SMARC method

J.Prado, B.Mathieu, TSI department

Institut Télécom; Télécom ParisTech; CNRS/LTCI
46 rue Barrault, 75683, Paris CEDEX 13

email : jacques.prado@telecom-paristech.fr
email : benoit.mathieu@telecom-paristech.fr

february 21, 2011

Contents

List of Figures

1 Method

1.1 Introduction

The development of the method described below follows a request to change the sampling rate of audio files in order to treat them by other methods such as extraction of features, resynchronization with video, but it may also extend to conversion issues associated with creating audio CDs, the soundtrack is often recorded at 48 or 96 kHz with a resolution 24-bit and converted into 44.1 kHz/16 bits which is the standard for a CD audio.

We have tested methods which are often derived from the fundamental article by J. Smith [?], and based on the resampling of the continuous signal reconstructed from the Shannon interpolation formula. If the method described in [?] yields good results for interpolation (upsampling), some problems remain in the decimation process (downsampling). Indeed, in the first case the reconstructed signal before resampling does not have frequencies above the Nyquist frequency (half sampling frequency) and therefore the resampling by interpolation is possible without major difficulties. It's different in the case of decimation, because frequencies above the Nyquist frequency of the output must be removed, otherwise aliasing appears and this effect is difficult to manage from the Shannon interpolation formula. It follows a difficult process that is apparently not fully resolved by the tools we have tested. Some changes, such as the transition from 48kHz to 44.1kHz, present difficulties in the way that we have to virtually up to a frequency, the lcm of the two frequencies, to create the samples at a multiple of the output frequency. As 48000/44100=160/147, the lcm is 7.056 MHz and in order to avoid aliasing, we need a low-pass filter of about 38560 coefficients to cut frequencies above 22,05 kHz. Finally some proposed methods make use of interpolation techniques, such as linear interpolation or spline cubic interpolation, which has the effect of altering the information contained in the signal and can biased the following signal processing such as features extraction.

We propose a method for frequency conversion without other modification of the information contained in the signal that the band reduction associated with the concept of decimation. Furthermore, we implement a multistage conversion that allows i) to reduce the complexity of the required filters and ii) to work these filters to the lowest frequency allowing a hardware implementation adapted to particular cases.

The description of the method is followed by the documentation of *Smarc*, a software that has been developed from the method.

Some fundamental concepts of decimation and interpolation are recalled in what follows.

1.2 Decimation by an integer ratio M

Here $x(n)$ denote a full-band signal sampled at a normalized sampling rate 1, $\left(\left|X(e^{j2\pi f})\right|\right) \neq 0, f \in \left[-\frac{1}{2}, \frac{1}{2}\right]$.

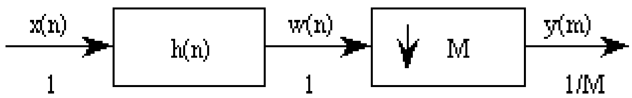


Figure 1: Downsampling process.

To avoid aliasing at the lower sampling rate, $1/M$, it is necessary to filter the original signal $x\left(n\right)$ with a low-pass filter giving the signal $w\left(n\right)$. The sampling rate reduction is then achieved by forming the sequence $y\left(n\right)$ by extracting evrey Mth sample of $w\left(n\right)$. The filter is defined by:

$$\left|H(e^{j2\pi f})\right| = \begin{cases} 1 \text{ pour } |f| < \frac{1}{2M} \\ 0 \text{ ailleurs} \end{cases}$$

In the case of an ideal low-pass filter the signals spectrum can be expressed as: $Y(e^{j2\pi f}) = \frac{1}{M} X\left(e^{j2\pi \frac{f}{M}}\right)$

1.3 Interpolation by an integer ratio L

In this case (figure 2) it follows that $1 = LF$. The sampling rate of the signal $x\left(n\right)$ is increased by the factor L by inserting $(L-1)$ zero-valued samples between each sample of $x\left(n\right)$.

Hence $w(m)$ and $x\left(n\right)$ are related by:

$$W\left(z\right) = \sum_{m=-\infty}^{+\infty} w\left(m\right) z^{-m} = \sum_{m=-\infty}^{+\infty} x\left(m\right) z^{-mL} = X\left(z^{L}\right)$$

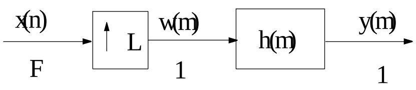


Figure 2: Upsampling process.

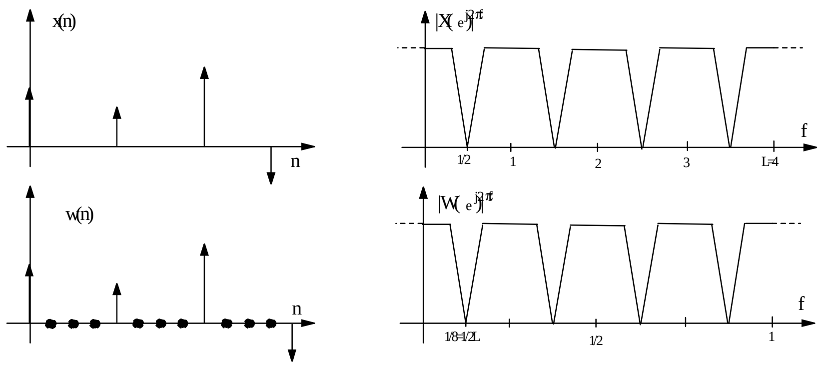


Figure 3: Signals and there spectrum before and after 0 insertion.

