Filtering a LTE channel

1 Introduction

The purpose of this lab is to study analog baseband filtering. Our target application will be the reception chain of a 4G LTE receiver. The system and electrical simulations will be carried out respectively on Octave and on Cadence.

2 Scenario

Figure 1 shows a classical homodyne receiver. In this lab, we will just focus on the baseband part showed in the doted boxes which consists of:

- A low pass anti-alias filter (AAF)
- A variable Gain Amplifier (VGA)
- An Analog-to-Digital Converter (ADC)

For the sake of simplicity, we will just consider the 4 neighboring interferers of the useful signal. After the mixer, the signal at both channels I and Q will be as shown in Figure 1:

- 0 to 10 MHz $\longrightarrow$ The useful signal that folded on itself
- 10 to 30 MHz $\longrightarrow$ Interfers B and C on the top of each other, the combination of the two will be called interferer 1
- 30 to 50 MHz $\longrightarrow$ Interfers A and D on the top of each other, the combination of the two will be called interferer 2

Using the I and Q outputs shifted by 90 degrees, the 20 MHz useful signal is then reconstructed in digital using a phase shifter and a Hilbert filter. As said earlier, in this lab, we will just focus on the baseband part. The AAF is matched to 50 Ω and the signal at its input will be modeled as shown in Figure 2.

