# The effects of quantizing the fractional interval in interpolation filters

**Article** · January 2000



**4 authors**, including:

Tapio Saramaki
Tampere University


Markku Renfors
Tampere University

# THE EFFECTS OF QUANTIZING THE FRACTIONAL INTERVAL IN INTERPOLATION FILTERS

*Jussi Vesma, Francisco Lopez, Tapio Saramäki, and Markku Renfors*

Tampere University of Technology, Telecommunications Laboratory
P.O. Box 553, FIN-33101 Tampere, Finland
E-mail: jussi.vesma@.tut.fi

**ABSTRACT**

The fractional interval, which is an adjustable continuous-valued parameter in interpolation filters, is used to determine the interval between the output sample and the previous input sample. This paper presents the effects of quantizing the fractional interval. It is shown that this quantization causes images to the stopband of the filter. These images are reduced by 6dB per every extra bit.

## 1. INTRODUCTION

Interpolation filters (or interpolators) are used in several digital signal processing applications to evaluate sample values of a discrete-time signal at arbitrary points between the existing samples. These applications include, e.g., echo cancellation in modems [1], symbol synchronization in digital receivers [2], and arbitrary sampling rate conversion [3].

This paper concentrates on the interpolation filters having one continuous-valued input parameter $\mu_l$ which is called as a fractional interval. This parameter is used to determine the time interval between the interpolated output sample $y(l)$ and the previous input sample $x(n_l)$. In most of the applications it is required that the fractional interval is adjustable during the computation.

One alternative to implement interpolation filters is to design several fractional delay (FD) FIR filters having a different delay value. These filter coefficients are stored into a lookup table. A new coefficient set can then be reloaded into the filter structure every time when the fractional interval is changed. The size of the lookup table is determined by the length of the filter and the resolution of the fractional interval.

Another alternative is based on the analog model of the interpolation filter. In this model the continuous-time signal is reconstructed using a D/A-converter and reconstruction filter [3]. The interpolated sample values are obtained by resampling this continuous-time signal. When designing interpolation filters using the analog model, the problem is to find the continuous-time impulse response of the reconstruction filter and, then to implement the overall system digitally. The impulse response of the reconstruction filter can be determined by utilizing, e.g., the Lagrange [2], B-spline [4], trigonometric [5], sinc [6], and generalized polynomial-based interpolation methods [4], [7].

One important question when designing interpolation filters is: What is the proper number of bits $B$ for representing the fractional interval? This is especially important for interpolators based on the FD filters, because the size of the lookup table is determined by $B$.

In this paper, the effects of quantizing the fractional interval is presented. It is shown that the quantization causes certain quantization images to the frequency response of the interpolation filter. These images decrease 6dB per every added bit. When $B$ is fixed, one solution to reduce this effect is to increase the sampling rate before the interpolator or to use oversampling at the output of the interpolator followed by a fixed decimation.

## 2. INTERPOLATION FILTERS

Figure 1 shows a block diagram for the interpolation filter, whereas the interpolation process in the time domain is illustrated in Fig. 2. The time instants for the output samples $y(l) = y(lT_{out})$ are determined by

$$lT_{out} = (n_l + \mu_l)T_{in}, \tag{1}$$

where $n_l$ is any integer, $\mu_l \in [0,1)$ is the fractional interval, and $T_{in} = 1/F_{in}$ is the input sampling interval. After knowing the two control parameters $n_l$ and $\mu_l$, which can be calculated using $T_{in}$ and $T_{out}$ (see, e.g., [4]), the role of the interpolation filter is to use the existing discrete-time samples before and after $x(n_l)$ to form the $l$th output sample value. The output sample can be given by the following convolution:

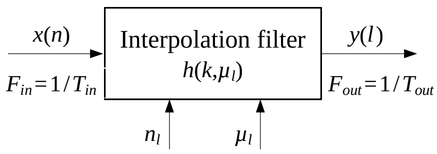


Fig. 1. Interpolation filter with the input signal $x(n)$ and the interpolated output samples $y(l)$. The time instant for the output sample is determined by $lT_{out} = (n_l+\mu_l)T_{in}$.

