SAMPLING THEORY IN SIGNAL AND IMAGE PROCESSING Vol. 10, No. 1-2, 2011, pp.17-35

ISSN: 1530-6429

Non-Uniform Filter Interpolation in the Frequency Domain

Brigitte Bidégaray-Fesquet
CNRS, Université Joseph Fourier, Grenoble INP, Université Pierre Mendès-France
Laboratoire Jean Kuntzmann
B.P. 53, 38041 Grenoble Cedex 9, France
Brigitte.Bidegaray@imag.fr

Laurent Fesquet
CNRS, Grenoble INP, Université Joseph Fourier
Techniques de l'Informatique et de la Microélectronique
pour l'Architecture des systèmes intégrés
46 avenue Felix-Viallet, 38031 Grenoble Cedex, France
Laurent.Fesquet@imag.fr

Abstract

We propose a filtering technique which takes advantage of a specific non-uniform sampling scheme which allows the capture of a very low number of samples for both the signal and the filter transfer function. This approach leads to a summation formula which plays the same role as the discrete convolution for usual FIR filters. Here the formula is much more complicated but it can easily be implemented and the evaluation of these more elaborate expressions is compensated by the very low number of samples to process.

Keywords and phrases: non-uniform sampling and processing

2000 AMS Mathematics Subject Classification — 94A08, 84A20, 33B10

1 Introduction

Reducing the power consumption of mobile systems – such as cell phones, sensor networks and many others electronic devices – by one to two orders of magnitude is extremely challenging but will be very useful to increase the system autonomy and reduce the equipment size and weight. In order to reach such a goal, the signal processing theory and the associated system architectures have to be rethought. This paper is a first step towards well suited signal processing techniques for non-uniform sampled signals [10].

Today signal processing systems uniformly sample analog signals (at Nyquist rate) without taking advantage of their intrinsic properties. For instance, temperature, pressure, electro-cardiograms, speech signals significantly vary only during short moments. Thus the digitizing system part is highly constrained due to the Shannon theory, which fixes the sampling frequency at least twice the input signal frequency bandwidth. It has been proved in [9] and [12] that Analog-to-Digital Converters (ADCs) using a non equi-repartition in time of samples lead to interesting power savings compared to Nyquist ADCs. A new class of ADCs called A-ADCs (for Asynchronous ADCs) based on level-crossing sampling (which produces non-uniform samples in time) [2, 3] and related signal processing techniques [1, 11] have been developed.

This work suggests an important change in the filter design. Like analog signals which are usually performed uniformly in time, filter transfer function are also usually sampled with a constant frequency step. Non-uniform sampling leads to an important reduction of the weight-function coefficients. Combined with a non-uniform level-crossing sampling technique performed by an A-ADC, this approach drastically reduces the computation load by minimizing the number of samples and operations, even if they are more complex.

2 Principle and Notations

For a large class of signals, especially sporadic signals, non-uniform sampling leads to a reduced number of samples, compared to a Nyquist sampling. This feature has already been used in [1] to design non-uniform FIR filtering techniques based on interpolation and in [7] for IIR filtering in the state representation. In these works the authors however used a classical (uniform) filter, that is a usual discretization in time of the impulse response.

Here we want to go further and take advantage of the fact that the filter transfer function evolves gently with respect to frequency. It can therefore be well approximated by the linear interpolation of quite few samples.

We consider the analog signal $s(t)$ in the time domain and the analog filter transfer function $H(\omega)$ in the frequency domain. Then the result $x(t)$ of the filtering process is the convolution of $s(t)$ with the impulse response $h(t)$ which is the inverse Fourier transform of $H(\omega)$:

$$\begin{aligned} x(t) &= \int_{-\infty}^{+\infty} h(t-\tau)s(\tau)d\tau, \\ h(t) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} H(\omega)e^{i\omega t}d\omega. \end{aligned}$$

In this paper we will only deal with the interpolation of causal filters. Although the sampling and interpolation procedures do not commute with impos-

ing causality, this is approximatively true and we will include the concept of causality here by changing the formula for $x(t)$ into

$$x(t) = \int_{-\infty}^{t} h(t-\tau)s(\tau)d\tau,$$

in order to use only past input samples to compute the output filtered signal. The general formula is

$$x(t) = \int_{-\infty}^{T(t)} h(t-\tau)s(\tau)d\tau,$$

where $T(t) = t$ if causality is taken into account, and $T(t) = \infty$ otherwise. A deeper analysis of the impact of causality will be part of the subject of an other paper dealing with non uniform filter design.

2.1 Level crossing sampling

These signals are sampled in their initial domain thanks to a level crossing technique (which has to be adapted in the case of the filter transfer function which is a complex valued function: level crossing for the amplitude, for the phase, for both, see Discussion Section 4).

2.1.1 Signal sampling

For the signal, this yields N samples $(s_n, \delta t_n)_{n=1,\dots,N}$, where s_n is the amplitude of the n^{th} sample and δt_n is the time delay elapsed since the previous sample (see Figure 1(a)). We can compute the times of the samples from the interval lengths with the formula $t_n = t_0 + \sum_{n'=1}^{n} \delta t_{n'}$.

2.1.2 Filter sampling

We suppose we have a symmetric filter with respect to frequency $\omega_0 = 0$, i.e. a filter with real-valued impulse response. The level crossing technique provides K samples $(H_k, \delta\omega_k)_{k=1,\dots,K}$ which only describe the transfer function for the positive frequencies (see Figure 1(b)). We can compute the frequencies of the samples from the interval lengths: $\omega_k = \sum_1^k \delta\omega_{k'}$. A symmetric filter has even amplitude and odd phase, which means that sampling the negative part of the filter transfer function would have led to a sample H_{-k} at frequency $-\omega_k$ with $H_{-k} = H_k^*$ (complex conjugate of H_k). We also need the value of the filter transfer function for the zero frequency ω_0 and denote it by $H_0 \in \mathbb{R}$.