The interpolation process is then achieved by suppressing the periodic components contained in $W\left(e^{j2\pi f}\right)$, hence we need a low-pass filter defined by:

$$H\left(e^{j2\pi f}\right) = \begin{cases} L \text{ pour } |f| < \frac{1}{2L} \\ 0 \text{ ailleurs} \end{cases}$$

The spectrum can be expressed as:

$$Y(e^{j2\pi f}) = H(e^{j2\pi f})X(e^{j2\pi fL})$$

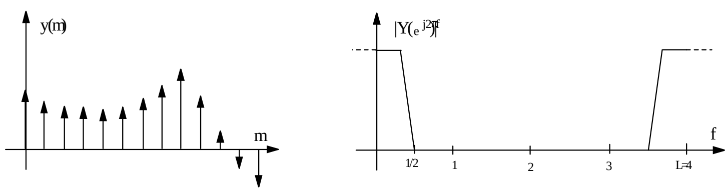


Figure 4: Signal and spectrum after filtering.

In the case of an ideal low-pass filter the spectrum of $y\,(n)$ is:

$$Y(e^{j2\pi f}) = \begin{cases} LX(e^{j2\pi fL}) \text{ pour } |f| < \frac{1}{2L} \\ 0 \text{ elsewhere} \end{cases}$$

1.4 Conversion d'un facteur L/M

$x(n)$ is a signal sampled at frequency F and it will be converted in a signal $y(m)$ sampled at frequency F' such that $\frac{F'}{F} = \frac{L}{M}, L, M \in \mathbb{N}$ (figure 5).

Decimation and interpolation processings are simultaneous so the transfert function $H(z)$ of the filter should be defined as:

$$H\left(e^{j2\pi f"}\right) = \begin{cases} L \text{ for } |f"| \leq \min\left(\frac{1}{2L}, \frac{1}{2M}\right) \\ 0 \text{ elsewhere} \end{cases}$$

where $f"$ is normalizedrelatively to LF.

Hence :

$$Y(e^{j2\pi f'}) = \begin{cases} \frac{L}{M}X\left(e^{j2\pi\frac{Lf'}{M}}\right) & \text{for} \quad \left|f'\right| \leq \min\left(\frac{1}{2}, \frac{M}{2L}\right) \\ 0 & \text{elsewhere} \end{cases}$$

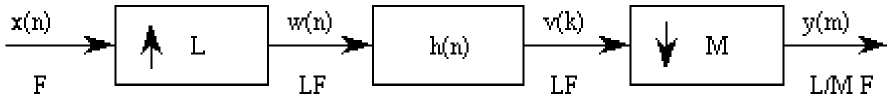


Figure 5: Principle of frequency conversion .

1.5 Structures

For an example, the conversion from 48kHz to 44.1kHz needs a digital filter with a very long size impulse response. However this problem can be solved with a two steps procedure. In the first step the operations of interpolation and decimation are reversed in order to reduce the number of processed samples (see [3]), and in a second step the conversion is performed in several stages. To illustrate this second step, just write the ratio as a product:

$$\frac{L}{M} = \frac{44100}{48000} = \frac{147}{160} = \frac{3*7*7}{2*2*2*2*2*5}$$

The decomposition is not unique, so the conversion can be performed in different ways :

$$\frac{L}{M} = \frac{3}{4} * \frac{7}{4} * \frac{7}{10} = \frac{3}{4} * \frac{7}{8} * \frac{7}{5} = ...$$

The optimization process is described in [4], [1] and [2].

1.5.1 First step

Interpolation and décimation operators properties

Permutation An important theorem (see [3]) allows to write the equivalences illustrated in figure (6).

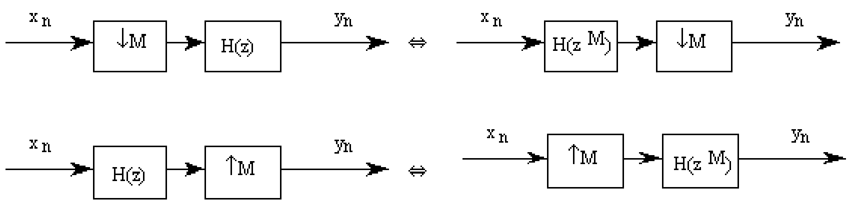


Figure 6: Illustration of permutation.

Linearity All memoryless operation can be permuted with decimation and interpolation (figure 7).

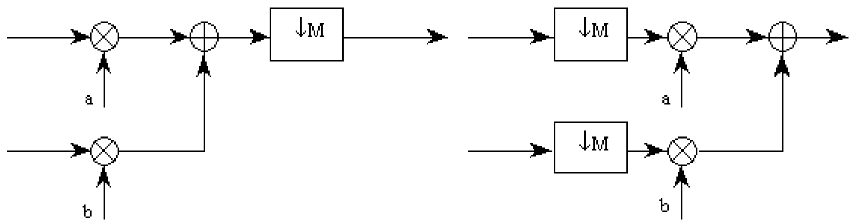


Figure 7: Linearity

Permutation of the operators Figure (8) is correct iff M and L are relatively prime.