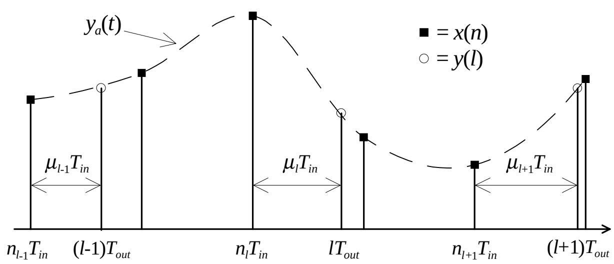


Fig. 2. Interpolation in the time domain. The approximating continuous-time signal $y_a(t)$ (dashed line) is sampled at $lT_{out} = (n_l+\mu_l)T_{in}$ to obtain $y(l)$.

$$y(l) = \sum_{k=-N/2}^{N/2-1} h(k, \mu_l)\, x(n_l - k), \qquad (2)$$

where $N$ is the length and $h(k, \mu_l)$ is the impulse response of the interpolation filter. This impulse response depends on the fractional interval $\mu_l$ and it is therefore time-varying.

A widely used solution to avoid the use of the time-varying impulse response $h(k, \mu_l)$ in analyzing interpolation filters is to utilize the analog model shown in Fig. 3 [3]. The interpolated output sample $y(l)$ is obtained by sampling the reconstructed signal $y_a(t)$ at $t = lT_{out}$ and, therefore, it is given by

$$y(l) = y_a(lT_{out}) = \sum_{k=-N/2}^{N/2-1} h_a((\mu_l + k)T_{in})\; x(\, n_l - k), \qquad (3)$$

where $NT_{in}$ is the length of the non-causal reconstruction filter $h_a(t)$.

By comparing Eqs. (2) and (3), it can be concluded that the continuous-time impulse response of the reconstruction filter $h_a(t)$ and the discrete-time impulse response of the interpolation filter $h(k, \mu_l)$ are related to each other as follows:

$$h(k, \mu_l) = h_a((\mu_l + k)\, T_{in}) \qquad (4)$$

for $k = -N/2, -N/2+1, \cdots, N/2-1$. The result of Eq. (4) suggests that discrete-time interpolation filters have an underlying continuous-time impulse response $h_a(t)$ which can be used to analyze and synthesize these filters.

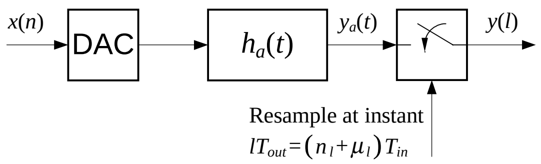


Fig. 3. The analog model for the interpolation filter.

There are basically two types of design methods for interpolation filters. The first is to utilize the analog model to find the impulse response $h_a(t)$ and then to use Eqs. (2) and (4) for deriving the digital implementation structure. Among these, the polynomial-based interpolation filters, which include, e.g., the Lagrange [2], B-spline [4] and generalized polynomial-based interpolation filters [4], [7], have turned out to be good choices, because they can be efficiently implemented using the Farrow structure [1].

The second alternative is to make the design directly in the discrete-time domain. That is, to design several ($M$) FD filters $h(k, \mu_m)$ for the delay values of $\mu_m = m/M$ for $m = 0, 1, \cdots, M-1$. These filter coefficients are then stored into a lookup table and the table is addressed by the desired value of the fractional interval. Since the impulse response of the FD filter having the delay of $\mu_m$ is the time reversed version of the filter having the delay of $1-\mu_m$, the number of stored filter coefficients is $NM/2$, where $N$ is the length of the filters [6].

In the next section, the effect of quantizing the fractional interval is considered. This analysis is especially important for interpolation filters based on the FD filters, because the number of quantization levels directly determines the size of the lookup table. The following derivation is based on the analog model of Fig. 3. However, it can be also used for FD filters, because the impulse response of the reconstruction filter $h_a(t)$ can be approximated by the impulse response of the FD filters $h(k, \mu_l)$ according to Eq. (4).

## 3. THE EFFECTS OF QUANTIZING THE FRACTIONAL INTERVAL

If the fractional interval $\mu_l$ is quantized in the analog model of Fig. 3, the output samples can be given by

$$\hat{y}(l) = \sum_{k=-N/2}^{N/2-1} h_a((\hat{\mu}_l + k)T_{in})\; x(\, n_l - k), \qquad (5)$$

where $\hat{\mu}_l$ is the quantized fractional interval. It is assumed that $\hat{\mu}_l$ has $K = 2^B$ uniformly spaced quantization levels, where $B$ is the number of bits.