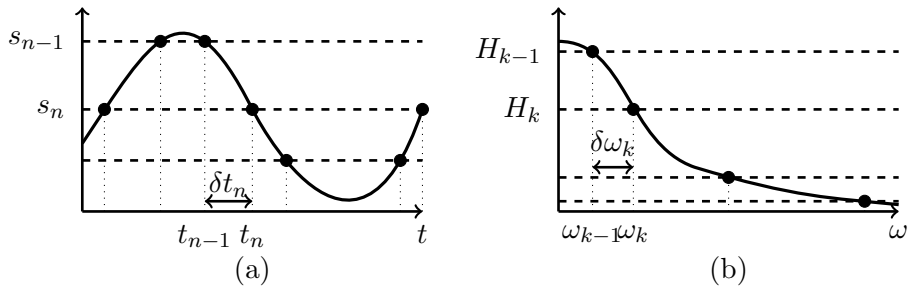

Figure 1: Non-uniform sampling of the input signal in the time domain (a) and the filter transfer function in the frequency domain (b).

Figure 1(b) suggests that we take samples of the filter transfer function when its amplitude crosses some predefined levels. We can combine this (or not) with taking samples when the phase also crosses levels. It is natural to take equi-spaced levels in phase, and non equi-spaced levels in amplitude. The choice of the levels (number, repartition) is not the goal of this paper, since we want to describe a generic algorithm, and strongly depends on the type of filter we want to sample.

2.2 Signal and filter interpolation

Thanks to interpolation we construct new analog functions from these samples, producing $\bar{s}$ for the signal and $\bar{H}$ for the filter transfer function. The numerical filter yields then for (possibly) all time t an approximate value for $x(t)$, namely

$$\bar{x}(t) = \int_{-\infty}^{T(t)} \bar{h}(t-\tau)\bar{s}(\tau)d\tau \text{ where } \bar{h}(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \bar{H}(\omega)e^{i\omega t}d\omega.$$

2.2.1 Signal interpolation

We consider here only piecewise constant and piecewise linear interpolation, which can both be cast as the piecewise linear case which reads

$$\bar{s}(t) = \sum_{n=1}^{N}(a_n + b_n t)\chi_{I_n}.$$

where χ_{I_n} is the characteristic function of the interval $I_n = [t_{n-1}, t_n]$. This formula covers in fact many situations.

Piecewise linear interpolation Piecewise linear interpolation yields

$$b_n = \frac{s_n - s_{n-1}}{\delta t_n} \text{ and } a_n = s_n - b_n t_n.$$

a_n can be seen as a weighted mean of s_{n-1} and s_n, namely

$$a_n = s_n - b_n t_n = s_n - \frac{s_n - s_{n-1}}{\delta t_n} t_n = \frac{t_n}{\delta t_n} s_{n-1} + \left(1 - \frac{t_n}{\delta t_n}\right) s_n = \frac{t_n}{\delta t_n} s_{n-1} - \frac{t_{n-1}}{\delta t_n} s_n.$$

This leads to define an initial sample at time t_0. There are basically three ways to perform this:

- consider that the signal starts at time t_0 and therefore $s_0 = 0$,
- consider that the signal is constant and therefore $s_0 = s_1$,
- consider that the signal slope is constant and therefore

$$s_0 = s_1 - (t_1 - t_0) b_2 = s_1 - (t_1 - t_0) \frac{s_2 - s_1}{t_2 - t_1} = \frac{t_2 - t_0}{t_2 - t_1} s_1 - \frac{t_1 - t_0}{t_2 - t_1} s_2.$$

The choice mainly affects the transient behavior, which is not an issue for filtering.

Piecewise constant interpolation. Piecewise constant interpolation means that $b_n = 0$ in the above formula. This case however splits in three sub-cases.

Right piecewise constant interpolation which consists in using the original samples which leads to $a_n = s_n$.

Left piecewise constant interpolation where we try to respect more or less the causality condition by taking $a_n = s_{n-1}$. However this is not completely achieved since we have to wait until time t_n to be able to process this sample. For the sample s_0, the choices $s_0 = 0$ and $s_0 = s_1$ are both possible.

Nearest neighbor piecewise constant interpolation. Piecewise constant approximations with nearest neighbor interpolation reads $a_n = s_n$ and $\tilde{t}_n = \frac{1}{2}(t_n + t_{n+1})$, which leads to redefine

$$\tilde{\delta t}_1 = \frac{1}{2}(2\delta t_1 + \delta t_2),\ \tilde{\delta t}_n = \frac{1}{2}(\delta t_n + \delta t_{n+1}) \text{ and } \tilde{\delta t}_N = \frac{1}{2}\delta t_N.$$

2.2.2 Causality

To take into account causality we have to be able to truncate the interpolated signal. Besides we have to interpolate the signal on the last interval, with no knowledge of the next sample, i.e. with left piecewise constant interpolation. If time t coincides with some sample, there is no specific problem, otherwise we have to define an extra sample at time t, which has the same amplitude as the previous sample. This leads to $N(t)$ samples for the truncated interpolated signal. We still call the amplitudes or coefficients s_n, a_n, b_n, but we keep in mind that the last sample $n = N(t)$ can be a duplicate of the previous one (with delay $t - t_{N(t)-1}$). If causality is not taken into account, $N(t) = N$ for all time.

2.2.3 Filter interpolation

For the filter transfer function, we have the same alternatives for piecewise constant approximations and we have many ways to write the linear interpolation. We denote by $J_k = [\omega_{k-1}, \omega_k]$ the frequency sampling intervals and $J_k^- = [-\omega_k, -\omega_{k-1}]$ the symmetric intervals. We investigate two methods. A graphical representation of both methods is given in Figures 2 and 3 in the discussion.

Real interpolation: Separate interpolation of the amplitude and the phase in the real domain. If we interpolate separately the amplitude ρ and the phase θ in the real domain, taking into account that amplitude is even and phase is odd, we obtain

$$\bar{H}(\omega) = \sum_{k=1}^{K} \left\{ (\rho_k^0 + \rho_k^1 \omega) e^{i(\theta_k^0 + \theta_k^1 \omega)} \chi_{J_k} + (\rho_k^0 - \rho_k^1 \omega) e^{-i(\theta_k^0 - \theta_k^1 \omega)} \chi_{J_k^-} \right\}.$$

where

$$\rho_k^1 = \frac{|H_k| - |H_{k-1}|}{\delta\omega_k},\ \rho_k^0 = |H_k| - \rho_k^1 \omega_k,$$
$$\theta_k^1 = \frac{\arg(H_k) - \arg(H_{k-1})}{\delta\omega_k} \text{ and } \theta_k^0 = \arg(H_k) - \theta_k^1 \omega_k.$$

Complex interpolation: Interpolation in the complex plane. Interpolation in the complex plane yields

$$\bar{H}(\omega) = \sum_{k=1}^{K} \left\{ (\zeta_k^0 + \zeta_k^1 \omega) \chi_{J_k} + (\zeta_k^{0*} - \zeta_k^{1*} \omega) \chi_{J_k^-} \right\}$$

where

$$\zeta_k^1 = \frac{H_k - H_{k-1}}{\delta\omega_k} \text{ and } \zeta_k^0 = H_k - \zeta_k^1 \omega_k.$$

For further computations we need to define the real and imaginary parts of the coefficients: $\zeta_k^0 = \xi_k^0 + i\eta_k^0$ and $\zeta_k^1 = \xi_k^1 + i\eta_k^1$.