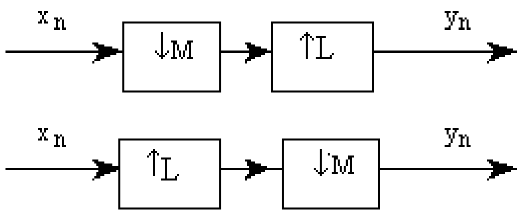


Figure 8: Operators permutation.

Generalization Generalization is illustrated by the following scheme (fig. 9) :

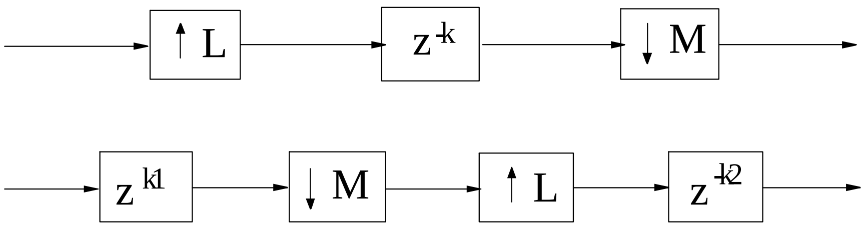


Figure 9: Generalization.

Assuming that L and M are relatively prime one can rewrite z^{-k} as :

$$z^{-k} = z^{Lk_1} z^{-Mk_2}$$

Where k_1 and k_2 are integers and solution of : $k = Mk_2 - Lk_1$.

Polyphase implementation There exist two types of polyphase components for a filter, the first one is used for *decimation* and is called *type 1* and the second one is used for *interpolation* and is called *type 2.*

Type 1 For this representation, the transfert function is written as:

$$H(z) = \sum_{m=0}^{M-1} E_m\left(z^M\right) z^{-m}$$

where : $E_m(z) = \sum_{n=-\infty}^{+\infty} e_m(n) z^{-n}$ and $e_m(n) = h_{nM+m}$.

The $E_m(z)$ are the polyphase components of $H(z)$ and assuming $H(z)$ is an ideal low-pass filter, then $E_m\left(e^{j2\pi f'}\right)$ is an all-pass filter in the frequency band $f' = [-1/2, 1/2]$ if $\left|H\left(e^{j2\pi f}\right)\right| = 1$ for $|f| < \frac{1}{2M}$, $\left(f' = \frac{f}{M}\right)$.

Type 2 For this representation, the transfert function is written as:

$$H(z) = \sum_{l=0}^{L-1} R_l\left(z^L\right) z^{-(L-1-l)}$$

where : $R_l(z) = E_{L-1-l}(z) = \sum_{n=-\infty}^{+\infty} r_l(n) z^{-n}$ and $r_l(n) = e_{L-1-l}(n) = h_{nL+L-1-l}$.

Remark : in the case of FIR filters, implementation is made easier if the filter length N is a multiple of M or L.

Examples

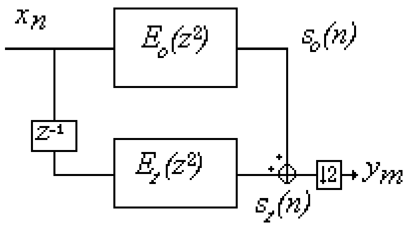


Figure 10: Direct implementation.

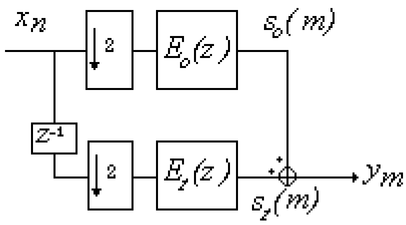


Figure 11: Polyphase type 1 implementation .

Decimation $M = 2$

Interpolation $L = 2$**.** The scheme is just a transposition of the previous one (type 1):

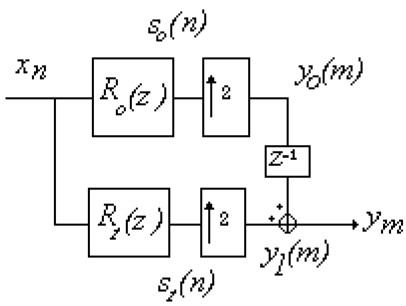


Figure 12: Polyphase type 2 implementation.

1.5.2 Second step

Multistage implementation Assume that the interpolation factor L can be decomposed as a product of prime factors: $L = \prod_{j=1}^{J} L_j$. This yields to the scheme of figure (13).

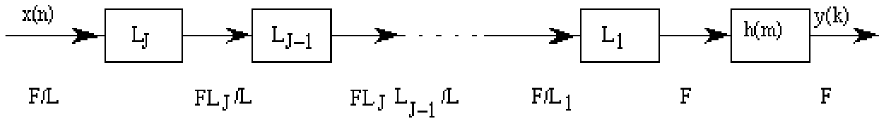


Figure 13: Multistage implementation with a unique filter.

This can be improved (figure 14) if $H(z)$ is decomposed in a series of $H_j(z)$ according to each factor L_j.

