Bank filtering for sound compression

Baptiste Cavarec
940321-T197

Hugo Lime
920707-4635

December 13, 2015

Abstract

In this report, we describe a simple implementation of bank filtering for sound compression. We explain how to construct theses filters and the effect of quantization of the decimated signals. We then provide a method to reduce the signal bitrate while preserving quality by choosing a proper bit allocation for the decimated signals.

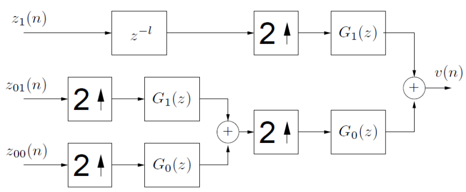

Figure 2: Synthesis filter bank.

1 Introduction

In this project, we intend to study the common problematic of data compression, in our case for sound signals. We have, as an input, a sound signal sampled at a 8 kHz frequency. In order to store and compress our signal, we need to implement an analysis filter bank. The one we decided to implement can be seen in Fig. 1. As one can also see in Fig. 1, we also compute quantization of our 3 signals obtained by the analysis. This is the core of our data storage issue, as we need to quantize in order to lower the signal's data rate. This obviously induces quantization noise and deteriorates the signal, we then have to find the best way to avoid the noise.

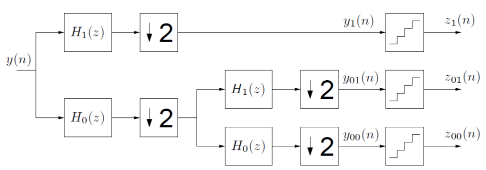

Figure 1: Analysis filter bank followed by quantization.

Then, we use the synthesis filter bank presented in Fig. 2 in order to reconstruct the signal. Sec. 2.1 will present the filters used and their impact on the signal.

In order to evaluate the error introduced by the above mentioned quantization scheme, we need to introduce a measure of our quantization/reconstruction scheme efficiency as Sec. 2.1 ensures perfect reconstruction, this measure would only seize the impact of the quantization on the compressed sound. The measure we consider is called Signal to Quantification Noise Ratio (SQNR) and can be evaluated as in Eq. 1.

$$SQNR = \frac{E[y^2(n)]}{E[(y(n-L)-v(n))^2]} \tag{1}$$

2 Bank filtering

2.1 Filters

We presented in Sec. 1, our structure is based on Analysis/Synthesis Filter Bank scheme (Fig. 1 and Fig. 2). In order to ensure the reconstruction of our signal, we need to design the filters so that they respect Eq. 2, Eq. 3 and Eq. 4.

$$H_1(z) = G_0(-z) \tag{2}$$

$$G_1(z) = -H_0(-z) \tag{3}$$

$$G_0(z)H_0(z) - G_0(-z)H_0(-z) = 2 \cdot z^{-l} \tag{4}$$

Moreover, in order to perform the synthesis scheme, we need to know the delay introduced by or Analysis/Synthesis bank. By using the formalism presented in Eq. 5 we ensure a delay $l = 3$ for our "first order" filter bank, if Q is an order 2 polynomial in z^{-1}. If we want to ensure that Eq. 4 is satisfied, we must have $Q(z) = -\frac{1}{16} + \frac{z^{-1}}{4} - \frac{z^{-2}}{16}$.

$$G_0(z)H_0(z) = (1+z^{-1})^4 Q(z) \tag{5}$$

Given this, we choose:

$$G_0(z) = \frac{1}{2}\left(1 + z^{-1}\right)^2 \tag{6}$$

it then leads to

$$H_0(z) = \frac{1}{8}\left(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}\right) \tag{7}$$

Hence,

$$H_1(z) = \frac{1}{2}\left(1 - z^{-1}\right)^2 \tag{8}$$

and

$$G_1(z) = -\frac{1}{8}\left(-1 - 2z^{-1} + 6z^{-2} - 2z^{-3} - z^{-4}\right) \tag{9}$$

Fig. 3 shows the shape of these filters in frequency domain.

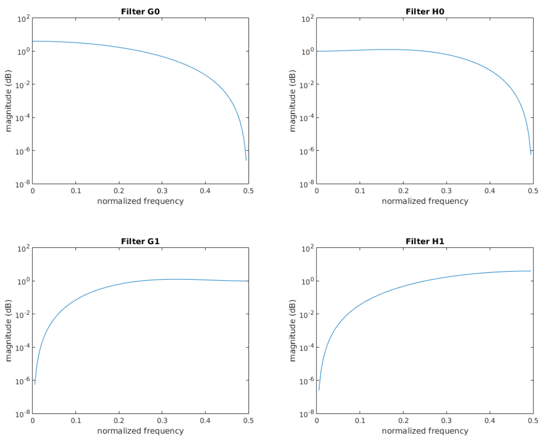

Figure 3: Frequency response of the filters used in our Analsys/Synthesis scheme

Given this, we will ensure in Sec. 2.3 that these filters perform perfect reconstruction.