3 Towards a summation formula

Now $\bar{h}(t)$ can be split as a sum of elementary impulse responses $\sum_{k=1}^{K} h_k(t)$ with a different expression according to the interpolation chosen for the filter transfer function. We compute explicitly the functions $h_k(t)$ in Section 3.2. Functions $h_k(t)$ have an infinite support. This is not a practical problem since we compute the convolution of these functions with the compact support signal $\bar{s}(t)$. This convolution equals

$$\begin{aligned}\bar{x}(t) &= \int_{-\infty}^{T(t)} \bar{h}(t-\tau)\bar{s}(\tau)d\tau = \sum_{n=1}^{N(t)}\sum_{k=1}^{K}\int_{t_{n-1}}^{t_n} h_k(t-\tau)(a_n+b_n\tau)d\tau \\ &= \sum_{n=1}^{N(t)}\left\{a_n\sum_{k=1}^{K}h_{nk}^0(t) + b_n\sum_{k=1}^{K}h_{nk}^1(t)\right\},\end{aligned}$$

where

$$h_{nk}^0(t) = \int_{t_{n-1}}^{t_n} h_k(t-\tau)d\tau \text{ and } h_{nk}^1(t) = \int_{t_{n-1}}^{t_n} h_k(t-\tau)\tau d\tau.$$

We have a summation formulation for which we have to compute the elementary contributions $h_{nk}^0(t)$ and $h_{nk}^1(t)$.

3.1 Elementary impulse responses

We first define the elementary impulse responses by integrating in frequency both on intervals H_k and H_{-k}, in the real case

$$\begin{aligned}h_k(t) &= \frac{1}{2\pi}\left\{\int_{\omega_{k-1}}^{\omega_k}(\rho_k^0+\rho_k^1\omega)e^{i(\theta_k^0+\theta_k^1\omega)}e^{i\omega t}d\omega + \int_{-\omega_k}^{-\omega_{k-1}}(\rho_k^0-\rho_k^1\omega)e^{-i(\theta_k^0-\theta_k^1\omega)}e^{i\omega t}d\omega\right\} \\ &= \frac{\rho_k^0\cos(\theta_k^0)}{\pi}\int_{\omega_{k-1}}^{\omega_k}\cos(\omega(t+\theta_k^1))d\omega - \frac{\rho_k^0\sin(\theta_k^0)}{\pi}\int_{\omega_{k-1}}^{\omega_k}\sin(\omega(t+\theta_k^1))d\omega \\ &+ \frac{\rho_k^1\cos(\theta_k^0)}{\pi}\int_{\omega_{k-1}}^{\omega_k}\cos(\omega(t+\theta_k^1))\omega d\omega - \frac{\rho_k^1\sin(\theta_k^0)}{\pi}\int_{\omega_{k-1}}^{\omega_k}\sin(\omega(t+\theta_k^1))\omega d\omega,\end{aligned}$$

and in the complex case

$$\begin{aligned} h_k(t) &= \frac{1}{2\pi}\left\{\int_{\omega_{k-1}}^{\omega_k}(\zeta_k^0+\zeta_k^1\omega)e^{i\omega t}d\omega+\int_{-\omega_k}^{-\omega_{k-1}}(\zeta_k^{0*}-\zeta_k^{1*}\omega)e^{i\omega t}d\omega\right\} \\ &= \frac{\xi_k^0}{\pi}\int_{\omega_{k-1}}^{\omega_k}\cos(\omega t)d\omega-\frac{\eta_k^0}{\pi}\int_{\omega_{k-1}}^{\omega_k}\sin(\omega t)d\omega \\ &+ \frac{\xi_k^1}{\pi}\int_{\omega_{k-1}}^{\omega_k}\cos(\omega t)\omega d\omega-\frac{\eta_k^1}{\pi}\int_{\omega_{k-1}}^{\omega_k}\sin(\omega t)\omega d\omega. \end{aligned}$$

3.2 Computation of the elementary impulse responses

In both the real and the complex cases, we can split each elementary impulse response into four contributions

$$h_k(t)=\alpha_k h_k^\alpha(t)+\beta_k h_k^\beta(t)+\gamma_k h_k^\gamma(t)+\delta_k h_k^\delta(t),$$

where

$$\alpha_k=\frac{\rho_k^0\cos(\theta_k^0)}{\pi},\ \beta_k=-\frac{\rho_k^0\sin(\theta_k^0)}{\pi},\ \gamma_k=\frac{\rho_k^1\cos(\theta_k^0)}{\pi},\ \delta_k=-\frac{\rho_k^1\sin(\theta_k^0)}{\pi},\ \text{in the real case,}$$
$$\alpha_k=\frac{\xi_k^0}{\pi},\ \beta_k=-\frac{\eta_k^0}{\pi},\ \gamma_k=\frac{\xi_k^1}{\pi},\ \delta_k=-\frac{\eta_k^1}{\pi},\ \theta_k^1=0,\ \text{in the complex case,}$$

and

$$\begin{aligned} h_k^\alpha(t) &= \int_{\omega_{k-1}}^{\omega_k}\cos(\omega(t+\theta_k^1))d\omega, \\ h_k^\beta(t) &= \int_{\omega_{k-1}}^{\omega_k}\sin(\omega(t+\theta_k^1))d\omega, \\ h_k^\gamma(t) &= \int_{\omega_{k-1}}^{\omega_k}\cos(\omega(t+\theta_k^1))\omega d\omega, \\ h_k^\delta(t) &= \int_{\omega_{k-1}}^{\omega_k}\sin(\omega(t+\theta_k^1))\omega d\omega. \end{aligned}$$

