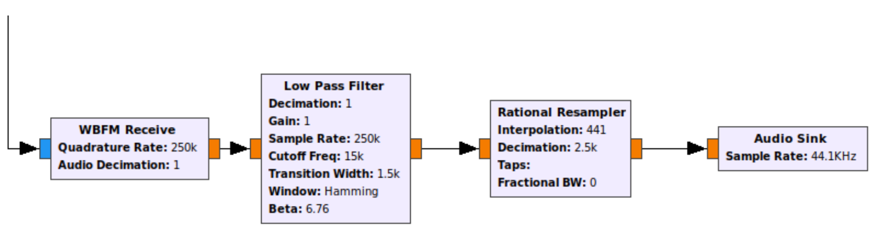

Figure 5: Chain for audio demodulating, mono sound filtering, and sound card adaptation

We first use a WBFM receive block to demodulate the audio signal. Then, we apply a low pass filter to cut off everything after the mono part. The last part is to resample the signal, originally at a 250 KHz sampling rate, into the 44.1 KHz sound card rate. We decimate and intertapolate by 2500 and 441 to have the perfect ratio and get the cleanest output.

3.6 Question 17

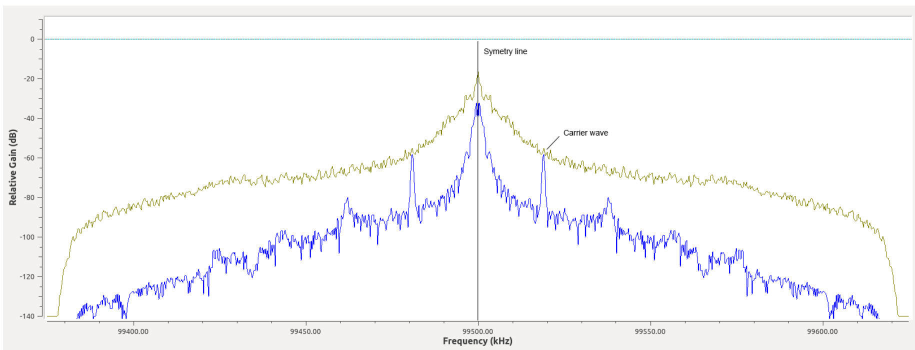

Figure 6: Plot of the demodulated signal

This plot was achieved by placing a frequency sink right after the WBFM receive block and before the low pass filter.
We can clearly see the symetry of the signal and the carrier wave at 19 KHz from the center frequency. We can roughly see the envelope of the mono and stereo signals, even though it's pretty hard to see on the screenshot.

3.7 Question 18

On Skyrock, it's **Jordi** who won the Stan Smith album.
On RFM, we can hear **Counting Stars by One Republic.**
YMCA by Village people is playing on Nostalgie.

3.8 Real-time implementation on a USRP receiver

We had the chance to try the acquisition system we just designed on live radio signal, by connecting the USRP we had at our disposal to our computer. The plotting of the live signal looked like this:

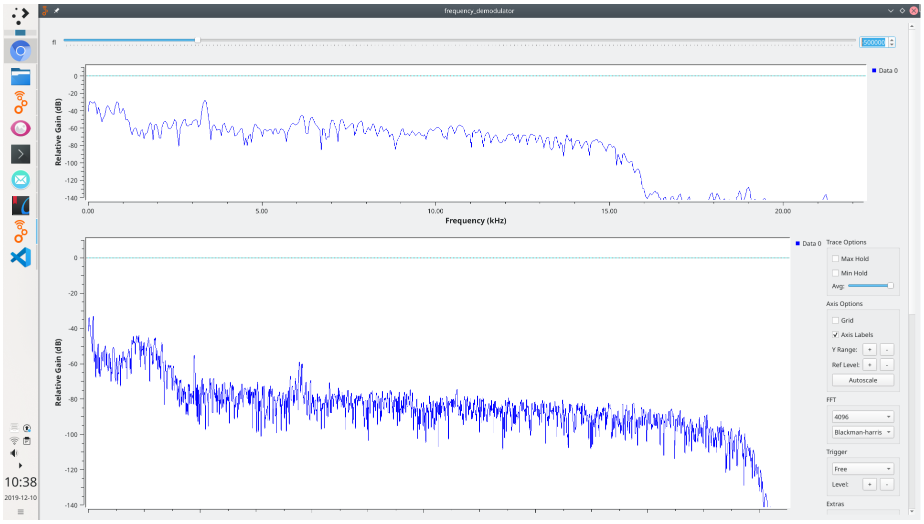

Figure 7: Plot of the live radio signal

4 Reception of VOLMET messages in AM-SSB

In this section, we try to use our USRP on a signal moduled in amplitude instead of frequency.