2.2 Total delay of the system

As one can see in Sec. 2.1, we introduced a delay by implementing these filters. The delay created by the "first order" Analysis/Synthesis scheme is given and is $l = 3$. Therefore, we implemented an artificial delay of 3 on $z1(n)$ to compensate the delay induced by the nested Analysis/Synthesis scheme on the other branch. These delays are done on downsampled signals and result in a $2 \cdot 3$ delay after upsampling. Moreover, the external Analysis/Synthesis also created a delay of 3 leading the whole delay to be $L = 9$.

2.3 Perfect reconstruction

In order to check our filter bank implementation, we measured the avarge error between the input signal and the delayed output signal.

$$error = \frac{1}{N}\sum_{n=1}^{N-1} |v(n+9) - y(n)|^2 \tag{10}$$

with N being the total number of samples. As expected by our choice of filter, we obtain a null error meaning that we have perfect reconstruction of the signal.

3 Decimated signals

3.1 Effects of the downsampling on the signal

The downsampling results on the spreading of the spectrum over a broader part of our normalized frequency domain. For instance, the $[0; \frac{1}{2}]$ domain is mapped onto $[0; 1]$ when we downsample of a factor 2 as for the considered application. Then aliasing occurs, adding the parts that are over [-1/2;1/2] periodically as one can see on Eq. 11.

$$y(\nu) = \frac{1}{2}\sum X((\nu - k)/2) \tag{11}$$

This phenomenon can be seen in Fig. 4 we can clearly see that the global behaviour of the downsampled signal looks like the $[0, 0.25]$ part of the original one but it is not totally alike, due to aliasing. As expected from the previous curve, the downsample signal sounds "deeper" than the original one, as we have more low frequencies.

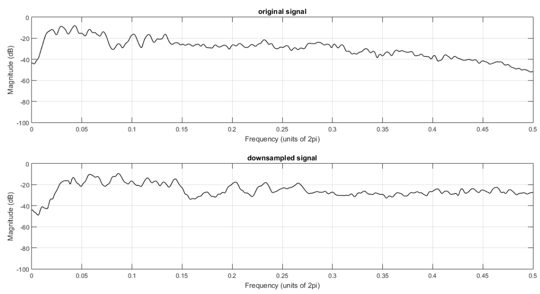

Figure 4: Effects of the downsampling on the thank.wav signal

3.2 Relation with the input signal

The decimated signals are linked to the input signal as they represent a certain band of its frequencies.

If we study the filters H_0 and H_1, we can consider them to be respectively a low-pass and high-pass filter. Graphical study with Fig. 3 shows that the $-3dB$ band pass of H_0 is approximately [0,0.3] and the one of H_1 is [0.2,0.5]. We can therefore consider that the decimated signals would represent the spectrum of the signal in theses bands. Fig. 5 shows how the spectrum of the input signal (here file "thank.wav") is split in three with these filters.

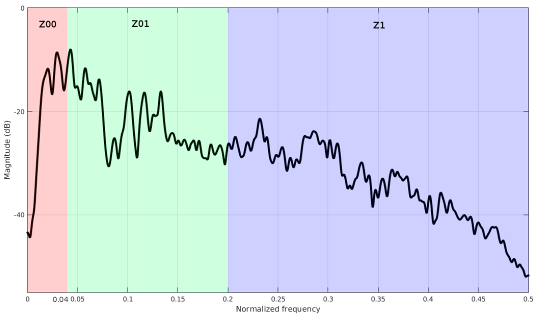

Figure 5: Spectrum splitting of input signal

4 Quantization

4.1 Effects of quantization

In order to store or transmit our signal we need to assign a certain number of bits to represent the value of each sample. A greater number of bits would allow to represent more values and thus allowing higher fidelity but would also require larger memory or bitrate. Fig. 6 shows that a 32 $kbit/s$ bitrate would give a SQNR of 12.87 dB for the file "thank.wav".