3.2.1 Regular formulae for elementary impulse responses

We first compute the $h_k^\varepsilon(t)$, for $\varepsilon=\alpha,\ \beta,\ \gamma,\ \delta$, assuming that $t\neq-\theta_k^1$,

$$\begin{aligned} h_k^\alpha(t) &= \left\{\frac{\sin(\omega_k(t+\theta_k^1))}{t+\theta_k^1}-\frac{\sin(\omega_{k-1}(t+\theta_k^1))}{t+\theta_k^1}\right\}, \\ h_k^\beta(t) &= -\left\{\frac{\cos(\omega_k(t+\theta_k^1))}{t+\theta_k^1}-\frac{\cos(\omega_{k-1}(t+\theta_k^1))}{t+\theta_k^1}\right\}, \end{aligned}$$

$$
\begin{aligned}
h_k^{\gamma}(t) &= \left\{ \frac{\cos(\omega_k(t+\theta_k^1))}{(t+\theta_k^1)^2} - \frac{\cos(\omega_{k-1}(t+\theta_k^1))}{(t+\theta_k^1)^2} \right\} \\
&\quad + \left\{ \omega_k \frac{\sin(\omega_k(t+\theta_k^1))}{t+\theta_k^1} - \omega_{k-1} \frac{\sin(\omega_{k-1}(t+\theta_k^1))}{t+\theta_k^1} \right\}, \\
h_k^{\delta}(t) &= \left\{ \frac{\sin(\omega_k(t+\theta_k^1))}{(t+\theta_k^1)^2} - \frac{\sin(\omega_{k-1}(t+\theta_k^1))}{(t+\theta_k^1)^2} \right\} \\
&\quad - \left\{ \omega_k \frac{\cos(\omega_k(t+\theta_k^1))}{t+\theta_k^1} - \omega_{k-1} \frac{\cos(\omega_{k-1}(t+\theta_k^1))}{t+\theta_k^1} \right\}.
\end{aligned}
$$

The integrals are not singular for $t = -\theta_k^1$ and therefore we want to find formulae that are valid for all t in order to avoid test cases in the filter implementation.

Definition 1 *We define the usual cardinal functions.*

$$\operatorname{sinc}(x) = \frac{\sin(x)}{x} \text{ and } \operatorname{cosc}(x) = \frac{\cos(x)-1}{x}.$$

sinc *is an even function with* $\operatorname{sinc}(0) = 1$ *and* cosc *is an odd function.*

With these special functions, we may rewrite

$$
\begin{aligned}
h_k^{\alpha}(t) &= \left\{ \omega_k \operatorname{sinc}(\omega_k(t+\theta_k^1)) - \omega_{k-1} \operatorname{sinc}(\omega_{k-1}(t+\theta_k^1)) \right\}, \\
h_k^{\beta}(t) &= -\left\{ \omega_k \operatorname{cosc}(\omega_k(t+\theta_k^1)) - \omega_{k-1} \operatorname{cosc}(\omega_{k-1}(t+\theta_k^1)) \right\}.
\end{aligned}
$$

Definition 2 *To rewrite* $h_k^{\gamma}(t)$ *and* $h_k^{\delta}(t)$ *we need specific "double" cardinal functions which are both continuous on* $\mathbb{R}$.

$$\operatorname{sincc}(x) = \frac{\operatorname{sinc}(x)-1}{x} \text{ and } \operatorname{coscc}(x) = \frac{\operatorname{cosc}(x)}{x}.$$

sincc *is an odd function and* coscc *is an even function with* $\operatorname{coscc}(0) = -\frac{1}{2}$.

We then have

$$
\begin{aligned}
h_k^{\gamma}(t) &= \left\{ \omega_k^2 \operatorname{coscc}(\omega_k(t+\theta_k^1)) - \omega_{k-1}^2 \operatorname{coscc}(\omega_{k-1}(t+\theta_k^1)) \right\} \\
&\quad + \left\{ \omega_k^2 \operatorname{sinc}(\omega_k(t+\theta_k^1)) - \omega_{k-1}^2 \operatorname{sinc}(\omega_{k-1}(t+\theta_k^1)) \right\}, \\
h_k^{\delta}(t) &= \left\{ \omega_k^2 \operatorname{sincc}(\omega_k(t+\theta_k^1)) - \omega_{k-1}^2 \operatorname{sincc}(\omega_{k-1}(t+\theta_k^1)) \right\} \\
&\quad - \left\{ \omega_k^2 \operatorname{cosc}(\omega_k(t+\theta_k^1)) - \omega_{k-1}^2 \operatorname{cosc}(\omega_{k-1}(t+\theta_k^1)) \right\}.
\end{aligned}
$$

The cardinal and double cardinal functions are regular and these new formulations of the $h_k^{\varepsilon}(t)$ are also valid for $t = -\theta_k^1$.

3.2.2 Centered elementary impulse responses

Now clearly $h_k^\alpha(t)$, $h_k^\beta(t)$, $h_k^\gamma(t)$ and $h_k^\delta(t)$ are functions of $t+\theta_k^1$ rather than t, therefore for $\varepsilon = \alpha,\ \beta,\ \gamma,\ \delta$ we define $\tilde{h}_k^\varepsilon(t+\theta_k^1) = h_k^\varepsilon(t)$, where

$$\begin{aligned}
\tilde{h}_k^\alpha(t) &= \left\{\omega_k \operatorname{sinc}(\omega_k t) - \omega_{k-1}\operatorname{sinc}(\omega_{k-1}t)\right\},\\
\tilde{h}_k^\beta(t) &= -\left\{\omega_k \operatorname{cosc}(\omega_k t) - \omega_{k-1}\operatorname{cosc}(\omega_{k-1}t)\right\},\\
\tilde{h}_k^\gamma(t) &= \left\{\omega_k^2 \operatorname{coscc}(\omega_k t) - \omega_{k-1}^2\operatorname{coscc}(\omega_{k-1}t)\right\} + \left\{\omega_k^2 \operatorname{sinc}(\omega_k t) - \omega_{k-1}^2\operatorname{sinc}(\omega_{k-1}t)\right\},\\
\tilde{h}_k^\gamma(t) &= \left\{\omega_k^2 \operatorname{sincc}(\omega_k t) - \omega_{k-1}^2\operatorname{sincc}(\omega_{k-1}t)\right\} - \left\{\omega_k^2 \operatorname{cosc}(\omega_k t) - \omega_{k-1}^2\operatorname{cosc}(\omega_{k-1}t)\right\}.
\end{aligned}$$