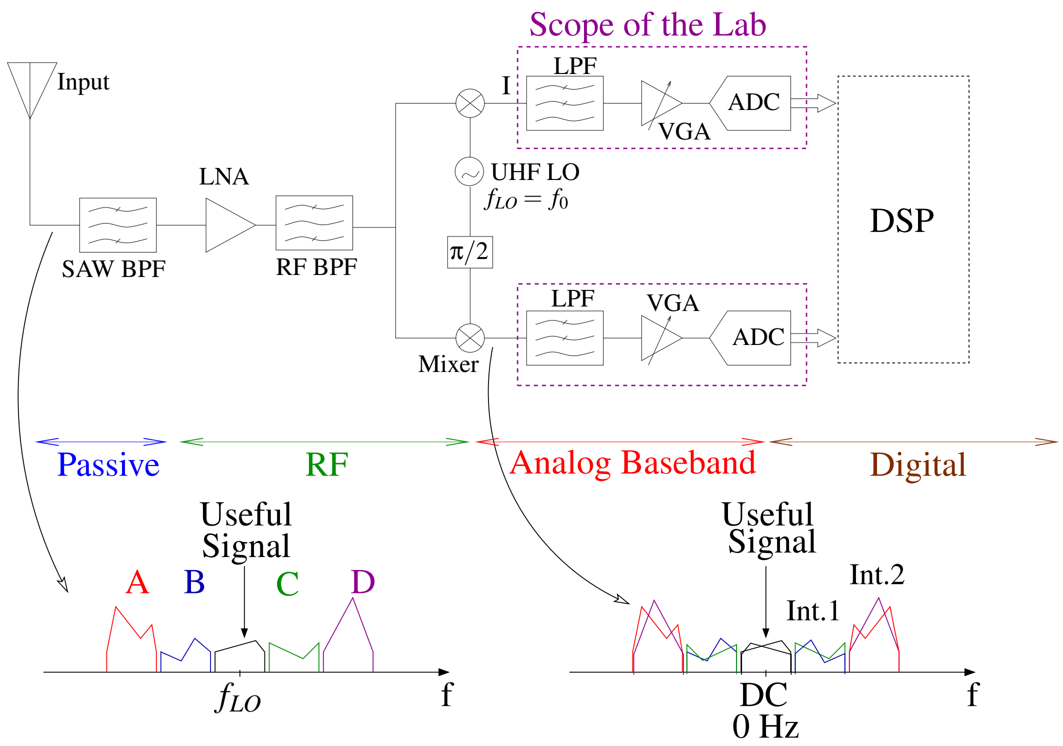

Figure 1: I/Q homodyne Receiver

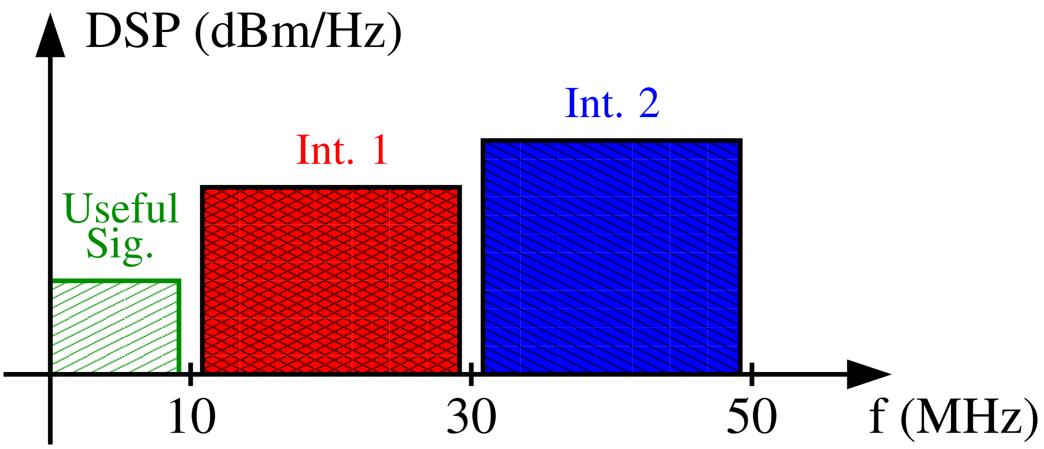

Figure 2: Spectrum at the AAF input

3 AAF system design

We would like to compare two solutions, the first one uses an ADC clocked at 20 MHz and the second at 40 MHz. The objective is to study the impact of this choice on the AAF complexity. Let us first begin by comparing the filter approximation.

Question 3.1 *Load the script* `FilterComparison.m`*, analyze the transfer functions and explain very briefly the advantages and drawbacks of each.*

For the remaining of the lab, for the sake of the simplicity of the analytical calculations, we will use a Butterworth approximation and we will assume that the channel power is concentrated in its center. We will therefore consider for filter specifications calculation that the interferers between 10 and 30 MHz are a sinewave at 20 MHz with a power of -45 dBm and the interferers between 30 and 50 MHz a sinewave at 40 MHz with a power of -40 dBm. This approximation is just done for the filter order calculation, when analyzing other aspects especially signal aliasing, the interferers should be considered as 20 MHz modulated signals.

Based on a error budgeting between the different components of the chain, it is required that the power of the highest interferers after aliasing in the useful band should be at least 20 dB lower than the input signal [1]. The maximum attenuation inside the 10 MHz useful band should be lower than 1 dB. The power of the useful signal is -60 dBm.

Let us first begin with the ADC clocked at 20 MHz.

Question 3.2 *Calculate the minimum attenuation that should be applied on interferer 1 [10 to 30 MHz] modeled by a sinewave at 20 MHz.*

Question 3.3 *Plot the filter attenuation template on which should be specified $f1$, $f2$, A_{min} and A_{max}.*

Question 3.4 *Calculate the order needed for a filter with a Butterworth approximation to satisfy these requirements.*

Question 3.5 *Load the script* `ButterworthFilterSpecifications.m`*. Set $f1$, $f2$, A_{min} and A_{max} to the values determined in the previous questions. Compare the calculated order to the script results. Determine the attenuation slope in dB/decade and compare to the theory.*

Question 3.6 *Load the script* `BasebandChain.m`*. Set $FsADC$ to 20 MHz and set $Fcut$ and $Order$ to the values determined in the previous question. Observe the signal spectrum at the different points of the chain. At which frequency interferer 1 has aliased [2], explain why. Pick up the power difference between the main signal and interferer 1 and make sure that it is higher than the targeted value.*

[1] For this calculation, you can also approximate the input signal to a sinewave with a power of -60 dBm

[2] The frequency of the sinewave modeling interferer 1 and 2 were slightly moved to 20.3 MHz and 41 MHz for simulation purposes

Let us now pass to the ADC clocked at 40 MHz. The objective of using an ADC clocked higher than the Nyquist rate is to reduce the constraints on the AAF. A high order digital filter will follow the ADC to select the bandwidth [0 10 MHz].

Question 3.7 *Draw by hand the spectrum after aliasing with an ADC clocked at 40 MHz. Use different colors or forms for the useful signal and the different interferers.*

Question 3.8 *Based on the result on question 3.7, explain which interferer becomes the most critical.*

Question 3.9 *Same as question 3.3 but with an ADC clocked at 40 MHz.*

Question 3.10 *Same as question 3.4 but with an ADC clocked at 40 MHz.*

Question 3.11 *Same as question 3.5 but with an ADC clocked at 40 MHz.*

Question 3.12 *Same as question 3.6 but with an ADC clocked at 40 MHz.*

Question 3.13 *What is the impact of interferer 1 on the chain performance? Tip: Change the power of the interferer and analyze its impact.*

Question 3.14 *List the advantages and drawbacks of the two solutions one with respect to the other.*

4 AAF Electrical design

After a long hesitation and hours of consideration, we decide finally to adopt the solution with a 40 MHz ADC. In this section, we will study the electrical implementation of the AAF on Cadence. We will use the Sallen-Key cell shown in Figure 3 as the elementary cell of the filter.