4.1 Question 19

When plotting the signal with a process identical to the one we used for the FM radio, we get this:

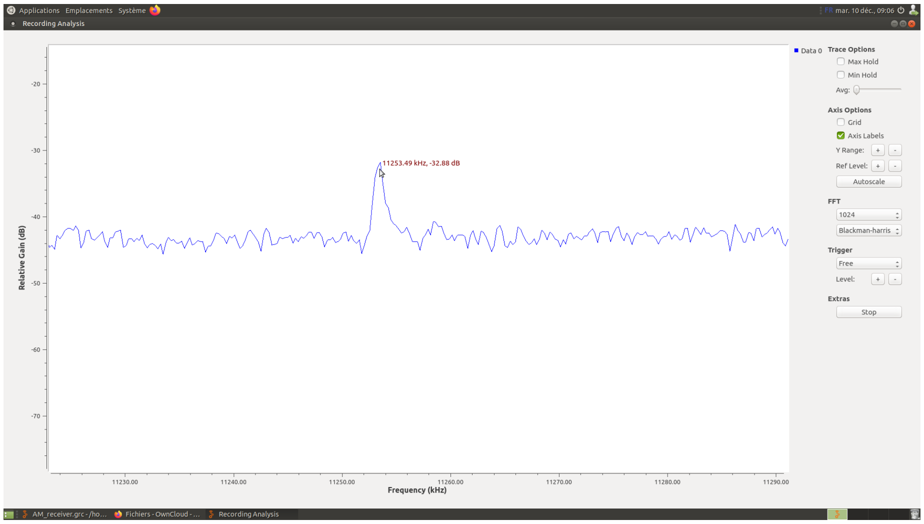

Figure 8: Plot of the plane signal

We can clearly see a peak at 11253 KHz, which exactly matches with the GBR 1 Military signal we were expecting to find. All the frequencies for Volmet messages can be found at this link: http://www.dxinfocentre.com/volmet.htm

4.2 Question 20

In order to center the channel with the maximum power, we simply need to shift by the difference between our current center frequency and the frequency we identified on the plot, like we did for the radios in question 12. We get f1 = 48 KHz, and the centered plot looks like this:

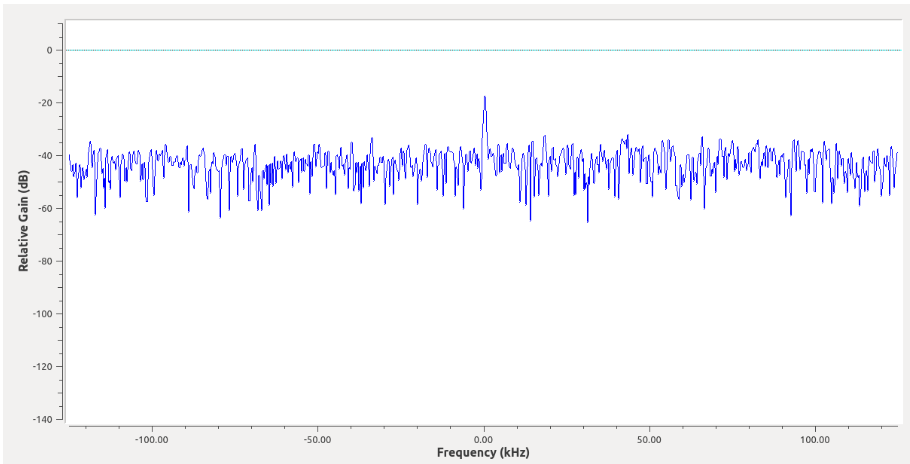

Figure 9: Plot of the plane signal centered

4.3 Question 22

If we look back at the plot of question 19, we can see that the envelope goes to the right of the 11253 peak and there is almost nothing to the left. This means that the source is only emitting on the right side of the peak, meaning we are in the upper side band.

4.4 Question 23

In order to demodulate and listen the AM signal, we first need to design a complex band filter. For that, we used the Filter Design Tool. We tweaked the default values a bit, first by diminishing the attenuation and then by enlarging the transition width. This was done for a purely practical reason, as keeping the default values was too much and crashed gnuradio every time we tried. Nonetheless, we managed to design the filter and got the following magnitude response:

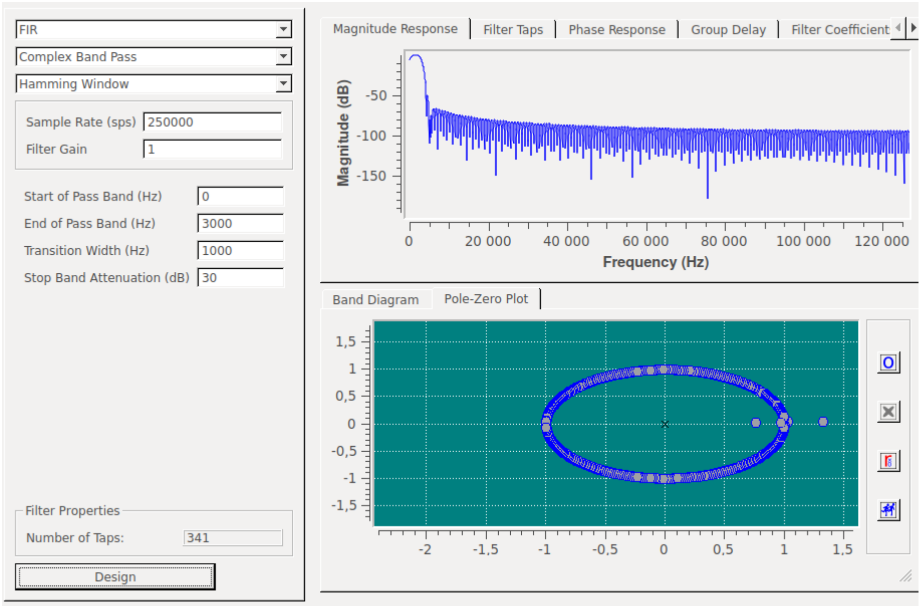

Figure 10: Design tool screenshot for our band pass filter

4.5 Question 24

If we apply our filter to the AM signal, we get the following:

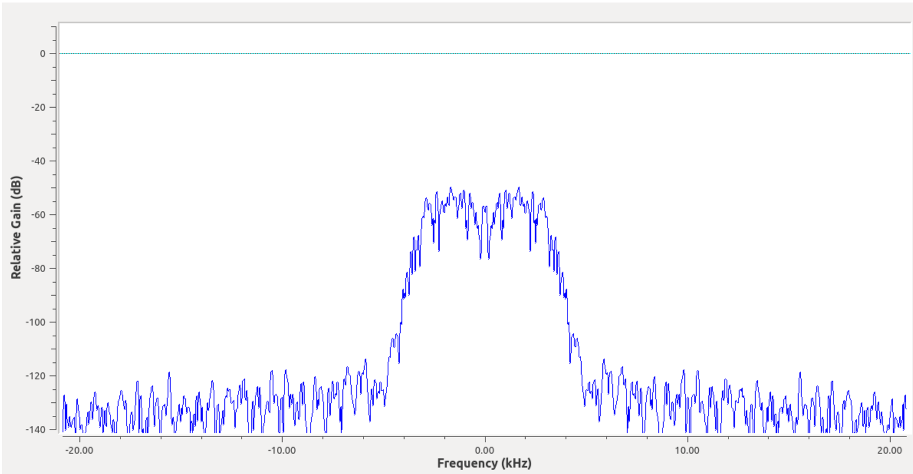

Figure 11: Plot of the plane signal after the filter

We can see that the signal is cut-off after about 4 KHz, which is fine for our 3 KHz-wide signal.

4.6 Question 25

To demodulate and listen to the signal, we proposed the following process:

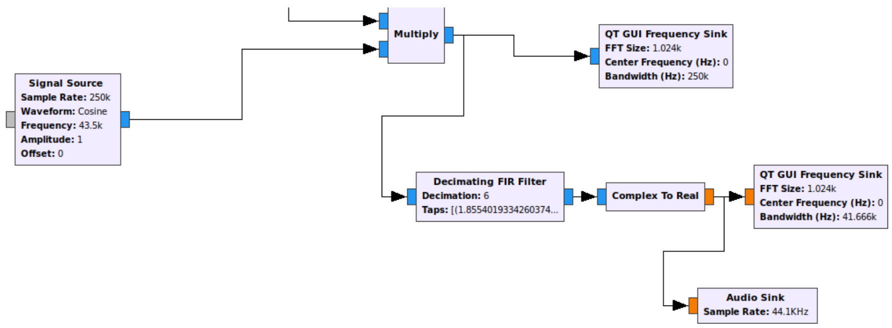

Figure 12: Blocks process to listen to the AM recording

We simply needed to convert our values to float, decimate to match the audio card sample rate as we did for the FM signal, and we were able to listen to the recording.

5 Conclusion

In the end, these three labs gave us a good practical introduction to Software-Defined Radios and especially the USRP. The theoretical part was necessary to make us understand the context and the power of this technology, while the application part was a really fun way to illustrate it. It sparked interest to look for similar technologies, notably HackRFs, which one of the group members had the chance to see in action. These devices are really powerful and offer a lot of possibilities, however these opportunities come with some risks, and one must be careful in what they go with them.