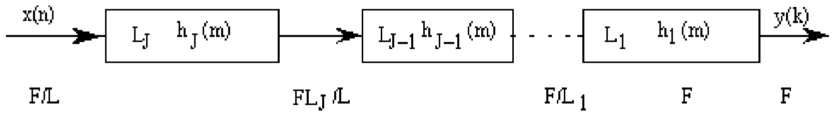


Figure 14: Multistage implementation with multiple filters.

This can also be done for the decimation part with $M = \prod_{j=1}^{J} M_j$. In this way the computation load can be drastically reduced and the synthesis of each filter is simplified. The main problem is the choice of the decomposition of L and M. An attractive solution is given in [4], [1] and [2], and has been used for the development of *smarc*.

Multistage implementation offers a significant avantage to the filter synthesis, indeed for some stages aliasing can be allow in the transition band of the filter because this aliasing will be suppressed by the following filters. This is illustrated in the following.

Decimation $(L/M < 1)$ Multistage implementation (figure 15):

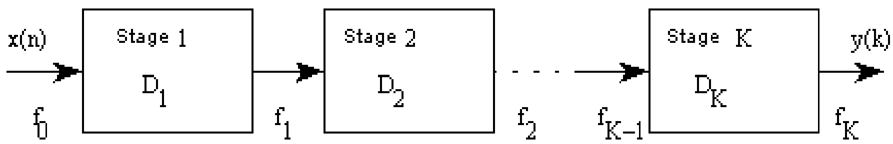


Figure 15: Multistage optimization

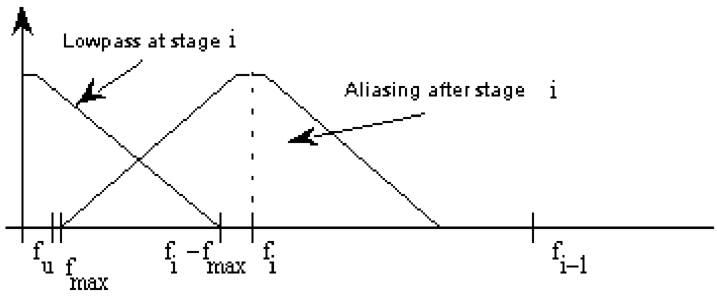


Figure 16: Characteristic of a lowpass filter

Global scheme :

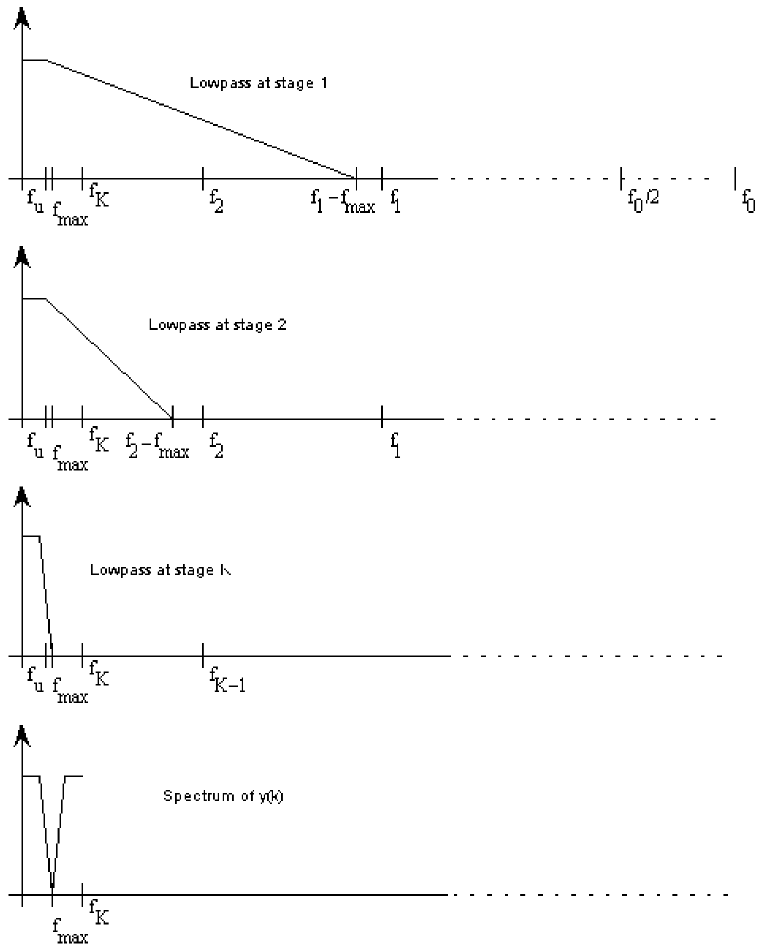


Figure 17: Global scheme for multistage filtering.

Denoting the bandpass ripple as δ_p, for a multistage realization with K stages, each filter should have a bandpass ripple of $\frac{\delta_p}{K}$. The stopband ripple can be kept the same for each filter.

Interpolation $(L/M > 1)$ The same reasoning can be used, it suffices to reverse the order of the filters.