Question 4.1 *Show that the transfer function of the Sallen-Key cell can be written as*[3]*:*

$$H(p) = \frac{V_{out}(p)}{V_{in}(p)} = \frac{1}{1 + 2RC_2p + R^2C_1C_2p^2}$$

Question 4.2 *Write $H(p)$ using the following form and calculate the expression of the quality*

[3] p represents the Laplace variable

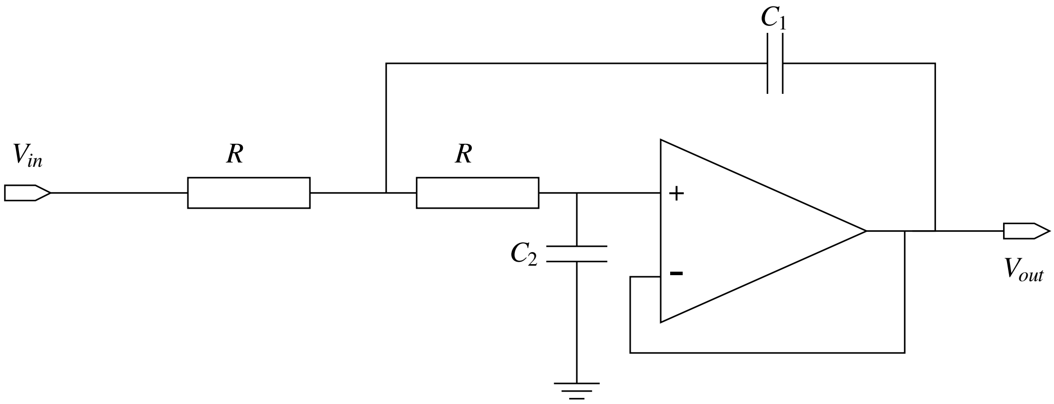

Figure 3: Sallen-Key cell

Table 1: Butterworth Quality factor for orders 2 to 8. For odd order filters, a normalized single pole stage $\frac{1}{1+\frac{s}{\omega_0}}$ should be cascaded with the Sallen-Key cells.

Filter Order	Stage 1 Q	Stage 2 Q	Stage 3 Q	Stage 4 Q
2	0.7071			
3	1			
4	0.5412	1.3065		
5	0.6180	1.6181		
6	0.5177	0.7071	1.9320	
7	0.5549	0.8019	2.2472	
8	0.5321	0.6013	0.899	2.5628

factor Q and the ω_0:

$$H(p) = \frac{V_{out}(p)}{V_{in}(p)} = \frac{1}{1 + \frac{1}{Q\omega_0}p + \frac{1}{\omega_0^2}p^2}$$

Question 4.3 *Based on the Butterworth Table (Table 1) and on the order and F_{cut} of the filter determined in the previous section, determine the number of cell needed to build your AAF and the values C_1 and C_2 for each cell. For sake of simplicity, all R will be set to 10 kΩ.*

5 AAF implementation in Cadence

In this section, we will focus on the implementation of the AAF on Cadence.

Question 5.1 *In Virtuoso, select Tools ⟶ library Manager and then go to the library IC_filter and open the cell AAF, view schematic.*

Question 5.2 *Build your AAF by instantiating*[4] *and connecting your components* [5]*.*

Question 5.3 Set the values [6] of your components to the values found in the previous section

Question 5.4 Name[7] the input net vin and the output net vout.

Now, that our circuit is built, it is time to simulate it. We will begin with a small-signal AC analysis.

Question 5.5 *In the Library Manager, library IC_filter, cell AAF, open the spectreAC view. This will open the simulation environment. Push on the play icon to launch the simulation. Observe the results and make sure that your filter has the desired transfer function.*

We will now carry out a time domain simulation of the AAF. A FFT will be performed on the input and output signals to observe the attenuation per signal. The input signal is similar to the one we had in section, 3 sinewaves modeling the useful signal, interferer 1 and interferer 2.

Question 5.6 *Close the ADE window. In the Library Manager, library IC_filter, cell AAF, open the spectreTran view. Push on the play icon to launch the simulation. Observe the results and make sure that your filter has the desired attenuation for interferer 1 and interferer 2.*

[4]To instantiate a component, push i. A window will open to allow you to select the symbol view of your component. Resistors (res), capacitors (cap) and ground (gnd) are in library analogLib. For Operational Amplifiers, use OpAmp cell of IC_filter.

[5]To connect the different components together, use w

[6]Select the component and push q

[7]To name a net, push l, write the name and then select the considered net