Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 147407, 10 pages
doi:10.1155/2008/147407

Research Article

Downsampling Non-Uniformly Sampled Data

Frida Eng and Fredrik Gustafsson

Department of Electrical Engineering, Linköpings Universitet, 58183 Linköping, Sweden

Correspondence should be addressed to Fredrik Gustafsson, fredrik@isy.liu.se

Received 14 February 2007; Accepted 17 July 2007

Recommended by T.-H. Li

Decimating a uniformly sampled signal a factor D involves low-pass antialias filtering with normalized cutoff frequency $1/D$ followed by picking out every Dth sample. Alternatively, decimation can be done in the frequency domain using the fast Fourier transform (FFT) algorithm, after zero-padding the signal and truncating the FFT. We outline three approaches to decimate non-uniformly sampled signals, which are all based on interpolation. The interpolation is done in different domains, and the inter-sample behavior does not need to be known. The first one interpolates the signal to a uniformly sampling, after which standard decimation can be applied. The second one interpolates a continuous-time convolution integral, that implements the antialias filter, after which every Dth sample can be picked out. The third frequency domain approach computes an approximate Fourier transform, after which truncation and IFFT give the desired result. Simulations indicate that the second approach is particularly useful. A thorough analysis is therefore performed for this case, using the assumption that the non-uniformly distributed sampling instants are generated by a stochastic process.

1. INTRODUCTION

Downsampling is here considered for a non-uniformly sampled signal. Non-uniform sampling appears in many applications, while the cause for nonlinear sampling can be classified into one of the following two categories.

Event-based sampling

The sampling is determined by a nuisance event process. One typical example is data traffic in the Internet, where packet arrivals determine the sampling times and the queue length is the signal to be analyzed. Financial data, where the stock market valuations are determined by each transaction, is another example.

Uniform sampling in secondary domain

Some angular speed sensors give a pulse each time the shaft has passed a certain angle, so the sampling times depend on angular speed. Also biological signals such as ECGs are naturally sampled in the time domain, but preferably analyzed in another domain (heart rate domain).

A number of other applications and relevant references can be found in, for example, [1].

It should be obvious from the examples above that for most applications, the original non-uniformly sampled signal is sampled much too fast, and that oscillation modes and interesting frequency modes are found at quite low frequencies compared to the inverse mean sampling interval.

The problem at hand is stated as follows.

Problem 1. The following is given:

(a) a sequence of non-uniform sampling times, t_m, $m = 1, \ldots, M$;

(b) corresponding signal samples, $u(t_m)$;

(c) a filter impulse response, $h(t)$; and

(d) a resampling frequency, $1/T$.

Also, the desired intersampling time, T, is much larger than the original mean intersampling time,

$$\mu_T \triangleq \mathrm{E}[t_m - t_{m-1}] \approx \frac{t_M}{M} = T_u. \tag{1}$$

Let $\lfloor x \rfloor$ denote the largest integer smaller than or equal to x. Find

$$\hat{z}(nT), \quad n = 1, \ldots, N, \qquad N = \left\lfloor \frac{t_M}{T} \right\rfloor \triangleq \frac{M}{D}, \tag{2}$$

such that $\hat{z}(nT)$ approximates $z(nT)$, where

$$z(t) = h \star u(t) = \int h(t-\tau)u(\tau)d\tau \tag{3}$$

is given by convolution of the continuous-time filter $h(t)$ and signal $u(t)$.

For the case of uniform sampling, $t_m = mT_u$, two well-known solutions exist; see, for example, [2].

(a) First, if $T/T_u = D$ is an integer, then (i) $u(mT_u)$ is filtered giving $z(mT_u)$, and (ii) $z(nT) = z(nDT_u)$ gives the decimated signal.

(b) Further, if $T/T_u = R/S$ is a rational number, then a frequency domain method is known. It is based on (i) zero padding $u(mT_u)$ to length RM, (ii) computing the discrete Fourier transform (DFT), (iii) truncating the DFT a factor S, and finally computing the inverse DFT (IDFT), where the (I)FFT algorithm is used for the (I)DFT calculation.

Conversion between arbitrary sampling rates has also been discussed in many contexts. The issues with efficient implementation of the algorithms are investigated in [3–6], and some of the results are beneficial also for the non-uniform case.

