PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 36, NOS. 3–4, PP. 227–234 (1992)

INTERPOLATION BY IRRATIONAL FACTOR

Zoltán KISS and Ferenc NAGY

Department of Measurement and Instrument Engineering
Technical University of Budapest
H–1521 Budapest, Hungary

Received: May 13, 1992

Abstract

The paper presents an interpolation method that can be used for implementing any sampling rate conversion ratio. The interpolator is basically a resonator-based digital filter (RBDF) with special decoupling that requires the computing of sine/cosine functions at arbitrary arguments. The resulting filter has a finite duration impulse response (FIR), but it cannot be designed by the commonly used Parks–McClellan algorithm because the filter response is a finite Fourier polynomial in the time-domain and not in the frequency-domain. A special windowing technique or linear programming can be used to design such filters. A design example, its simulation as well as a digital signal processor based realization are also presented.

Keywords: interpolation, sampling rate change, resonator-based digital filter (RBDF).

Introduction

Changing the sampling frequency is a common procedure in many areas of digital processing of inherently analog signals. When the interpolation/decimation rate is an integer or a quotient of two relatively small integers the problem is very well treated by using poliphase filter banks. Sometimes we need to change the sampling rates by a virtually irrational factor. This arises for instance when two systems with independent sampling control are connected to each other or we need change the sampling frequency between standardized values in digital audio applications. In these cases the value of the rate change may be a quotient of two large, relatively prime numbers or it is really irrational. Realization of such sampling rate changes is not realistic using conventional techniques. Let us consider a sampling rate conversion between two standard frequencies in digital audio applications, 44.1 kHz and 48 kHz. Their ratio is 147 over 160. The implementation of a sampling rate converter would require a filter-bank of 147 filters for the interpolation. This means that the amount of calculation does not simply increase by the output sampling rate, but is a function of the ratio of the sampling frequencies. To develop a method for implementing any sampling rate change, we have to find a formula that gives the value of the continuous signal from the samples at any given moment. The sampling rate change will then be the evaluation of that formula in the output sampling moments.

The Proposed Method

The following formula gives the value of the continuous signal at any moment:

$$y(t) = \sum_{n=-\infty}^{\infty} x(t_n)h(t-t_n). \tag{1}$$

(1) is a discrete convolution, in other words we can treat the continuous signal as the output of a filter. The ideal impulse response of the filter is:

$$h(t) = \mathrm{sinc}(f_s t) \tag{2}$$

where

$$\begin{aligned} \mathrm{sinc}(x) &= \frac{\sin(\pi x)}{\pi x}, \quad \text{if} \quad x \neq 0 \\ &= 1 \qquad\qquad \text{if} \quad x \neq 0 \end{aligned} \tag{3}$$

Substituting (2) into (1) results the Whittaker Interpolation Formula from the sampling theorem that gives the exact value of the signal for any t moment. On the other hand, this cannot be implemented, because if the impulse response is not time limited, then it would mean an infinite summation. So instead of the ideal impulse response we have to find another, a time limited impulse response, whose value can easily be determined for any time moment. Such a function will have the following form:

$$h(t) = I_T(t) \sum_{m=-M}^{M} H_m e^{j2\pi mt/T}, \tag{4}$$

where

$$\begin{aligned} I_t =&1, \quad \text{if} \quad 0 \leq x \geq 1, \\ =&0, \quad \text{otherwise.} \end{aligned} \tag{5}$$

(4) is one period of a periodic signal that has finite Fourier series. Its Fourier transformation is the transfer function of the filter

$$H(f) = \sum_{m=-M}^{M} H_m T \mathrm{sinc}\, T(f - \frac{m}{T}) e^{j\pi mt}. \tag{6}$$

Let us examine now the interpolating formula using this filter impulse response.

$$y(t) = \sum_{m=-M}^{M} H_m e^{j2\pi mt/T} \sum_{n=-\infty}^{\infty} x(t_n) I_T(t-t_n) e^{-j2\pi mt_n/T}. \qquad (7)$$