3.3 Computation of the elementary contributions

The elementary contributions are computed using the values of the elementary impulse responses. We therefore should compute

$$h_{nk}^0(t) = \alpha_k h_{nk}^{0\alpha}(t) + \beta_k h_{nk}^{0\beta}(t) + \gamma_k h_{nk}^{0\gamma}(t) + \delta_k h_{nk}^{0\delta}(t)$$

and

$$h_{nk}^1(t) = \alpha_k h_{nk}^{1\alpha}(t) + \beta_k h_{nk}^{1\beta}(t) + \gamma_k h_{nk}^{1\gamma}(t) + \delta_k h_{nk}^{1\delta}(t),$$

for $\varepsilon = \alpha,\ \beta,\ \gamma,\ \delta$, where

$$\begin{aligned}
h_{nk}^{0\varepsilon}(t) &= \int_{t_{n-1}}^{t_n} h_k^\varepsilon(t-\tau)d\tau = \int_{t_{n-1}}^{t_n} \tilde{h}_k^\varepsilon(t-\tau+\theta_k^1)d\tau,\\
h_{nk}^{1\varepsilon}(t) &= \int_{t_{n-1}}^{t_n} h_k^\varepsilon(t-\tau)\tau d\tau = \int_{t_{n-1}}^{t_n} \tilde{h}_k^\varepsilon(t-\tau+\theta_k^1)\tau d\tau.
\end{aligned}$$

The final results will be easily written in terms of the shifted times $t_{n,k}(t) = t - t_n + \theta_k^1$ and we can cast the elementary contributions as

$$\begin{aligned}
h_{nk}^{0\varepsilon}(t) &= \int_{t_{n,k}(t)}^{t_{n-1,k}(t)} \tilde{h}_k^\varepsilon(\tau)d\tau,\\
h_{nk}^{1\varepsilon}(t) &= \int_{t_{n,k}(t)}^{t_{n-1,k}(t)} \tilde{h}_k^\varepsilon(\tau)(t-\tau+\theta_k^1)d\tau = (t+\theta_k^1)h_{nk}^{0\varepsilon}(t) - \hat{h}_{nk}^\varepsilon(t)
\end{aligned}$$

where

$$\hat{h}_{nk}^\varepsilon(t) = \int_{t_{n,k}(t)}^{t_{n-1,k}(t)} \tilde{h}_k^\varepsilon(\tau)\tau d\tau.$$

Using the special functions and primitives rescaled in Appendix A:, we compute

$$h_{nk}^{0\alpha}(t) = \{\mathrm{Si}(\omega_k t_{n-1,k}(t)) - \mathrm{Si}(\omega_k t_{n,k}(t))\} - \{\mathrm{Si}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{Si}(\omega_{k-1} t_{n,k}(t))\},$$

$$
\begin{aligned}
h_{nk}^{0\beta}(t) &= \{\mathrm{Cin}(\omega_k t_{n-1,k}(t)) - \mathrm{Cin}(\omega_k t_{n,k}(t))\} \\
&\quad -\{\mathrm{Cin}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{Cin}(\omega_{k-1} t_{n,k}(t))\}, \\
h_{nk}^{0\gamma}(t) &= -\omega_k\{\mathrm{cosc}(\omega_k t_{n-1,k}(t)) - \mathrm{cosc}(\omega_k t_{n,k}(t))\} \\
&\quad +\omega_{k-1}\{\mathrm{cosc}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{cosc}(\omega_{k-1} t_{n,k}(t))\}, \\
h_{nk}^{0\delta}(t) &= -\omega_k\{\mathrm{sinc}(\omega_k t_{n-1,k}(t)) - \mathrm{sinc}(\omega_k t_{n,k}(t))\} \\
&\quad +\omega_{k-1}\{\mathrm{sinc}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{sinc}(\omega_{k-1} t_{n,k}(t))\}.
\end{aligned}
$$

and

$$
\begin{aligned}
\hat{h}_{nk}^{\alpha}(t) &= -\{t_{n-1,k}(t)\,\mathrm{cosc}(\omega_k t_{n-1,k}(t)) - t_{n,k}(t)\,\mathrm{cosc}(\omega_k t_{n,k}(t))\} \\
&\quad +\{t_{n-1,k}(t)\,\mathrm{cosc}(\omega_{k-1} t_{n-1,k}(t)) - t_{n,k}(t)\,\mathrm{cosc}(\omega_{k-1} t_{n,k}(t))\}, \\
\hat{h}_{nk}^{\beta}(t) &= -\{t_{n-1,k}(t)\,\mathrm{sinc}(\omega_k t_{n-1,k}(t)) - t_{n,k}(t))\,\mathrm{sinc}(\omega_k t_{n,k}(t))\} \\
&\quad +\{t_{n-1,k}(t)\,\mathrm{sinc}(\omega_{k-1} t_{n-1,k}(t)) - t_{n,k}(t)\,\mathrm{sinc}(\omega_{k-1} t_{n,k}(t))\}, \\
\hat{h}_{nk}^{\gamma}(t) &= -\{\mathrm{Cin}(\omega_k t_{n-1,k}(t)) - \mathrm{Cin}(\omega_k t_{n,k}(t))\} \\
&\quad +\{\mathrm{Cin}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{Cin}(\omega_{k-1} t_{n,k}(t))\} \\
&\quad -\{\cos(\omega_k t_{n-1,k}(t)) - \cos(\omega_k t_{n,k}(t))\} \\
&\quad +\{\cos(\omega_{k-1} t_{n-1,k}(t)) - \cos(\omega_{k-1} t_{n,k}(t))\}, \\
\hat{h}_{nk}^{\delta}(t) &= \{\mathrm{Si}(\omega_k t_{n-1,k}(t)) - \mathrm{Si}(\omega_k t_{n,k}(t))\} \\
&\quad -\{\mathrm{Si}(\omega_{k-1} t_{n-1,k}(t)) - \mathrm{Si}(\omega_{k-1} t_{n,k}(t))\} \\
&\quad -\{\sin(\omega_k t_{n-1,k}(t)) - \sin(\omega_k t_{n,k}(t))\} \\
&\quad +\{\sin(\omega_{k-1} t_{n-1,k}(t)) - \sin(\omega_{k-1} t_{n,k}(t))\}.
\end{aligned}
$$