Resampling and reconstruction are closely connected, since a reconstructed signal can be used to sample at desired time points. The task of reconstruction is well investigated for different setups of non-uniform sampling. A number of iterative solutions have been proposed, for example, [1, 7, 8], several more are also discussed in [9]. The algorithms are not well-suited for real-time implementations and are based on different assumptions on the sampling times, t_m, such as bounds on the maximum separation or deviation from the nominal value mT_u.

Russel [9] also investigates both uniform and non-uniform resampling thoroughly. Russell argues against the iterative solutions, since they are based on analysis with ideal filters, and no guarantees can be given for approximate solutions. A noniternative approach is given, which assumes periodic time grids, that is, the non-uniformity is repeated. Another overview of techniques for non-uniform sampling is given in [10], where, for example, Ferreira [11] studies the special case of recovery of missing data and Lacaze [12] reconstructs stationary processes.

Reconstruction of functions with a convolutional approach was done by [13], and later also by [14]. The sampling is done via basis functions, and reduces to the regular case if delta functions are used. These works are based on sampling sets that fulfill the non-uniform sampling theorem given in [15].

Reconstruction has long been an interesting topic in image processing, especially in medical imaging, see, for example, [16], where, in particular, problems with motion artifacts are addressed. Arbitrary sampling distributions are allowed, and the reconstruction is done through resampling to a uniform grid. The missing pixel problem is given attention in [9, 17]. In [18], approximation of a function with bounded variation, with a band-limited function, is considered and the approximation error is derived. Pointwise reconstruction is investigated in [19], and these results will be used in Section 5.

Here, we neither put any constraints on the non-uniform sampling times, nor assumptions on the signal's function class. Instead, we take a more application-oriented approach, and aim at good, implementable, resampling procedures. We will consider three different methods for converting from non-uniform to uniform sampling. The first and third algorithm are rather trivial modifications of the time and frequency-domain methods for uniformly sampled data, respectively, while the second one is a new truly non-uniform algorithm. We will compare performance of these three. In all three cases, different kinds of interpolation are possible, but we will focus on zero-order hold (nearest neighbor) and first order hold (linear interpolation). Of course, which interpolation is best depends on the signal and in particular on its inter-sample behavior. Though we prefer to talk about decimation, we want to point out that the theories hold for any type of filter $h(t)$.

A major contribution in this work is a detailed analysis of the algorithms, where we assume additive random sampling, (ARS),

$$t_m = t_{m-1} + \tau_m, \tag{4}$$

where τ_m is stochastic additive sampling noise given by the known probability density function $p_\tau(t)$. The theoretical results show that the downsampled signal is unbiased under fairly general conditions and present an equivalent filter that generates $z(t) = \widetilde{h} \star u(t)$, where $\widetilde{h}$ depends on the designed filter h and the characteristic function of the stochastic distribution.

The paper is organized as follows. The algorithms are described in Section 2. The convolutional interpolation gives promising results in the simulations in Section 3, and the last sections are dedicated to this algorithm. In Section 4, theoretic analysis of both finite time and asymptotic performance is done. The section also includes illustrative examples of the theory. Section 5 investigates an application example and issues with choosing the filter $h(t)$, while Section 6 concludes the paper.

2. INTERPOLATION ALGORITHMS

Time-domain interpolation can be used with subsequent filtering. Since LP-filtering is desired, we also propose two other methods that include the filter action directly. The main idea is to perform the interpolation at different levels and the problem was stated in Problem 1.

For Problem 1, with $T_u = t_M/M$, compute
(1) $t_m^j = \arg\min_{t_m < jT_u} |jT_u - t_m|$,
(2) $\hat{u}(jT_u) = u(t_m^j)$,
(3) $\hat{z}(kT) = \sum_{j=1}^{M} h_d(kT - jT_u)\hat{u}(jT_u)$,
where $h_d(t)$ is a discrete time realization of the impulse response $h(t)$.

Algorithm 1: Time-domain interpolation.

2.1. Interpolation in time domain

It is well described in literature how to interpolate a signal or function in, for instance, the following cases.

(i) The signal is band-limited, in which case the sinc interpolation kernel gives a reconstruction with no error [20].
(ii) The signal has vanishing derivatives of order $n+1$ and higher, in which case spline interpolation of order n is optimal [21].
(iii) The signal has a bounded second-order derivative, in which case the Epanechnikov kernel is the optimal interpolation kernel [19].