It can be seen from Eq. (5) that the quantization has an effect on the impulse response of the interpolation filter, i.e., the unquantized response $h_a((\mu_l + k)T_{in})$ is replaced by the quantized response $h_a((\hat{\mu}_l + k)T_{in})$. An example of these impulse responses for a polynomial-based interpolation filter is shown in Fig. 4 for $B = 3$.

The filter in this example is designed using the minimax optimization method [7] with the following parameters: length of the filter $N = 8$, degree of the polynomials $L = 5$, passband and stopband edges $f_p = 0.35F_{in}$ and $f_s = 0.65F_{in}$, passband ripple 0.018, and stopband attenuation 55dB. The frequency response of this filter for unquantized $\mu_l$ is depicted in Fig. 5.

As can be seen from Fig. 4, the impulse response of the quantized filter is obtained first by sampling the impulse response of the unquantized filter with the sampling interval of $T_{in}/K$ and then by using the zero-order hold (ZOH) to reconstruct the continuous-time response.

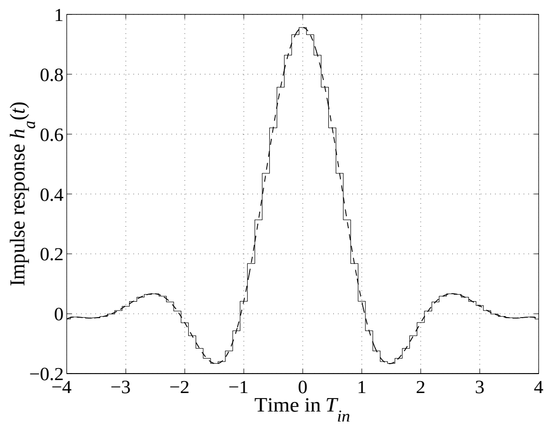


Fig. 4. An example of the impulse response of the interpolation filter for unquantized (dashed line) and quantized (solid line) fractional interval. The number of quantization levels is $B = 3$.

The effect of quantization in the frequency domain is shown in Fig. 6 for $B = 3$. The frequency response of the sampled filter has images at the frequencies $KF_{in}$ and its multiplies. When these images are attenuated by the ZOH, we end up with the frequency response of the quantized interpolation filter.

The quantization of the fractional interval has three different effects in the frequency domain. First, the ZOH causes some distortion/attenuation to the passband of the interpolation filter. This, however, does not usually need any compensations in other filtering stages. Second, there is some aliasing when the unquantized impulse response is sampled. This effect is not either significant if $K$ is sufficiently large and the stopband of the unquantized filter decays fast. The third effect is the most significant and it is caused by the attenuated images in the vicinity of the frequencies $nKF_{in}$ for $n = 1, 2, 3, \cdots$. These high frequency components, which are called as quantization images, may fold into the baseband when the reconstructed signal $y_a(t)$ is sampled to get the output samples $y(l)$ (see Fig. 3).

The level of these quantization images are mainly affected by the number of bits, i.e., by $K = 2^B$ and the passband edge of the interpolation filter. The first quantization image has the highest value. For the example filter, the maximum value is –26.8 dB at $f = 7.63F_{in}$.

The maximum value of the quantization images denoted by $\hat{I}_{max}$ can be approximated by

$$\hat{I}_{max} \approx \mathrm{sinc}\left(1 - \frac{f_p}{KF_{in}}\right), \tag{6}$$

where $f_p$ is the passband edge in Hz. This maximum value as a function of number of bits $B$ is shown in Fig. 7 for three different values of $f_p$. As can be seen $\hat{I}_{max}$ decays approximately 6dB per bit. In order to have the quantization images below the stopband attenuation level (55dB) in the example filter mentioned above, the number of bits for $\hat{\mu}_l$ should be at least $B = 8$ ($K = 256$).

Equation (6) can be used for any lowpass type interpolation filter having a well defined passband edge. For instance, the Lagrange interpolation filters do not have a passband edge. The maximum value of the quantization images can be, however, obtained by using the same kind of analysis as in Fig. 6.

## 4. EXAMPLES

This example illustrates the effect of quantization when the interpolation filter is used for sampling rate conversion between $F_{in} = 44.1$kHz and $F_{out} = 48$kHz. The input signal consists of two sinusoidals at frequencies $0.01F_{in}$ and $0.35F_{in}$. The interpolation filter is the same as in the previous example.