3.4 Summation formulae according to the signal interpolation type

3.4.1 Piecewise constant approximation

In the case of piecewise constant approximations the summation formula restricts to

$$
\bar{x}(t) = \sum_{n=1}^{N(t)} s_n \sum_{k=1}^{K} h_{nk}^{0}(t).
$$

3.4.2 Linear interpolation

In the case of linear interpolation, for fixed n and k, we have

$$
\begin{aligned}
& a_n h_{nk}^{0}(t) + b_n h_{nk}^{1}(t) \\
= \;& (a_n + b_n(t + \theta_k^1)) h_{nk}^{0}(t) - b_n \hat{h}_{nk}(t) \\
= \;& -\frac{1}{\delta t_n}(t_{n,k}(t) h_{nk}^{0}(t) - \hat{h}_{nk}(t)) s_{n-1} + \frac{1}{\delta t_n}(t_{n-1,k}(t) h_{nk}^{0}(t) - \hat{h}_{nk}(t)) s_n.
\end{aligned}
$$

Therefore the summation formula reads in the linear case

$$\begin{aligned}\bar{x}(t) &= \sum_{n=1}^{N(t)-1}\sum_{k=1}^{K} s_n \Big(\frac{1}{\delta t_n}(t_{n-1,k}(t)h^0_{nk}(t) - \hat{h}_{nk}(t)) \\ &\qquad - \frac{1}{\delta t_{n+1}}(t_{n+1,k}(t)h^0_{n+1,k}(t) - \hat{h}_{n+1,k}(t))\Big) \\ &\quad -s_0\frac{1}{\delta t_1}\sum_{k=1}^{K}(t_{1,k}(t)h^0_{1,k}(t) - \hat{h}_{1,k}(t)) \\ &\quad +s_{N(t)}\frac{1}{\delta t_{N(t)}}\sum_{k=1}^{K}(t_{N(t)-1,k}(t)h^0_{N(t),k}(t) - \hat{h}_{N(t),k}(t)).\end{aligned}$$

4 Discussion

4.1 Choice of the transfer function linear interpolation

We have given two means to perform the linear interpolation of the transfer function, above-mentioned as the real and complex interpolations. They lead to very similar algorithms since only the coefficients α_k, β_k, γ_k and δ_k depend on the choice of the linear interpolation type. However they lead to very different result, and the reasons can be deduced from Figure 2.

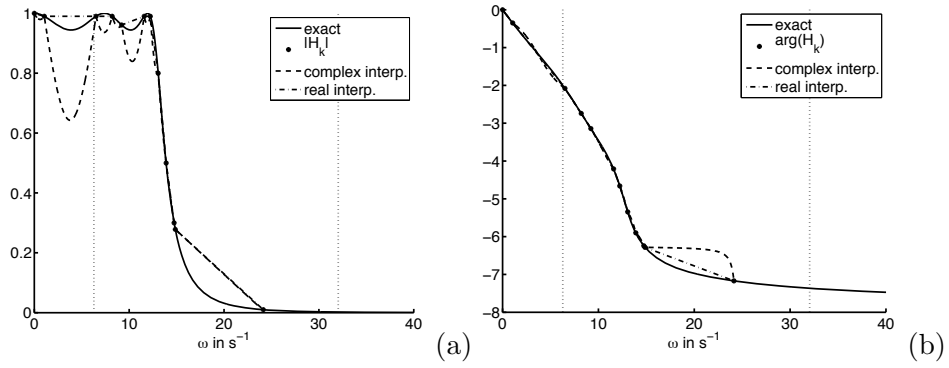

Figure 2: Amplitude response (a) and phase response (b) of the linear interpolations (real (dashed-dotted plot) and complex (dashed plot) interpolations) of a Chebychev filter compared to the exact values (solid line).

This figure represents the amplitude response (a) and the phase response (b) of a Chebyshev filter and its linear interpolations *via* the real (dashed-dotted plot) and complex (dashed plot) interpolations. Real interpolation consists in linearly interpolating the amplitude and the phase and therefore the dashed-dotted plot consists of piecewise affine functions. Complex interpolation is a

linear interpolation in the complex plane and as can be seen in Figure 3, interpolating in the complex plane, lead to smaller amplitudes.

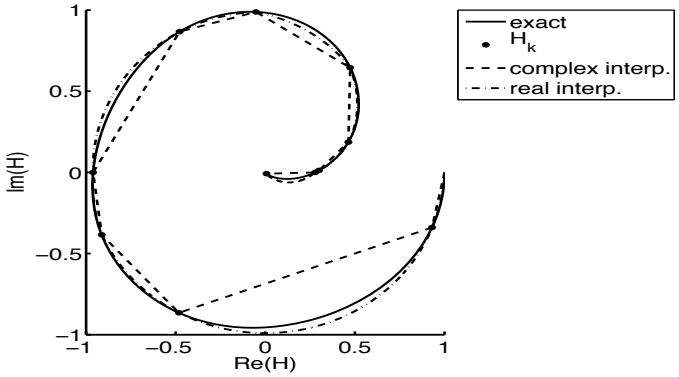

Figure 3: Complex plane representation of linear interpolations (real (dashed-dotted plot) and complex (dashed plot) interpolations) of a Chebychev filter compared to the exact values (solid line).

This lead to the dip in Figure 2(a) and really bad filtering of the low frequencies. In the case of the Chebyshev filter, real interpolation has another advantage to cut the oscillations in the low frequencies, leading to a better filtering of the low frequencies than usual filtering methods.

We also notice a bump between the last two samples on the phase interpolation (Figure 2(b)) which is more developed for complex interpolation. If one of the frequencies we want to filter out lies in this range, the filtering result will be altered.

In the examples below the linear interpolations of the transfer function will therefore be presented with real interpolation.

4.2 Numerical results

To illustrate this filtering algorithm we filter the signal

$$s(t) = 0.45\sin(2\pi t) + 0.45\sin(10.2\pi t) + 0.9$$

with various low pass filters with the cutoff frequency $\omega_c = 4\pi$. This is not the typical sort of signal which is supposed to be addressed by our technique since it is not a sporadic one and a relatively large number of samples are taken. We perform the computations within the MATLAB SPASS (Signal Processing for ASynchronous Systems) framework [6]).