The computation burden in the first case is a limiting factor in applications, and for the other two examples, the interpolation is not exact. We consider a simple spline interpolation, followed by filtering and decimation as in Algorithm 1. This is a slight modification of the known solution in the uniform case as was mentioned in Section 1.

Algorithm 1 is optimal only in the unrealistic case where the underlying signal $u(t)$ is piecewise constant between the samples. The error will depend on the relation between the original and the wanted sampling; the larger the ratio M/N, the smaller the error. If one assumes a band-limited signal, where all energy of the Fourier transform $U(f)$ is restricted to $f < 0.5N/t_M$, then a perfect reconstruction would be possible, after which any type of filtering and sampling can be performed without error. However, this is not a feasible solution in practice, and the band-limited assumption is not satisfied for real signals when the sensor is affected by additive noise.

Remark 1. Algorithm 1 finds $\hat{u}(jT_u)$ by zero-order hold interpolation, where of course linear interpolation or higher-order splines could be used. However, simulations not included showed that this choice does not significantly affect the performance.

2.2. Interpolation in the convolution integral

Filtering of the continuous-time signal, u, yields

$$z(kT) = \int h(kT - \tau)u(\tau)d\tau, \tag{5}$$

For Problem 1, compute
(1) $\hat{z}(kT) = \sum_{m=1}^{M} \tau_m h(kT - t_m)u(t_m)$.

Algorithm 2: Convolution interpolation.

and using Riemann integration, we get Algorithm 2. The algorithm will be exact if the integrand, $h(kT - \tau)u(\tau)$, is constant between the sampling points, t_m, for all kT. As stated before, the error, when this is not the case, decreases when the ratio M/N increases.

This algorithm can be further analyzed using the inverse Fourier transform, and the results in [22], which will be done in Section 4.1.

Remark 2. Higher-order interpolations of (5) were studied in [23] without finding any benefits.

When the filter $h(t)$ is causal, the summation is only taken over m such that $t_m < kT$, and thus Algorithm 2 is ready for online use.

2.3. Interpolation in the frequency domain

LP-filtering is given by a multiplication in the frequency domain, and we can form the approximate Fourier transform (AFT), [22], given by Riemann integration of the Fourier transform, to get Algorithm 3. This is also a known approach in the uniform sampling case, where the DFT is used in each step. The AFT is formed for $2N$ frequencies to avoid circular convolution. This corresponds to zero-padding for uniform sampling. Then the inverse DFT gives the estimate.

Remark 3. The AFT used in Algorithm 3 is based on Riemann integration of the Fourier transform of $u(t)$, and would be exact whenever $u(t)e^{-i2\pi ft}$ is constant between sampling times, which of course is rarely the case. As for the two previous algorithms, the approximation is less grave for large enough M/N. This paper does not include an investigation of error bounds.

More investigations of the AFT were done in [22].

2.4. Complexity

In applications, implementation complexity is often an issue. We calculate the number of operations, N_{op}, in terms of additions (a), multiplications (m), and exponentials (e). As stated before, we have M measurements at non-uniform times, and want the signal value at N time points, equally spaced with T.

(i) Step (3) in Algorithm 1 is a linear filter, with one addition and one multiplication in each term,

$$N_{\text{op}}^1 = (1m + 1a)MN. \tag{6}$$

For Problem 1, compute
(1) $f_n = n/2NT$, $n = 0, \ldots, 2N-1$,
(2) $\hat{U}(f_n) = \sum_{m=1}^{M} \tau_m u(t_m) e^{-i2\pi f_n t_m}, n = 0, \ldots, N$,
(3) $\hat{Z}(f_n) = \hat{Z}(f_{2N-n})' = H(f_n)\hat{U}(f_n)$, $n = 0, \ldots, N$,
(4) $\hat{z}(kT) = 1/2NT \sum_{n=0}^{2N-1} \hat{Z}(f_n) e^{i2\pi kT f_n}$ $k = 0, \ldots, N-1$.
Here, $\hat{Z}'$ is the complex conjugate of $\hat{Z}$.

Algorithm 3: Frequency-domain interpolation.

Computing the convolution in step (3) in the frequency domain would require the order of $M \log_2(M)$ operations.

(ii) Algorithm 2 is similar to Algorithm 1,

$$N_{\text{op}}^2 = (2m+1a)MN, \tag{7}$$

where the extra multiplication comes from the factor τ_m.