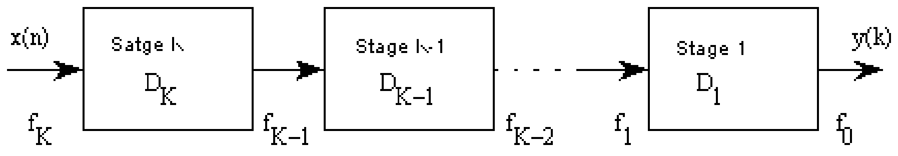


Figure 18: Multistage interpolation

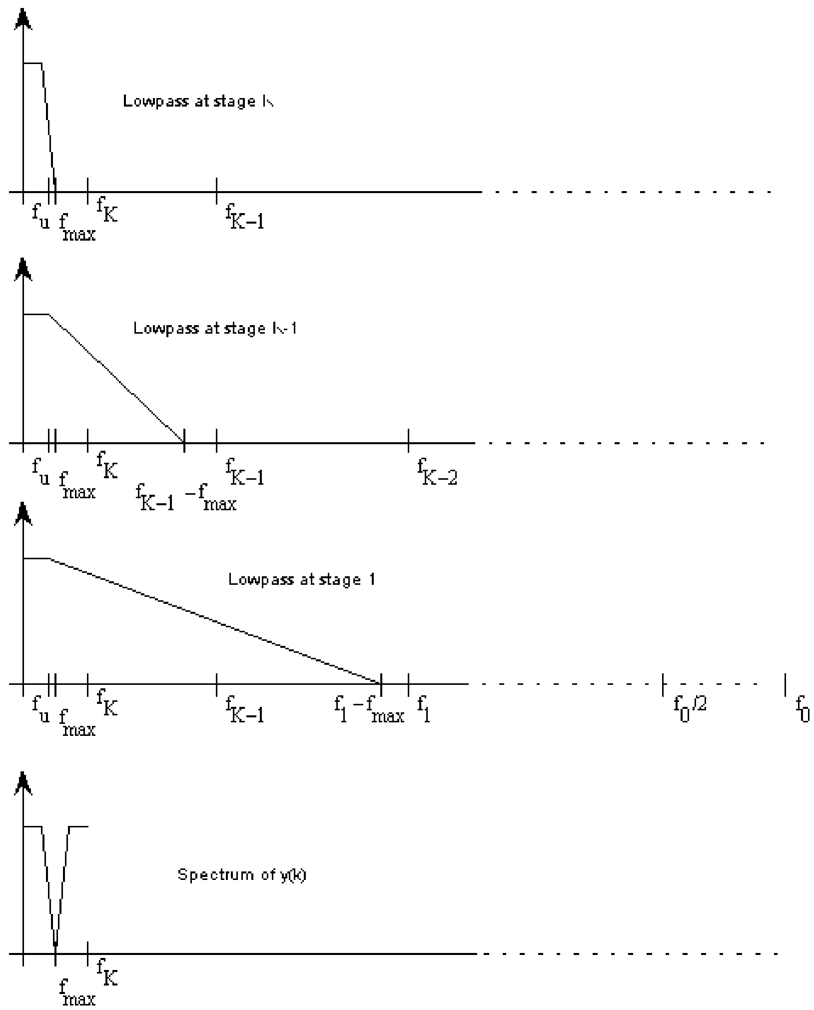


Figure 19: Global scheme for multistage filtering.

2 Example 48 kHz to 12.8 khz sampling rate conversion

Frequency ratio is $\frac{12.8}{48} = \frac{4}{15}$, the conversion can be realized with the following scheme:

Here we have $L = 4$ and $M = 15$, and the cut-off frequency of the filter is chosen has $\min\left(\frac{4*48000}{2L}, \frac{4*48000}{2M}\right) = \min(24000, 6400) = 6400Hz$. This yelds to a low-pass filter with a transition band between $6000Hz$ and $6400Hz$, and a ripple of -100dB according to the fact that 16 bits rise to a dynamic of $96db$ (approximatly $6dB/bit$).

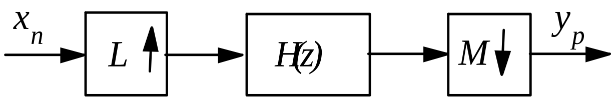


Figure 20: Sample rate conversion.

2.1 Direct implementation

2.1.1 First step

In the first step, decimation is implemented with polyphases of type 1:

$$H(z) = \sum_{n=0}^{N-1} h_n z^{-n} = \sum_{m=0}^{M-1} E_m\left(z^M\right) z^{-m} \tag{1}$$

where: $E_m(z) = \sum_{k=0}^{5L-1} e_m(k) z^{-k}$ et $e_m(k) = h_{kM+m}$.
Using $E_m\left(z^M\right)(M\downarrow) \Longleftrightarrow (M\downarrow) E_m(z)$, this gives the following scheme (fig 21):