This signal is sampled with an M-bit Asynchronous A/D Converter (AADC) which leads to a level crossing sampling over the amplitude range $[0, 1.8]$. 1.8 V

is a classical nominal voltage for a 180nm CMOS technology used for fabricating the AADC [3].

We can choose as we want the times at which the filtered signal is computed. To display the results we choose a sequence of times t_m such that the sampling points are dispatched irregularly over the obtained solution.

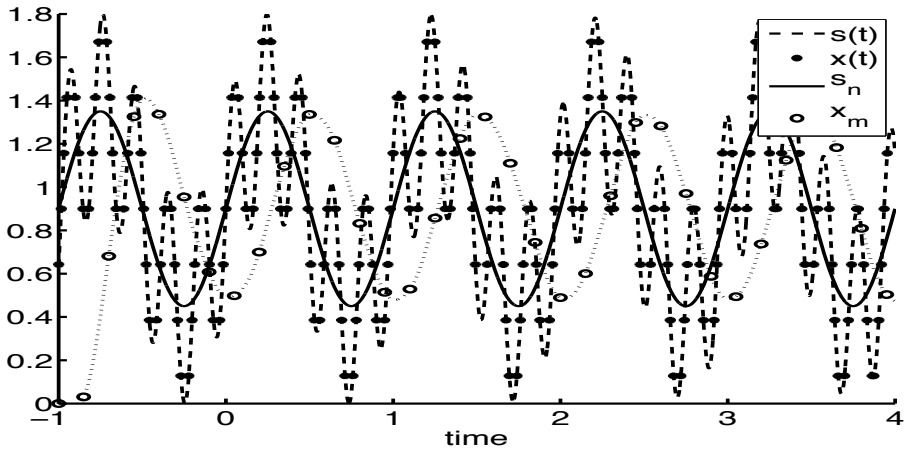

Figure 4: Example of filtering result. Initial signal (dashed line), theoretical filtered signal (solid line), non-uniformly sampled initial signal (asterisk markers) and computed filtered samples (circle markers connected with a dotted line)

In Figure 4, you can see an example of the result with a 5th order Butterworth filter and a 3-bit AADC. We plot continuous functions with lines: the initial signal $s(t)$ (dashed line) and the theoretical filtered signal $x(t)$ (solid line). We plot the sampled results with markers: the non-uniformly sampled initial signal s_n (asterisk markers) and the computed filtered samples x_m (circle markers) at times t_m.

Figure 4 is only an example of a possible result since there are very numerous parameters to tune the method:

- number of bits of the AADC,
- continuous filter chosen (Butterworth, Chebyshev, elliptic, Cauer, and many others),
- choice of the levels to quantize the transfer function, number of these levels, equi-spaced or not in amplitude, levels in amplitude and in phase,
- choice of interpolation type for both the signal and the filter.

The theoretical filtered signal displayed in Figure 4 is given by

$$x(t) = 0.45 \sin(2\pi t) + 0.9,$$

but this is not a fair function to compare with since there is always a phase shift due to the value of the filter phase at the 2π-frequency. Therefore we will compare our results to $x(t) = 0.45\sin(2\pi(t+\phi)) + 0.9$, where ϕ depends on the theoretical filter used. We also consider from Figure 4 that the transient evolution takes place in the $[-1, 0]$ time interval and therefore only compare the results for positive times.

Since we cannot explore here the whole range of possible parameters, we chose to present only a 5th order Butterworth filter. We sample it both in amplitude (0.99, 0.8, 0.5, 0.3, and 0.01) and phase (multiples of π) leading to 11 frequency samples. We compare here the results obtained for left piecewise constant and linear interpolation for both the signal and the transfer function and for different values (2, 3, 4 and 5) of the AADC resolution. Causality is taken into account.

On Tables 1 and 2 we give the relative l^2 and l^∞ errors between computed filtered samples x_m at times $t_m = .01m$ (m integer) and the theoretical values $x(t_m)$.

signal	filter	$M=2$	$M=3$	$M=4$	$M=5$
left	left	0.0569	0.0597	0.0597	0.0597
	linear	0.0120	0.0025	0.0017	0.0018
linear	left	0.0577	0.0601	0.0599	0.0598
	linear	0.0096	0.0033	0.0021	0.0020

Table 1: l^2 error with an interpolated Butterworth filter. Comparison of various interpolation types and AADC resolutions

signal	filter	$M=2$	$M=3$	$M=4$	$M=5$
left	left	0.4549	0.4950	0.4854	0.4846
	linear	0.1154	0.0264	0.0170	0.0210
linear	left	0.4636	0.5035	0.4904	0.4870
	linear	0.0963	0.0347	0.0187	0.0214

Table 2: l^∞ error with an interpolated Butterworth filter. Comparison of various interpolation types and AADC resolutions

We first see that left interpolation is a very bad choice for interpolating the transfer function and we investigate this point further in Table 3 and 4.

In the case of the 2-bit AADC, there are 2.8 points per period for the highest frequency part of the signal. This is a very low rate, and we are however able to have only 1% l^2 error on the filtered result which is quite sufficient for a large range of applications. The other results all show less than 1% error. However we have to be very careful about the values displayed on Tables 1 and 2 which

are very dependent on the choice of the function to filter.

We now fix the AADC resolution to 3 bits and choose a linear interpolation for the signal to only compare the possible interpolation of the transfer function. This time, we choose a Chebyshev filter, with exactly the same sampling choice as for the above Butterworth filter.

filter interpolation	left	right	complex linear	real linear
exact	0.2058	0.0682	0.0626	0.0427
approximate	0.1756	0.0871	0.0550	0.0122

Table 3: l^2 error with an interpolated Chebyshev filter. Comparison of various interpolation types for the transfer function

filter interpolation	left	right	complex linear	real linear
exact	0.0480	0.0170	0.0159	0.0101
approximate	0.0427	0.0201	0.0126	0.0038

Table 4: l^∞ error with an interpolated Chebyshev filter. Comparison of various interpolation types for the transfer function

The first line on Tables 3 and 4 is the error with respect to the shifted exact filtered signal for the theoretical shift at 2π (-0.3651 rad), whereas the second line, which shows smaller errors, is the error for the approximated shift, i.e. the linearly interpolated phase, 0.3321 rad. Only the real linear interpolation yields a good approximation. The other methods give incorrect amplitudes either for the constant or the 2π frequencies, or add a low-frequency component.