(iii) Algorithm 3 performs an AFT in step (2), frequency-domain filtering in step (3) and an IDFT in step (4),

$$\begin{aligned} N_{\text{op}}^3 &= (2m + \text{l}e + \text{l}a)2M(N+1) \\ &\quad + (\text{l}m)(N+1) \\ &\quad + (\text{l}e + \text{l}m + \text{l}a)2N^2. \end{aligned} \tag{8}$$

Using the IFFT algorithm in step (4) would give $N\log_2(2N)$ instead, but the major part is still MN.

All three algorithms are thus of the order MN, though Algorithms 1 and 2 have smaller constants.

Studying work on efficient implementation, for example, [9], performance improvements, could be made also here, mainly for Algorithms 1 and 2, where the setup is similar.

Taking the length of the filter $h(t)$ into account can significantly improve the implementation speed. If the impulse response is short, the number of terms in the sums in Algorithms 1 and 2 will be reduced, as well as the number of extra frequencies needed in Algorithm 3.

3. NUMERIC EVALUATION

We will use the following example to test the performance of these algorithms. The signal consists of three frequencies that are drawn randomly for each test run.

Example 1. A signal with three frequencies, f_j, drawn from a rectangular distribution, Re, is simulated

$$s(t) = \sin(2\pi f_1 t - 1) + \sin(2\pi f_2 t - 1) + \sin(2\pi f_3 t), \tag{9}$$

$$f_j \in \text{Re}\left(0.01, \frac{1}{2T}\right), \quad j = 1, 2, 3. \tag{10}$$

The desired uniform sampling is given by the intersampling time $T = 4$ seconds. The non-uniform sampling is defined by

$$t_m = t_{m-1} + \tau_m, \tag{11}$$

$$\tau_m \in \text{Re}(t_l, t_h), \tag{12}$$

and the limits t_l and t_h are varied. In the simulation, N is set to 64 and the number of non-uniform samples are chosen so that $t_M > NT$ is assured. This is not in exact correspondence with the problem formulation, but assures that the results for different τ_m-distributions are comparable.

The samples are corrupted by additive measurement noise,

$$u(t_m) = s(t_m) + e(t_m), \tag{13}$$

where $e(t_m) \in N(0, \sigma^2), \sigma^2 = 0.1$.

The filter is a second-order LP-filter of Butterworth type with cutoff frequency $1/2T$, that is,

$$h(t) = \sqrt{2}\frac{\pi}{T} e^{-(\pi/T\sqrt{2})t} \sin\left(\frac{\pi}{T\sqrt{2}} t\right), \quad t > 0, \tag{14}$$

$$H(s) = \frac{(\pi/T)^2}{s^2 + \sqrt{2}\pi/Ts + (\pi/T)^2}. \tag{15}$$

This setup is used for 500 different realizations of f_j, τ_m, and $e(t_m)$.

We will test four different rectangular distributions (12):

$$\tau_m \in \text{Re}(0.1, 0.3), \quad \mu_T = 0.2, \quad \sigma_T = 0.06, \tag{16a}$$

$$\tau_m \in \text{Re}(0.3, 0.5), \quad \mu_T = 0.4, \quad \sigma_T = 0.06, \tag{16b}$$

$$\tau_m \in \text{Re}(0.4, 0.6), \quad \mu_T = 0.5, \quad \sigma_T = 0.06, \tag{16c}$$

$$\tau_m \in \text{Re}(0.2, 0.6), \quad \mu_T = 0.4, \quad \sigma_T = 0.12, \tag{16d}$$

and the mean values, μ_T, and standard deviations, σ_T, are shown for reference. For every run, we use the algorithms presented in the previous section and compare their results to the exact, continuous-time, result,

$$z(kT) = \int h(kT - \tau) s(\tau) d\tau. \tag{17}$$

We calculate the root mean square error, RMSE,

$$\lambda \triangleq \sqrt{\frac{1}{N} \sum_k |z(kT) - \hat{z}(kT)|^2}. \tag{18}$$

The algorithms are ordered according to lowest RMSE, (18), and Table 1 presents the result. The number of first, second and third positions for each algorithm during the 500 runs, are also presented. Figure 1 presents one example of the result, though the algorithms are hard to be separated by visual inspection.

A number of conclusions can be drawn from the previous example.