The inner summation is a Discrete Fourier Transform of the input samples. This can efficiently be calculated after each input sample, somewhat like a moving window, by a Recursive Discrete Fourier Transformation algorithm. The rest of the calculation is a finite summation that can be calculated from the result of the RDFT. Altogether this is a Resonator Based Digital Filter [2], with a special, time variant decoupling. The RBDF structure is on *Fig. 1.*

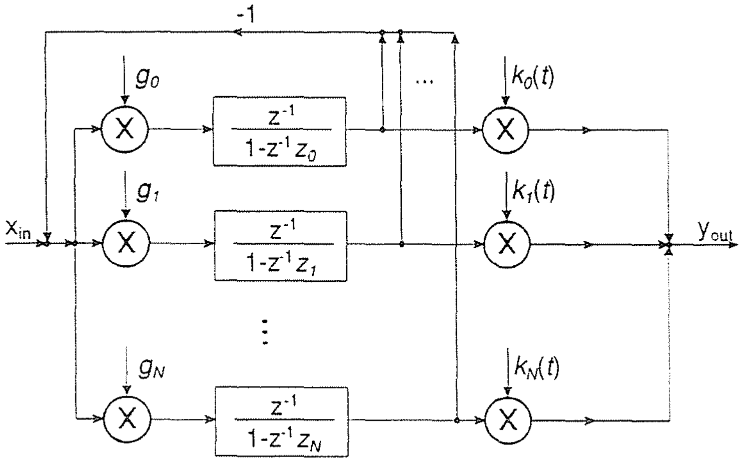
-1
g_0
g_1
g_N
$\frac{z^{-1}}{1-z^{-1}z_0}$
$\frac{z^{-1}}{1-z^{-1}z_1}$
$\frac{z^{-1}}{1-z^{-1}z_N}$
$k_0(t)$
$k_1(t)$
$k_N(t)$
x_{in}
y_{out}

Fig. 1. Resonator Based Digital Filter

Filter Design

Designing the filter means determining the number and value of the H_m coefficients. The efficient Parks–McClellan algorithm cannot be used, because the impulse response given in (4) is not a sum of cosine functions in the frequency domain. We have different possibilities for the design of the filter.

1. We can go back to the basis of filter design, and do a Chebyshev approximation of the desired frequency response. We shall apply linear programming for filter design [3].
2. We can apply windowing to our case. This will give the filter coefficients with very little calculation work. By using this method we introduce some error, but this can be so little, that we will still obtain a good filter.

Linear Programming

We shall approximate the amplitude response only. From (6) we get the filter's amplitude characteristics in the following:

$$H(f) = \sum_{m=-M}^{M} H_m T \mathrm{sinc} T(f - \frac{m}{T}). \tag{8}$$

Our goal is to find such H_m coefficients that minimize the difference between (8) and the ideal low-pass filter characteristics. Minimizing the maximum absolute difference is minimizing the error in the Chebyshev norm. In vector notation: the

$$\mathbf{Ax} = \mathbf{b} \tag{9}$$

linear system is to be solved in Chebyshev norm sense, where

$\mathbf{b}$: desired frequency response on a dense grid
$\mathbf{x}$: vector of coefficients (H_m parameters)
$\mathbf{A}$:$A_{ij} = T\mathrm{sinc}T(f_i - \frac{j}{T})$.

The frequency grid does not have to be equidistant, e.g. it could be a lot denser in the passband than in the transition band. Size of $\mathbf{b}$ can greatly exceed the size of $\mathbf{x}$, so (9) is an overdetermined linear system. Its solution is given by linear programming. We have found an efficient algorithm in the literature [4] that solves the linear programming problem, without having strict conditions on matrix $\mathbf{A}$.