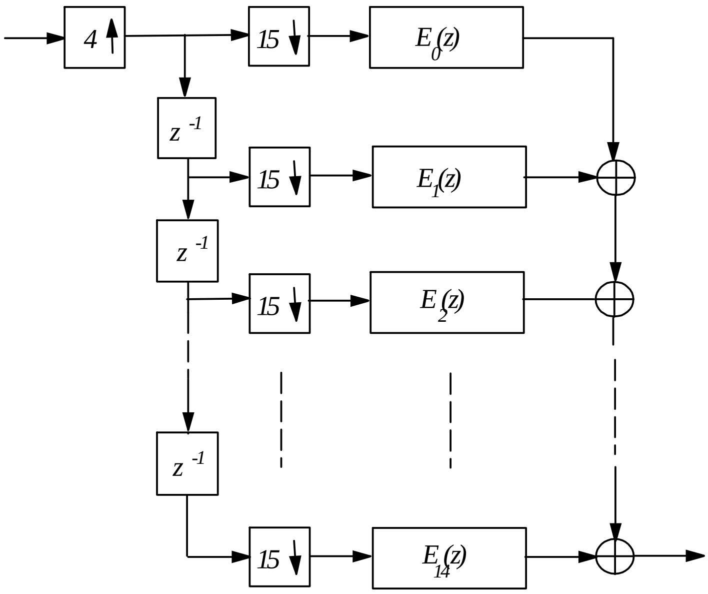


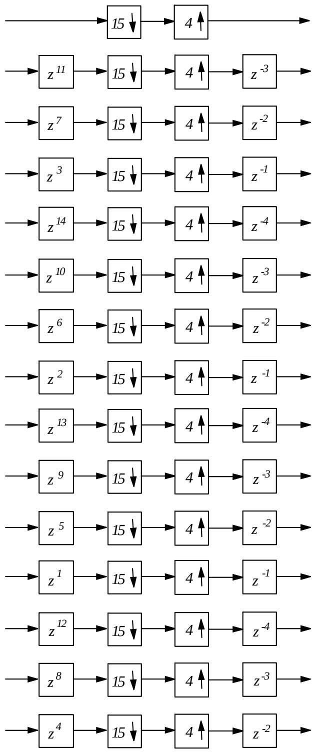
Figure 21: Decimation.

2.1.2 Second step

In order to use the polyphase structure for interpolation, the operators of decimation ($\downarrow$) and interpolation ($\uparrow$) have to be permuted. This is not a problem when they are relatively prime $(4\uparrow)(15\downarrow) = (15\downarrow)(4\uparrow)$. This is the case for channel 0, but in the others channels we have to process the special case $(4\uparrow) z^{-m} (15\downarrow)$. This can be achieve using the following identity : $z^{-m} = z^{4k_1} z^{-15k_2}$. Solutions are given in the following table.

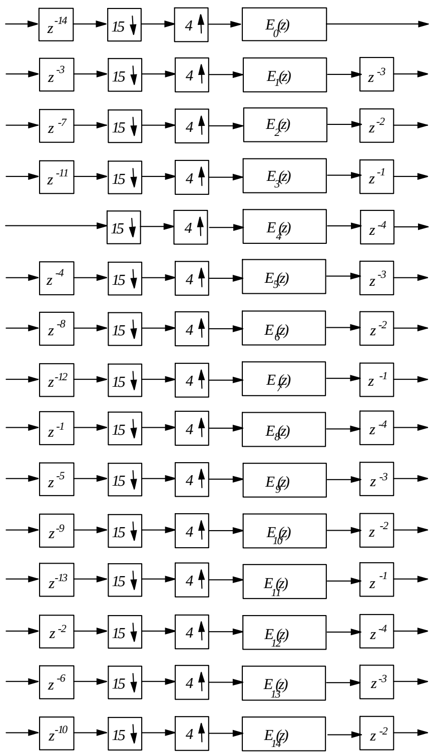
$z^{-1} = z^{44} z^{-45}$	$z^{-8} = z^{52} z^{-60}$
$z^{-2} = z^{28} z^{-30}$	$z^{-9} = z^{36} z^{-45}$
$z^{-3} = z^{12} z^{-15}$	$z^{-10} = z^{20} z^{30}$
$z^{-4} = z^{56} z^{-60}$	$z^{-11} = z^{4} z^{-15}$
$z^{-5} = z^{40} z^{-45}$	$z^{-12} = z^{48} z^{-60}$
$z^{-6} = z^{24} z^{-30}$	$z^{-13} = z^{32} z^{-45}$
$z^{-7} = z^{8} z^{-15}$	$z^{-14} = z^{16} z^{-30}$

Resulting in the following scheme:



Permutations scheme.

In order to obtain causality, it is sufficient to delay all the channels with $z^{-\max} = z^{-14}$.



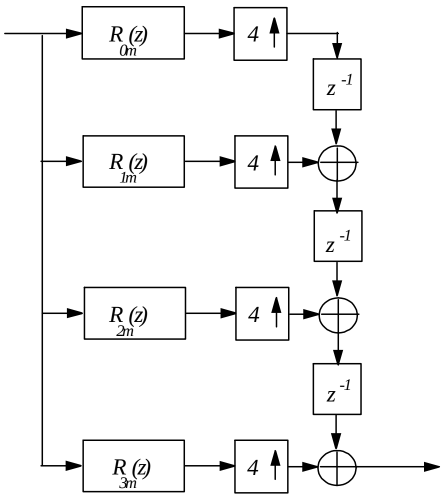
Causality.