Table 1: RMSE values, λ in (18), for estimation of $\hat{z}(kT)$, in Example 1. The number of runs, where respective algorithm finished 1st, 2nd, and 3rd, is also shown.

		E[λ]	Std(λ)	1st	2nd	3rd
Setup in (16a)	Alg. 1	0.281	0.012	98	258	144
	Alg. 2	0.278	0.012	254	195	51
	Alg. 3	0.311	0.061	148	47	305
Setup in (16b)	Alg. 1	0.338	0.017	9	134	357
	Alg. 2	0.325	0.015	175	277	48
	Alg. 3	0.330	0.038	316	89	95
Setup in (16c)	Alg. 1	0.360	0.018	6	82	412
	Alg. 2	0.342	0.015	144	329	27
	Alg. 3	0.341	0.032	350	89	61
Setup in (16d)	Alg. 1	0.337	0.015	59	133	308
	Alg. 2	0.331	0.015	117	285	98
	Alg. 3	0.329	0.031	324	82	94

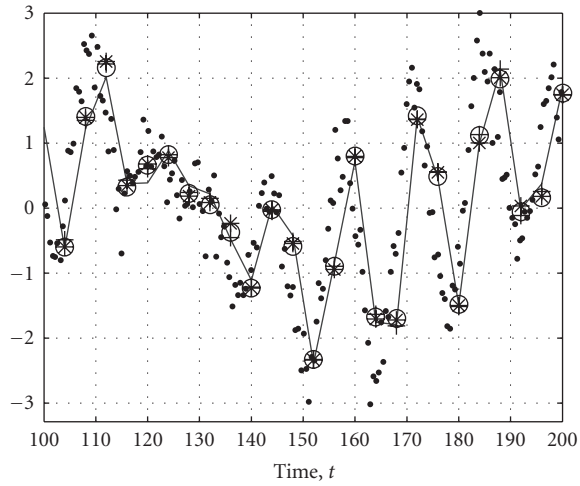

Figure 1: The result for the four algorithms, in Example 1, and a certain realization of (16c). The dots represent $u(t_m)$, and $z(kT)$ is shown as a line, while the estimates $\hat{z}(kT)$ are marked with a $*$ (Alg. 1), $\circ$ (Alg. 2) and + (Alg. 3), respectively.

(i) Comparing a given algorithm for different nonuniform sampling time pdf, Table 1 shows that $p_\tau(t)$, in (16a), (16b), (16c), (16d), has a clear effect on the performance.

(ii) Comparing the algorithms for a given sampling time distribution shows that the lowest mean RMSE is no guarantee of best performance at all runs. Algorithm 2 has the lowest E[λ] for setup (16a), but still performs worst in 10% of the cases, and for (16d), Algorithm 3 is number 3 in 20% of the runs, while it has the lowest mean RMSE.

(iii) Usually, Algorithm 3 has the lowest RMSE (1st position), but the spread is more than twice as large (standard deviation of λ), compared to the other two algorithms.

(iv) Algorithms 1 and 2 have similar RMSE statistics, though, of the two, Algorithm 2 performs slightly better in the mean, in all the four tested cases.

In this test, we find that Algorithm 3 is most often number one, but Algorithm 2 is almost as good and more stable in its performance. It seems that the realization of the frequencies, f_j, is not as crucial for the performance of Algorithm 2. As stated before, the performance also depends on the downsampling factor for all the algorithms.

The algorithms are comparable in performance and complexity. In the following, we focus on Algorithm 2, because of its nice analytical properties, its online compatibility, and, of course, its slightly better performance results.

4. THEORETIC ANALYSIS

Given the results for Algorithm 2 in the previous section, we will continue with a theoretic discussion of its behavior. We consider both finite time and asymptotic results. A small note is done on similar results for Algorithm 3.