5 Conclusions

We have presented a novel approach to filtering based on the non-uniform sampling of the signal but also the non-uniform sampling in frequency of the filter transfer function. The final result is complex but nonetheless possible to implement in hardware devices and of course in numerical codes. This complexity is balanced by the very low number of samples and the relatively low number of operations needed for each evaluation. This approach is very promising to achieve a lower power consumption in the upcoming mobile systems.

Indeed, the power consumption of electronic devices is correlated to the activity of the system, which is in our case drastically reduced, thanks to our specific samplings applied to the signal and to the filter. A preliminary study has already been done to estimate the overall consumption of such a system and the first implementation results show interesting power saving with the use of our A-ADC [4]. Moreover, the activity is directly connected to the signal and

is able to produce a non-uniform sampling with a reduced number of samples. We can notice that this approach is particularly interesting with some sporadic signals. Moreover, coupling this technique with asynchronous (clockless) digital is really appropriate since asynchronous logic only computes when samples are available (event-driven logic). This will lead to drastic power savings compared to the classical approach, i.e. uniform sampling and clocked circuits.

ACKNOWLEDGEMENT

This work has been supported by a funding from the Joseph Fourier-Grenoble 1 University: MSTIC project TATIE.

References

[1] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, Asynchronus fir filters, towards a new digital processing chain, in *10th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'04)*, 198-206, Crete, Greece, April 2004.

[2] F. Akopyan, R. Manohar, and A.B. Apsel, A level-crossing flash asynchronous analog-to-digital converter, in *12th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'06)*, 11-22, Grenoble, France, March 2006.

[3] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, A new class of asynchronous A/D converters based on time quantization, in *9th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'03)*, 197-205, Vancouver, Canada, May 2003.

[4] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, Asynchronous Level Crossing Analog to Digital Converters, Special Issue on ADC Modelling and Testing of *Measurement*, **37**, 296-309, 2005.

[5] B. Bidégaray-Fesquet and L. Fesquet, A fully non-uniform approach to FIR filtering, in *8th International Conference on Sampling Theory and Applications (SampTa'09)*, Marseille, France, May 2009.

[6] B. Bidégaray-Fesquet and L. Fesquet, SPASS 2.0: Signal Processing for ASynchronous Systems, May 2010, `http://ljk.imag.fr/membres/Brigitte.Bidegaray/SPASS/`.

[7] L. Fesquet and B. Bidégaray-Fesquet, IIR digital filtering of non-uniformly sampled signals via state representation, *Signal Processing*, **90**, 2811-2821, 2010.

[8] L. Fesquet, G. Sicard, and B. Bidégaray-Fesquet, Targeting ultra-low power consumption with non-uniform sampling and filtering, in *IEEE International Symposium on Circuits and Systems (ISCAS2010)*, Paris, France, June 2010.

[9] J.W. Mark and T.D. Todd, A nonuniform sampling approach to data compression, *IEEE Trans. on Communications*, **29**, 24-32, 1981.

[10] F. A. Marvasti, *Nonuniform Sampling. Theory and Practice*. Information Technology: Transmission, Processing and Storage, Springer, 2001.

[11] S. Mian Qaisar, L. Fesquet, and M. Renaudin, Adaptive rate filtering for a signal driven sampling scheme, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*, vol. III, 1465-1468, Honolulu, Hawaii, USA, April 2007.

[12] N. Sayiner, H.V. Sorensen, and T.R. Viswanathan, A level-crossing sampling scheme for A/D conversion, *IEEE Trans. on Circuits and Systems II*, **43**, 335-339, April 1996.

Appendix A: Special functions and primitives

A.1 Mathematical tools: Exponential integral, sine integral and cosine integral

The exponential integral is defined for positive x by

$$\mathrm{Ei}(ix) = -\int_x^{+\infty} e^{iy}\frac{dy}{y} + i\frac{\pi}{2}.$$

The sine integral is defined by

$$\mathrm{Si}(x) = \int_0^x \sin(y)\frac{dy}{y}.$$

For a positive x is equal to

$$\mathrm{Si}(x) = \frac{1}{2i}(\mathrm{Ei}(ix) - \mathrm{Ei}(-ix)) + \frac{\pi}{2}.$$

The cosine integral is defined by

$$\mathrm{Ci}(x) = -\int_x^{\infty} \cos(y)\frac{dy}{y}.$$

For a positive x is equal to

$$\mathrm{Ci}(x) = \frac{1}{2}(\mathrm{Ei}(ix) + \mathrm{Ei}(-ix)).$$

We use in fact the integral of the cosine cardinal. We classically denote for x positive

$$\mathrm{Cin}(x) = \int_0^x (1-\cos(y))\frac{dy}{y} = -\,\mathrm{Ci}(x) + \gamma + \ln x,$$

where γ is the Euler constant:

$$\gamma = \lim_{n\to\infty}\left(\sum_{k=1}^{n}\frac{1}{k} - \ln n\right).$$

Clearly, Si is an odd function and Cin is an even function with $\mathrm{Cin}(0) = 0$.

A.2 A row of primitives

$$\begin{aligned}
\int^x \mathrm{sinc}(cy)dy &= \frac{1}{c}\int^{cx}\mathrm{sinc}(y)dy = \frac{1}{c}\,\mathrm{Si}(cx),\\
\int^x \mathrm{cosc}(cy)dy &= \frac{1}{c}\int^{cx}\mathrm{cosc}(y)dy = -\frac{1}{c}\,\mathrm{Cin}(cx),\\
\int^x \mathrm{sincc}(cy)dy &= \frac{1}{c}\int^{cx}\mathrm{sincc}(y)dy = -\frac{1}{c}\{\mathrm{Cin}(cx)+\mathrm{sinc}(cx)\},\\
\int^x \mathrm{coscc}(cy)dy &= \frac{1}{c}\int^{cx}\mathrm{coscc}(y)dy = -\frac{1}{c}\{\mathrm{Si}(cx)+\mathrm{cosc}(cx)\}.
\end{aligned}$$

We specifically use the combinations

$$\begin{aligned}
\int^x (\mathrm{sinc}(cy)+\mathrm{coscc}(cy))dy &= -\frac{1}{c}\,\mathrm{cosc}(cx),\\
\int^x (\mathrm{sincc}(cy)-\mathrm{cosc}(cy))dy &= -\frac{1}{c}\,\mathrm{sinc}(cx).
\end{aligned}$$