Acoustics'08 Paris
June 29-July 4, 2008
www.acoustics08-paris.org

Averaged Lagrange Method for interpolation filter

Jonathan Andrea, Frederic Coutard, Patrick Schweitzer and Etienne Tisserand

LIEN - BP 239, Université Henri Poincaré, 54506 Vandoeuvre, France
jonathan.andrea@lien.uhp-nancy.fr

This paper presents a new method for Lagrange interpolation for reducing distortions without introducing any complexity. The aim is to improve the linearity of the phase and the gain responses of the interpolation filter by an averaging method. A first FIR interpolation filter of the second order computes the values between three successive samples of the input signal. A second filter, computes the values intermediate digital values . Finally, the common values provided by the two filters are averaged two by two . This double interpolation can be simply with a Farrow structure filter. Compared to the usual Lagrange's third order interpolation filter, the behaviour of the filter we propose is more regular especially in the high frequencies of the Nyquist band. The results shows that the method significantly reduces distortions.

1 Introduction

In numerous applications such as communications, audio technology, speech coding, time delay estimation and echo canceler in modems, interpolation filters are used to calculate new samples at arbitrary time instants in between existing discrete-time samples. They are required whenever there is a need to change from one sampling rate to another or for timing recovery. This can be achieved by using different methods. The most simple one consists in converting the digital signal to an analog signal and then to convert this analog signal to a digital signal at a different sampling frequency. This corresponds to the more expensive of the existing methods. Other methods are using digital filters like the interpolation filter based on splines functions, interpolation using the discrete cosine transform [1] for zero padding [2] or using the Sinc function [3]. Finally, the method which is providing the most flat response is the Lagrange interpolation method which is often used for digital Fractional Delay (FD) filters [4]. The FD filter provides a useful building block that can be used for fine-tuning the sampling instants [5]. But this method suffers from distortion problems especially at high frequencies. Moreover, the transfer function of Lagrange polynomial based filters or Lagrange FD filters depending on the treated interpolated sample is not constant. So, some signal frequencies which are nearly equal to the Nyquist frequency are removed. In addition, for the same reasons, the phase is not linear.

This paper presents an averaged Lagrange method that can be used to significantly reduce distortion. The dynamic of the variable transfer function is also reduce. As a consequence, the phase becomes more linear and the frequencies nearly equal to the Nyquist frequency are not removed. First, we introduce the Lagrange method in the approach that permits to determine the averaged strategy. Second we show the non-smooth point (discontinuity) phenomenon that causes distortions in the interpolated signal. Third, we present the averaging strategy for reducing these distortions. Finally, the results of simulation are analysed.

2 The Lagrange Interpolation

2.1 Generality

It is possible to interpolate a curve with a few numbers of basepoints using Lagrange's method. Therefore, the polynomial function $P_n(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t^1 + a_0$ is also defined. The coefficients a_n can be calculated according to the following equations:

$$L_m(t) = \prod_{i=0; i \neq m}^{n} \frac{t - t_i}{t_m - t_i} \tag{1}$$

$$P_n(t) = \sum_{m=0}^{n} P_n(t_m) \cdot L_m(t) \tag{2}$$

Where :

- $P_n(t_m)$ are the interpolation basepoints (figure ??).
- n is the order of the polynom.
- $n+1$ is the number of basepoints.
- m is the index of t_m instants ($m \in \{0, 1, ..., n\}$).

2.2 Application to signal processing

To apply Lagrange's method to real time systems, we define t_m so that $t_m - t_{m-1} = T_{in}$. In this case, a basepoint can be considered as an input signal sample x_k. The polynom $P_n(t)$ is the function which passes through the $(n+1)$ input signal samples. We can thus calculate a defined number N of added samples between two input signal samples. We introduce a new time index j which represents the time between two interpolated samples.

$$P_n(j) = \sum_{m=0}^{n} (x_{k-(n-m)} \cdot L_m(j)) \tag{3}$$

$$L_m(j) = \frac{1}{N^n} \prod_{i=0; i \neq m}^{n} \frac{j - iN}{m - i} \tag{4}$$

These equations are directly implementable in a calculator such as Digital Signal Processors or FPGAs. $P_n(j)$ is the recursive equation of a FIR Filter where $L_m(j)$ are the variable coefficients.