Windowing

From (1) and (2) it is clear that we have to approximate the ideal low-pass filter.

$$\begin{aligned} h(t) &= \mathrm{sincc}(f_s t) \\ H(f) &= \mathrm{rect}(\frac{f}{f_s}) \end{aligned} \tag{10}$$

where

$$\begin{aligned} \mathrm{rect}(x) &= 1 \qquad \text{if} \qquad |x| \le 0.5, \\ &= 0, \qquad \text{otherwise.} \end{aligned} \tag{11}$$

We saw that we need a causal, time limited impulse response.

$$\begin{aligned} h(t) &= I_T(t)\mathrm{sinc}(f_s(t - T/2)), \\ H(f) &= \mathrm{sinc}(Tf) * \mathrm{rect}(\frac{f}{f_s})e^{j\pi T}. \end{aligned} \tag{12}$$

As the result of the cut in time domain, the well-known Gibbs oscillation occurs in frequency domain. This weakens the quality of the designed filter. To achieve higher stop band signal suppression, multiply (10) instead of $I_T(t)$ with a window function. The chosen window is a min-3-term window [5]. The result of windowing can be compared to the original filter in *Fig. 2.*

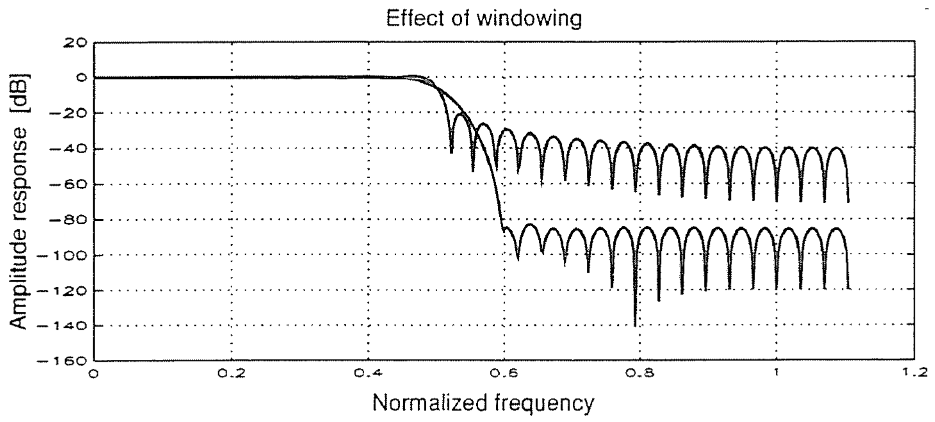

Fig. 2. Filter design — The effect of windowing

As a final step, we have to modify the cutoff frequency to $f_s/2$.

$$\begin{aligned} h(t) &= w(t)\mathrm{sinc}(0.8f_s(t - t/2)), \\ H(f) &= W(Tf) * \mathrm{rect}(\frac{f}{0.8f_s})e^{j\pi T} \cdot \end{aligned} \tag{13}$$

The filter's H_m coefficients are samples of the transfer function given in (13). The impulse response is time limited, so $H(f)$ can be sampled at k/T frequency points.

The designed filter has a transition band from $0.3f_s$ to $0.5f_s$. The width of the transition band equals to the width of the main lobe of the window function, in the case of the min-3-term window used it is $6/T$. The transfer function has to be sampled only below $f_s/2$, since over it the coefficients would almost be equal to zero. (We introduce the error mentioned at the beginning of this chapter because of the following: The zeroes of the transfer function in (6) and (13) do not equal, they are only very close, and get closer, as we go towards higher frequencies. This is why the samples taken over $f_s/2$ are not zeros, only very small. But the error we make here is not significant, in *Fig. 3* we can see that the stop band signal suppression of the designed filter is better than 80 dB.)

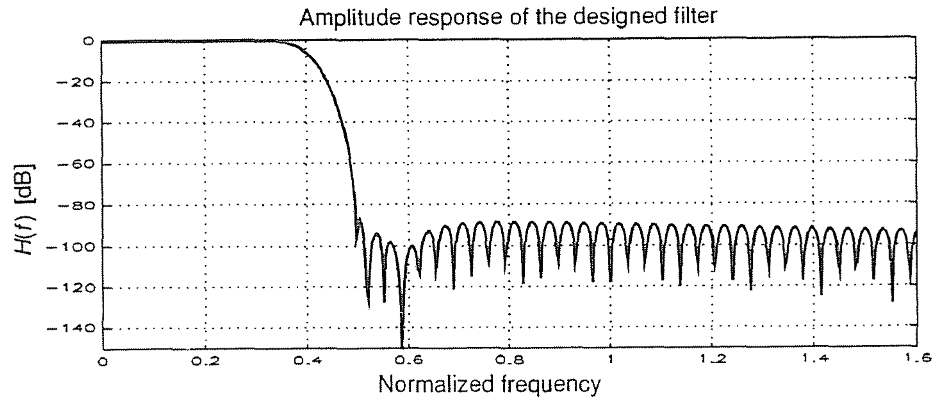

Fig. 3. The designed filter

Example

After the design of the filter a simulation was made to implement this interpolation method. As well as the filter design this simulation runs on a PC. As mentioned earlier, the calculation is divided into two stages:

RDFT and decoupling. Whenever an input sample is taken an RDFT is calculated, and every output sample is a result of a decoupling.

The simulator can calculate any interpolation ratio. In *Fig. 4* we can see the result of an interpolation by a factor of π. (Since the filter has a constant delay, the input sample is delayed on the plot to ease the comparison.) From the results we can see two properties of the filter: starting from zeroed state, the output of the filter is also zero, and then it reaches the value of the input signal. After time moment $t = T$, where T is the length of the impulse response, the error of the interpolation gets under a level, what is determined by the designed filter. (10^{-4} in the case of the used filter.)

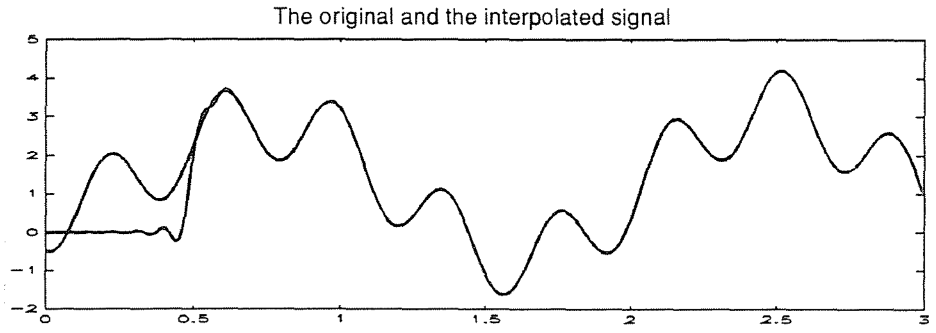

Fig. 4. Input and output of the interpolator

The interpolator was also implemented on a general purpose digital signal processing board. Due to the real time conditions, unlike for the simulation, certain restrictions are applied for the input and output sampling frequencies. Obviously they cannot be too high for the computing power of the processor. In this implementation, where input and output events are serviced by the processor, these events must not get very close. This limits the ratio of the interpolation. However, this limit can be extended by using some additional hardware, something like a pipeline, for the interrupt servicing.

Conclusions

The presented interpolation method can be used to implement any frequency rate change in off-line conditions. The real-time conditions introduce some limitations to the possible use of this method. The first one is that this method requires a large amount of calculation work, so it should only be used when traditional interpolating methods fail. The second limitation arose from the nature of our DSP implementation, namely from the software handling of sampling interrupt events. As stated earlier, this limitation could be removed by using some additional hardware on a DSP board. Then such an implementation will be able to realize a very wide range of sampling rate conversions, with easy reconfiguration by software.

References

1. CROCHIERE, R. E. – RABINER, L. R.: Multirate Digital Signal Processing, Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1983, pp. 22–42, 175–178.
2. PÉCELI, G.: 'Common Structure for Recursive Discrete Transformation,' *IEEE Transactions on Circuits and Systems*, Vol. CAS 33, October 1986, pp. 1035–1036.
3. RABINER, L. R.: 'The Design of Finite Impulse Response Digital Filters Using Linear Programming Techniques', *The Bell System Technical Journal*, Vol. 51, No. 6, pp 1177–1198, July-August 1972.
4. BARRODALE, I. – PHILLIPS, C.: 'ALGORITHM 495 Solution of an Overdetermined System of Linear Equations in the Chebyshev Norm,' *ACM Transactions on Mathematical Software*, Vol. 1, No. 3, September 1975, pp. 264–270.
5. NUTTAL, A. H.: 'Some Windows with Very Good Sidelobe Behaviour,' *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP–29, No. 1, February 1981.