4.2 Quantization in filter banks

In order to improve this result, we can use our filter bank by transmitting the decimated signals $z_1(n)$, $z_{01}(n)$ and $z_{00}(n)$ instead of $y(n)$. We can allocate different numbers of bits for the decimated signals which allows to have higher fidelity for some frequencies which are more important. Lets define b_1, b_{01} and b_{00} as the number of bits allocated to represents samples of the signals $z_1(n)$, $z_{01}(n)$ and $z_{00}(n)$. Given the fact that the rate of $z_1(n)$ is half the one of $y(n)$ and that the rate of $z_{01}(n)$ and $z_{00}(n)$ is one fourth the one of $y(n)$, the total bitrate is given by Eq. 12.

$$bitrate = 8(\frac{b_1}{2} + \frac{b_{01} + b_{00}}{4})\ kbit/s \qquad (12)$$

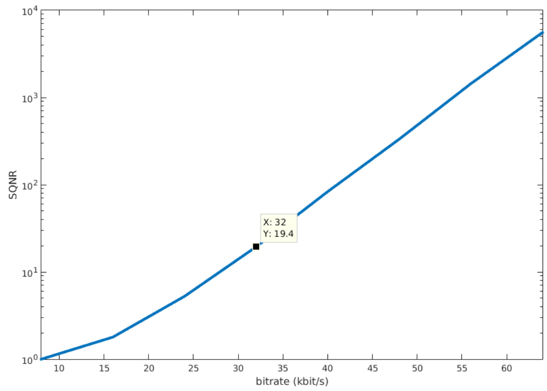

Figure 6: SQNR for different bitrates without filter bank

4.3 Bit allocation

If we want to improve the SQNR obtaine in Sec. 4.1 the Eq. 12 leads to equation 13

$$2b_1 + b_{01} + b_{02} = 16 \qquad (13)$$

To obtain the best bit allocation, we tested all combinations. For the file "thank.wav", the best bit allocation is $b_1 = 3$, $b_{01} = 4$ and $b_{00} = 6$ giving a SQNR of 16.9 dB.

However, when testing on the file "orinoccio.wav", the best bit allocation is $b_1 = 2$, $b_{01} = 5$ and $b_{00} = 7$ giving a SQNR of 20.3 dB instead of 9.6 dB without filter bank.

The best bit allocation thus depend on the sound signal. As we can see on Fig. 7, The magnitude of the spectrum decrease with the frequency which explains why for both signals, the best allocation is to give more than half the bandwith to the low-frequencies. Moreover, for the "orinoccio.wav" signal, we can notice that the spectrum is very much concentrated around very low frequencies which explains why $b_{00} = 7$. Therefore, it is also possible to devise a bit allocation only by analysing the signal spectrum but it will be worse than testing all bit allocations.

4.4 Perceived quality

To measure the fidelity of the reconstucted signal, we have used the SNQR (Eq. 1). Basicaly, it measures the error, sample by sample, between the recontructed signal ans the input signal. In order to determine if the SQNR is a good measure of the

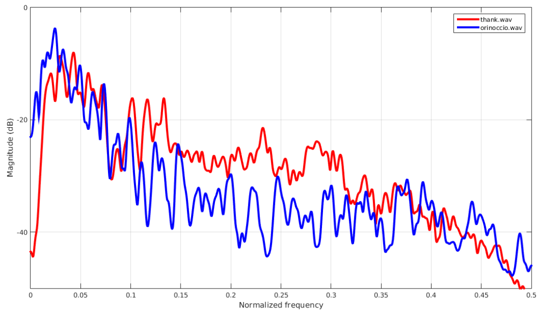

Figure 7: Spectrum of input signals

perceived distorsion we computed different configurations of bit repartition that lead to a different SQNR. We have heard that the higher the SQNR, the less audible the noise was, that was to be expected from the formula of the SQNR (Eq. 1). During this experiment we also heard that two close SQNR can lead to a different "feeling" of the noise you hear, depending if the quantization noise is located more in the low frequency band or the high frequency ones. This leads us to say that even if SQNR is a good measure of the quantization noise, it is not really representative of the experience of the quality of a human user.

5 Conclusions

To conclude this study of bank filtering and sound compression, we can say that:

- We can devise filters to create filter banks with perfect reconstruction.
- Quantization, which is necessary to store or transmit a signal, creates error in the reconstructed signal.
- Allowing different number of bits to the decimated signal can greatly improve the SNQR. We described a method to obtain the best bit allocation.

We can also add that in order to improve the compression, a deeper study of how humans perceive a sound should be done. For example, it could be interesting to see if some frequencies are more important for a human hear and also how humans perceive two neighbouring frequencies.