In practice, the implementation can be carried out using a classical architecture presented figure 1. The filter coefficients $L_m(j)$ are stored in a ROM (see figure 1).

3 Distortions and Discontinuities

The classical architecture of the RIF filter (figure 1) was tested with Matlab Simulink. We observe distortions in the output signal. These distortions are independant of the way that the filter is implemented. The figure 2 illustrates how the distortions appears :

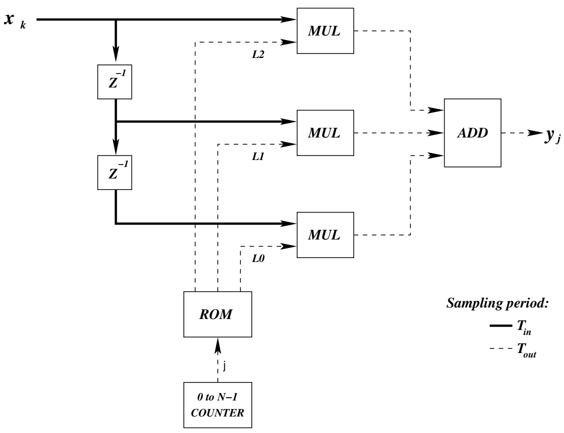

Figure 1: Classical FIR Interpolation Filter

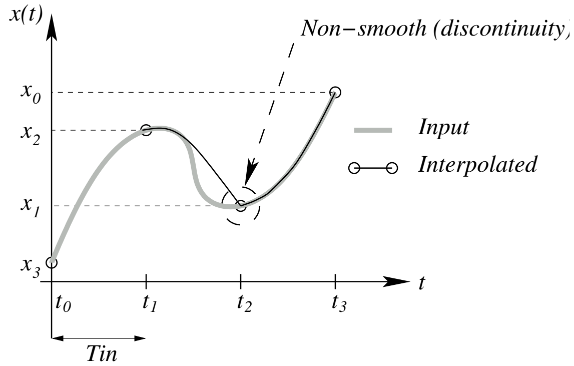

Figure 2: Waveform of the output signal.

The first interpolation is calculated with the basepoints $\{[x_0, t_3], [x_1, t_2], [x_2, t_1]\}$, the second with the basepoints $\{[x_1, t_2], [x_2, t_1], [x_3, t_0]\}$. If this two interpolations are computed succesively to provide the output, then a discontinuity can be created. A discontinuity in the ouput signal is characterised by an input signal inflexion. The discontinuity will be harder if the inflexion is hard as shown figure 2.

4 Averaging Method

4.1 Averaged Lagrange Interpolation for classical FIR filter

The method is to realize, in the same time, two interpolations for each new input sample. The common output values are averaged to provides the final values.

The first interpolation (figure 3) computes the points between (x_0, x_1, x_2). The second interpolation (figure 3) computes computes the points between (x_1, x_2, x_3). The common values beetwen (x_1, x_2) are averaged to obtain the final values.

$$y_j = \sum_{m=0}^{2n-1} (x_{k-(2n-1-m)} \cdot G_m(j)) \qquad (5)$$

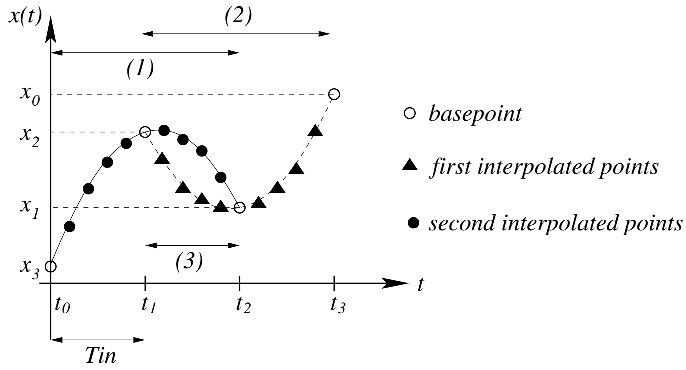

Figure 3: Average method apply to the second order

If $m < n$:

$$G_m(j) = \frac{1}{nN^n} \sum_{p=0}^{n} \left(\prod_{i=0;i\neq m-p}^{n} \frac{j + (n-1-p-i)N}{m-p-i} \right) \qquad (6)$$

Instead, if $m \geq n$ then we use the coefficients symetry by replacing j by $N - j$.