The spectrum of the reconstructed signal $y_a(t)$ for $B = 3$ is shown in Fig. 8(a). The quantization images cause high frequency components to the reconstructed signal. The maximum value of these components is –27dB at $7.65F_{in}$. When $y_a(t)$ is resampled at the rate of

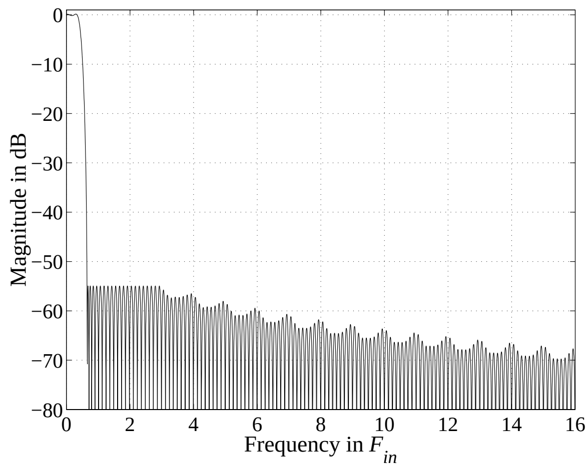


Fig. 5. The frequency response of the example interpolation filter when $\mu_l$ is not quantized.

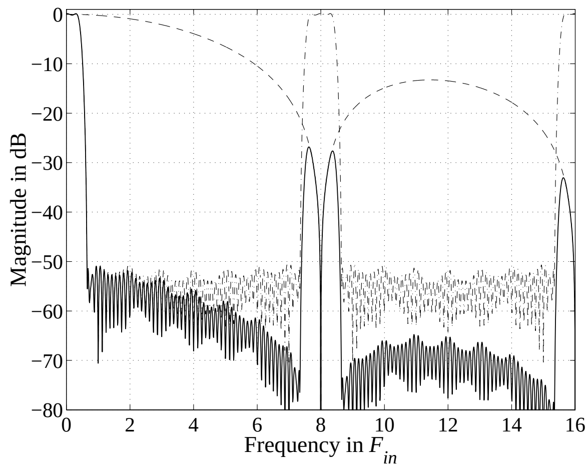


Fig. 6. The frequency response of the example interpolation filter when $\mu_l$ is quantized for $B = 3$ (solid line). This response is obtained by using the sampled version of the unquantized interpolation filter (dash-dotted line) and ZOH reconstruction (dashed line).

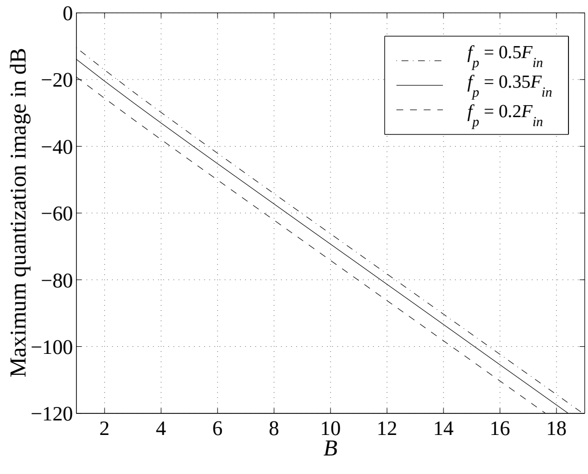


Fig. 7. The maximum value of the quantization images as a function of number of bits $B$ for $f_p / F_{in}$ = 0.5, 0.35, and 0.2.

$F_{out} = 48/44.1F_{in}$ to obtain the output signal $y(l)$, the high frequency components fold to the based band. This can be seen from the spectrum of the output signal shown in Fig. 8(b). When $B = 8$, these aliased components are attenuated at least by 55dB as shown in Fig. 8(c).

If the number of bits $B$ is fixed, let's say to 6 bits, the maximum quantization image is –45.2dB (see Fig. 7). This is 10dB more that the stopband gain. One way to avoid this problem is to use the interpolation filter for conversion between $F_{in}$ and $RF_{out}$. By properly choosing the oversampling factor $R$, the highest quantization images fold outsize the baseband. The final output sampling rate $F_{out}$ is then obtained by using a fixed decimation stage, which also attenuates the out-of-band quantization images. The spectrum of the over-sampled signal at the output of the interpolator for $B = 6$ and $R = 4$ is shown in Fig. 8(d).

Another solution is to increase the sampling rate before the interpolation filter. Every doubling the sampling rate reduces the quantization images by 6dB. Higher sapling rate means also relaxed filter requirements for the interpolator.