2.2 Third step

From now, one can write each ployphse of type 1, using polyphase of type 2:

$$E_m(z) = \sum_{l=0}^{L-1} R_{lm}\left(z^L\right) z^{-(L-1-l)} \tag{2}$$

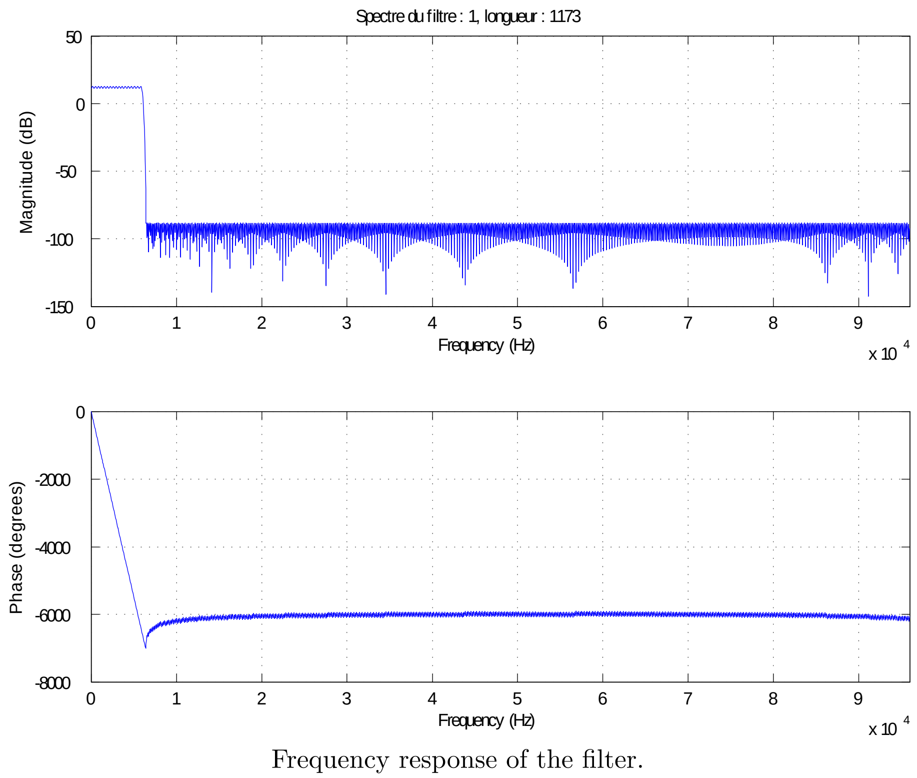
Each $E_m(z)$ is realised using the following scheme:



$E_m(z)$ polyphase of type 2.

2.3 Filter design

For a one stage implementation the filter is defined by (see figure) :
$f_u = 5920\ Hz$, $f_{\max} = 6400\ Hz$ and $length = 1173$



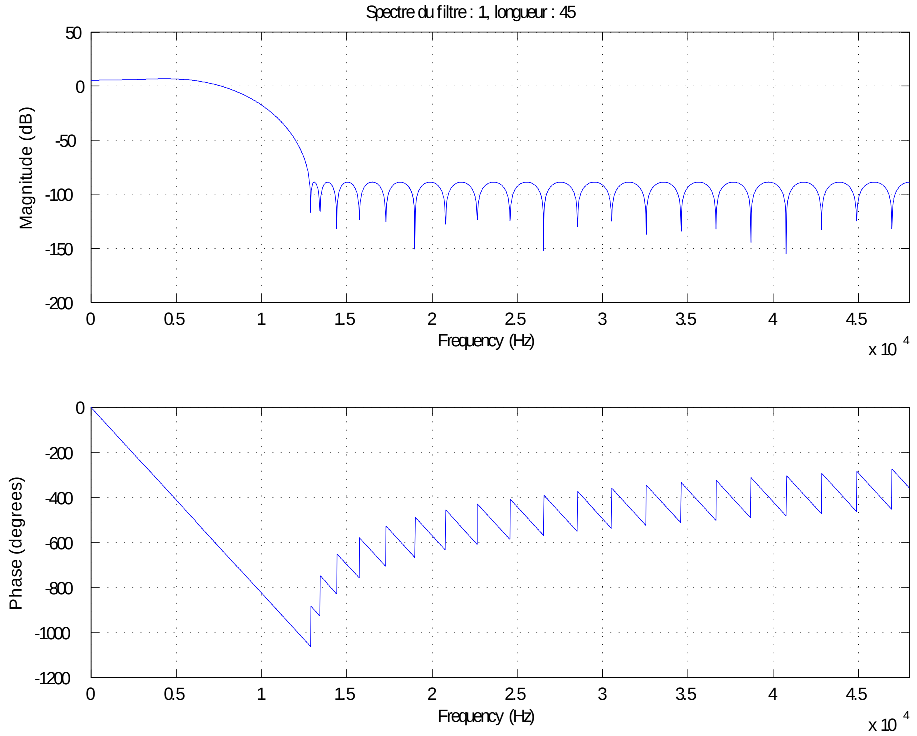
Frequency response of the filter.