4.2 Averaged Lagrange Interpolation for fractional delay filter

Lagrange interpolation is one of the techniques often used for fractional delay filter. It is usually implemented using a direct-form FIR filter structure. As Farrow shows in [4], it is possible to re-arrange the structure of the filter to obtain is the fractional delay D.

By replacing j by $N \cdot D$ we can rewrites equations (7) (8) :

$$P_n(D) = \sum_{m=0}^{n} (x_{k-(n-m)} \cdot L_m(D)) \qquad (7)$$

$$L_m(D) = \prod_{i=0;i\neq m}^{n} \frac{D-i}{m-i} \qquad (8)$$

When using the averaging method :

$$y_j = \sum_{m=0}^{2n-1} (x_{k-(2n-1-m)} \cdot G_m(D)) \qquad (9)$$

If $m < n$:

$$G_m(D) = \frac{1}{n} \sum_{p=0}^{n} \left(\prod_{i=0;i\neq m-p}^{n} \frac{D+n-1-m}{m-p-i} + 1 \right) \qquad (10)$$

Instead, if $m \geq n$ then we use the coefficients symetry by replacing D by $1 - D$.

The figure 4 presents the averaged second order filter diagram with the followings coefficients

$$z^{-D}Y(z) = C_0(z) + C_1(z) \cdot D + C_2(z) \cdot D^2 \qquad (11)$$

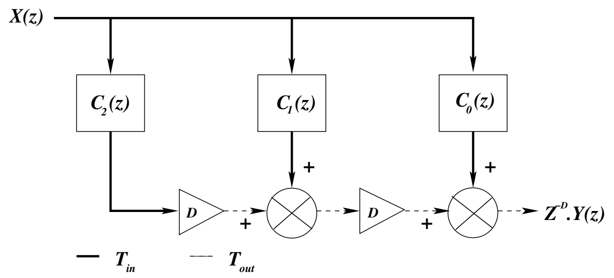

Figure 4: Farrow Interpolation Filter

$$\begin{aligned} C_0(z) &= z^{-2} \\ C_1(z) &= \frac{-1}{4} + \frac{5}{4}z^{-1} - \frac{3}{4}z^{-2} - \frac{1}{4}z^{-3} \\ C_2(z) &= \frac{1}{4} - \frac{1}{4}z^{-1} - \frac{1}{4}z^{-2} + \frac{1}{4}z^{-3} \end{aligned}$$

5 Matlab simulink simulation

The third order non-averaged filter (1) and averaged filter (2) are tested with MATLAB Simulink. The oversampling ratio N is equal to 32. The second filter is implemented using the Farrow structure (figure ??).

5.1 THD estimation

The simulation results are obtain by using a sampled and holded input sine wave. The output is filtered by a notch filter at the fundamental frequency. Then a RMS calculation is made to measure the THD level. We can observe in figure 5 that the averaging interpolator reduces distortions at high frequencies compared to the non-averaging Lagrange filter. However, at low frequencies (figure 5) the non-averaged third order is slightly better than the averaged interpolator (figure 5). But, the third order filter needs a more complex architecture.

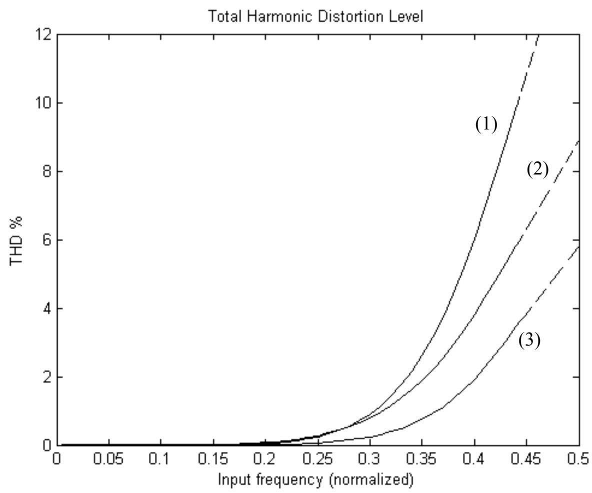

Figure 5: Total Harmonic Distortion versus input signal frequency. (1) is a non-averaging order 3 Lagrange interpolator, (2) is an averaging order 2 Lagrange interpolator.