## 5. CONCLUSIONS

The frequency-domain effects of the quantization of the fractional interval was studied. It was shown that the quantization causes images to the stopband of the interpolation filter. The level of these images can be approximated by the well-known 6dB/bit-rule.

Because the analysis of the finite word length effects was based on the general analog model, it is valid for any interpolation filter regardless of the implementation structure.

## REFERENCES


[1] C. W. Farrow, “A continuously variable digital delay element,” in *Proc. IEEE Int. Symp. on Circuits and Systems*, Espoo, Finland, June 1988, pp. 2641–2645.

[2] L. Erup, F. M. Gardner, and R. A. Harris, “Interpolation in digital modems–Part II: Implementation and performance,” *IEEE Trans. Commun*., vol. 41, pp. 998–1008, June 1993.

[3] T. A. Ramstad, “Digital methods for conversion between arbitrary sampling frequencies,” *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP–32, pp. 577–591, June 1984.

[4] J. Vesma, *Optimization and Applications of Polynomial-Based Interpolation Filters*. Dr. Tech. Thesis, Tampere, Finland: Tampere University of Tech., Department of Information Technology, May 1999.

[5] D. Fu and A. N. Willson, Jr., “Interpolation in timing recovery using a trigonometric polynomial and its implementation,” in *IEEE Globecom 1998 Communications Mini* Conference Record, Sydney, Australia, Nov. 1998, pp. 173–178.

[6] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, “Splitting the unit delay,” *IEEE Signal Processing Magazine*, vol. 13, pp. 30–60, Jan. 1996.

[7] T. Saramäki and J. Vesma, “Polynomial-based interpolation for digital signal processing applications–A tutorial review,” in *Proc. Computer Science & Information Technologies Conf.*, Yerevan, Armenia, Aug. 1999, pp.120-129.


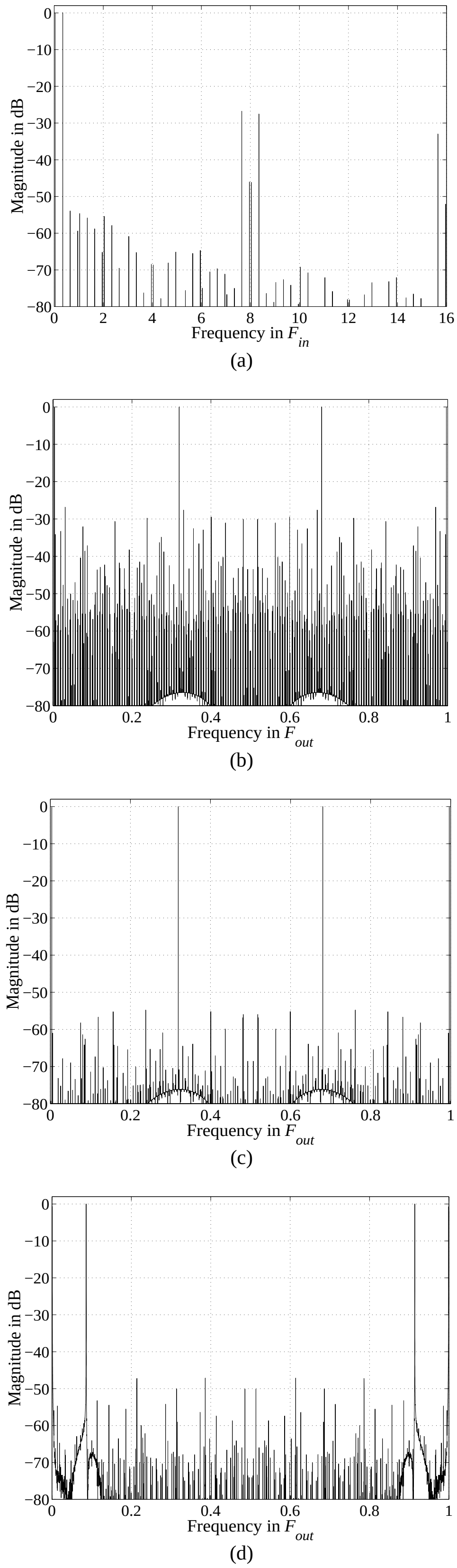


Fig. 8. (a) The spectrum of the reconstructed signal for $B = 3$. (b) The spectrum of the output of the interpolator for $B = 3$ and (c) for $B = 8$. (d) The oversampled spectrum of the output of the interpolator for $B = 6$ and for the oversampling factor $R = 4$.