2.4 Multi stage implementation

As $\frac{L}{M} = \frac{4}{15}$ can be written as $\frac{L_1}{M_1}\frac{L_2}{M_2} = \frac{2}{5}\frac{2}{3}$, on can use a multi stage implementation:

2.4.1 Stage 1 $\frac{L_1}{M_1} = \frac{2}{5}$

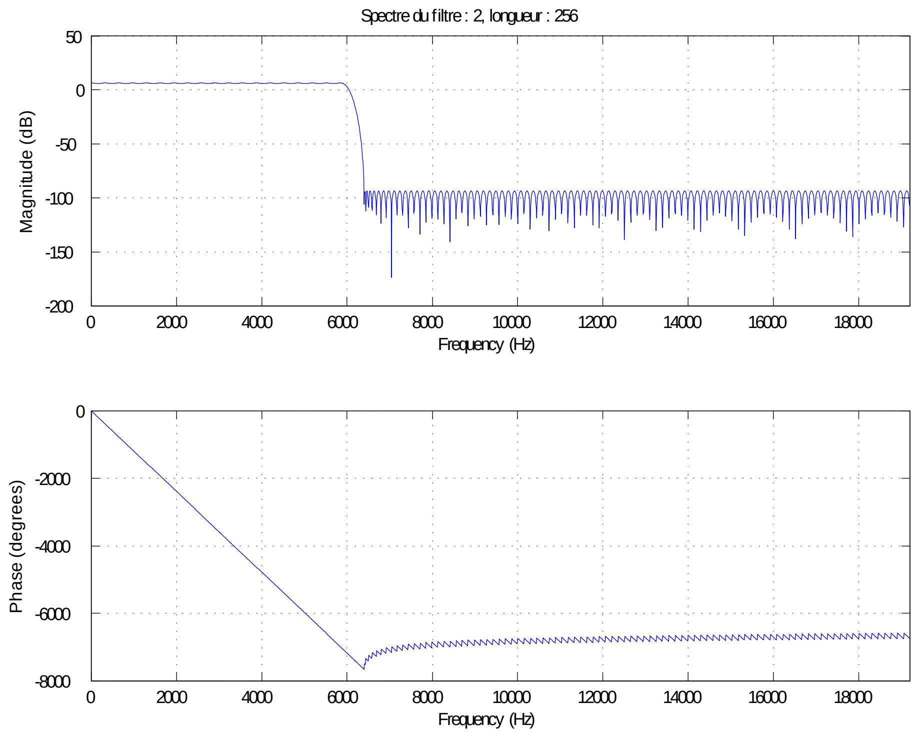
- $f_u = 5920\ Hz$
- $f_1 - f_{\max} = 19200 - 6400 = 12800\ Hz$
- $length = 45$



Stage 1 filter.

2.4.2 Stage 2 $\frac{L_2}{M_2} = \frac{2}{3}$

- $f_u = 5920\ Hz$
- $f_0 - f_{\max} = 12800 - 6400 = 6400\ Hz$
- $length = 256$



Stage 2 filter.

Even if multistage implementation results in the minimum of computation load, the management of the data flow remains quite complicated and generates a non negligible overload. A better approach for a C implementation, is to start by the decomposition of the interpolation.

2.5 New implementation

Considere a ratio $\frac{L}{M}$, where L and M are primes. We use notation of type 1 for the decomposition of type 2. An example for $L = 3$ is given on the following figure:

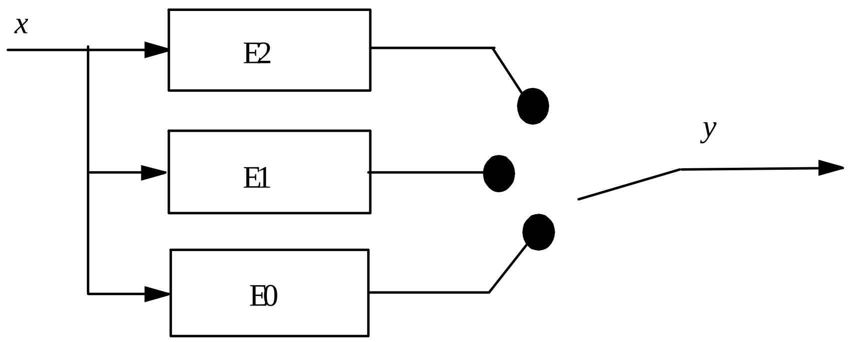


Figure 22: Structure for a C implementation.

At the first time one compute $y_t = filter(E0, x_n)$, then we have to compute $y_{t+1} = filter(Em, x_{n+p})$ and so on. The problem is to find m and p at each step. It's quite easy to verify that starting with $k = 0$, the algorithm is given by:

$$\begin{aligned} k &= k + M \\ p &= floor\left(\frac{k}{L}\right) \\ k &= \mathrm{mod}(k, L) \\ m &= k \end{aligned}$$

This number of operations by sample is the same than in the previous one, but the management of the data-flow is easier.

2.6 Conclusion

The biggest difficulty remains the decomposition of the global ratio L/M, but it's relatively easy to predefine the best decomposition (shortest filters length) for the standard frequencies used in audio. This has been done in *smarc*. For others ratios, the decompostion is automatically computed but the result is often not optimal and a manual help is recommanded, in this way one can manually defined a better decomposition.

Delays of filter operations are automatically compensated and the original signal and converted signal are of same duration. The filters used are equiriple FIR linear phase filters, the algorithm of synthesis is included in *smarc*.

References

[1] R. E. Crochiere and L. R. Rabiner, "Optimum fir digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23, no. 5, Oct 1975.

[2] ——, "Interpolation and decimation of digital signals - a tutorial review," *Proc. IEEE*, vol. 69, no. 3, Mar 1981.

[3] ——, *Multirate Digital Signal Processing.* Prentice-Hall Inc., 1983.

[4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Prentice-Hall Processing Series, 1993.