5.2 Gain and phase estimation

We have compared the non averaged third order filter (1) and the averaged second order filter (2) frequency response between (see figure 6 and 7). By considering the interpolation of part x_{k-1} x_{k-2}, i.e. for $0 < D < 1$ for the filter (1) and for $1 < D < 2$ for the filter (2), we observe the simalary frequency response.

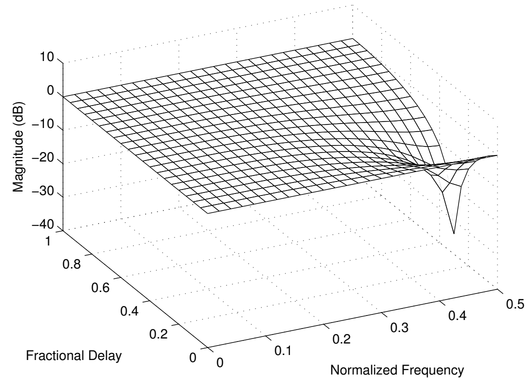

Figure 6: Magnitude response of averaged the second order interpolation.

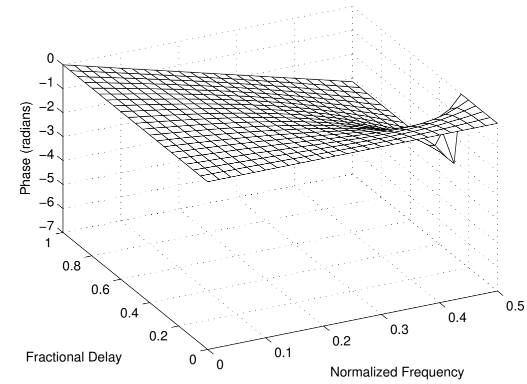

Figure 7: Phase response of averaged the second order interpolation

The both filters present an infinite attenuation at Nyquist frequency for D=1.5 for (1) and D=0.5 for (2). Group Delay diagrams represented are quiet identical.

6 Conclusion

In this paper we present the averaged Lagrange interpolation filter for reducing distortions in the output signal. Our method joins the results provided by two second order interpolation filters in a single filter of the third order. There is two main advantages : the solution can be easily designed for a Farrow fractional delay filter, the third order polynom used is more flat than the usual third order, which explains the improvement of the THD results. Results show that this method presents an indentical frequency response to the third order filter (1). The coefficients are multiple of two and the number of multpliers and adders is reduced. This method will be implemented in a FPGA of altera called cyclone using a simplify architecture. Result of implementation will be presented in a further article.

References

[1] J.I. Agbinya, "Interpolation using the discrete cosine transform," *Electronics Letters,* Vol. 28, No. 20, September. 1992.

[2] G Oetken, T.W. Parks and H.W. Schussler, "New results in the design of digital interpolators," *IEEE Transactions on Acoustics, Speech, and signal Processing,* Vol. ASSP-23,NO. 3, June. 1975.

[3] T. Schanze, "Sinc interpolation of discrete periodic signals," *IEEE Transaction on signal Processing,* Vol. 43, No 46. June. 1995.

[4] C.W. Farrow, "A continuously variable digital delay element," *IEEE International Symposium on Circuits and Systems,* pp. 2641-2645, June. 1988.

[5] D.Luengo, C. J. Pantaleon, J. Ibanez and I. Santamaria, "Design of simultaneous sampling systems based on fractional dealy Lagrange filter," *IEEE Transactions on circuits and systems,* Vol. 47, NO. 5, May. 2000.

[6] Z. Ye, "Linear phase Lagrange interpolation filter using odd number of basepoints," *IEEE International Conference on Acoustics, Speech, and signal Processing,* vol. VI pp. 237-240, April. 2003.

[7] ZA. Haftbaradaran and K. Martin, "An interpolation filter based on spline functions for nonsynchronized timming recovery," *IEEE International Symposium on Circuits and Systems* Vol. IV. pp. 309-312, May. 2004.

[8] C. Dick and F. Harris, "FPGA interpolators using polynomial filters," *The 8th International Conference on signal Processing applications and technologie,* pp. 361-364, September. 1998.

[9] R. El-Kouatly and S.H. El-Ramly, "Improving temperal resolution by using Lagrange interpolation filter," *IEEE International Conference on Acoustics, Speech, and signal Processing,* pp. 361-364, June. 2000.