4.1. Analysis of Algorithm 2

Here, we study the *a priori* stochastic properties of the estimate, $\hat{z}(kT)$, given by Algorithm 2. For the analytical calculations, we assume that the convolution is symmetric, and get

$$\begin{aligned}\hat{z}(kT) &= \sum_{m=1}^{M} \tau_m h(t_m) u(kT - t_m) \\ &= \sum_{m=1}^{M} \tau_m \int H(\eta) e^{i2\pi\eta t_m} d\eta \int U(\psi) e^{i2\pi\psi(kT-t_m)} d\psi \\ &= \iint H(\eta) U(\psi) e^{i2\pi\psi kT} \sum_{m=1}^{M} \tau_m e^{-i2\pi(\psi-\eta)t_m} d\psi\, d\eta \\ &= \iint H(\eta) U(\psi) e^{i2\pi\psi kT} W(\psi - \eta; t_1^M) d\psi\, d\eta \end{aligned} \tag{19}$$

with

$$W(f; t_1^M) = \sum_{m=1}^{M} \tau_m e^{-i2\pi f t_m}. \tag{20}$$

Let

$$\varphi_\tau(f) = E[e^{-i2\pi f\tau}] = \int e^{-i2\pi f\tau} p_\tau(\tau) d\tau = \mathcal{F}(p_\tau(t)) \tag{21}$$

denote the characteristic function for the sampling noise τ. Here, $\mathcal{F}$ is the Fourier transform operator. Then, [22, Theorem 2] gives

$$E[W(f)] = -\frac{1}{2\pi i} \frac{d\varphi_\tau(f)}{df} \frac{1 - \varphi_\tau(f)^M}{1 - \varphi_\tau(f)}, \tag{22}$$

where also an expression for the covariance, $\mathrm{Cov}(W(f))$, is given. The expressions are given by straightforward calculations using the fact that the sampling noise sequences τ_m are independent stochastic variables and $t_m = \sum_{k=1}^{m} \tau_k$ in (20).

These known properties of $W(f)$ make it possible to find $\mathrm{E}[\hat{z}(kT)]$ and $\mathrm{Var}(\hat{z}(kT))$ for any given characteristic function, $\varphi_\tau(f)$, of the sampling noise, τ_k.

The following lemma will be useful.

Lemma 1 (see [22, Lemma 1]). *Assume that the continuous-time function $h(t)$ with FT $H(f)$ fulfills the following conditions.*

(1) *$h(t)$ and $H(f)$ belong to the Schwartz class, $\mathcal{S}$.*[1]

(2) *The sum $g_M(t) = \sum_{m=1}^{M} p_m(t)$ obeys*

$$\lim_{M\longrightarrow\infty} \int g_M(t)h(t)dt = \int \frac{1}{\mu_T} h(t)dt = \frac{1}{\mu_T} H(0), \tag{23}$$

for this $h(t)$.

(3) *The initial value is zero, $h(0) = 0$.*

Then, it holds that

$$\lim_{M\longrightarrow\infty} \int \frac{1-\varphi_\tau(f)^M}{1-\varphi_\tau(f)} H(f)df = \frac{1}{\mu_T} H(0). \tag{24}$$

Proof. The proof is conducted using distributions from functional analysis and we refer to [22] for details. □

Let us study the conditions on $h(t)$ and $H(f)$ given in Lemma 1 a bit more. The restrictions from the Schwartz class could affect the usability of the lemma. However, all smooth functions with compact support (and their Fourier transforms) are in $\mathcal{S}$, which should suffice for most cases. It is not intuitively clear how hard (23) is. Note that, for any ARS case with continuous sampling noise distribution, $p_m(t)$ is approximately a Gaussian for higher m, and we can confirm that, for a large enough fixed t,

$$g_M(t) = \sum_{m=1}^{M} \frac{1}{\sqrt{2\pi m}\sigma_T} e^{-(t-m\mu_T)^2/2m\sigma_T^2} \longrightarrow \frac{1}{\mu_T}, \quad M \longrightarrow \infty, \tag{25}$$

with μ_T and σ_T being the mean and the standard deviation of the sampling noise τ, respectively. The integral in (23) can then serve as some kind of mean value approximation, and the edges of $g_N(t)$ will not be crucial. Also, condition 3 further restricts the behavior of $h(t)$ for small t, which will make condition 2 easier to fulfill.

Theorem 1. *The estimate given by Algorithm 2 can be written as*

$$\hat{z}(kT) = \widetilde{h} \star u(kT), \tag{26a}$$

where $\widetilde{h}(t)$ is given by

$$\widetilde{h}(t) = \mathcal{F}^{-1}(H \star W(f))(t), \tag{26b}$$

with $W(f)$ as in (20).

Furthermore, if the filter $h(t)$ and signal $u(t)$ belong to the Schwartz class, then $\mathcal{S}$ [24],

$$E\hat{z}(kT) \longrightarrow z(kT) \quad \text{if} \sum_{m=1}^{M} p_m(t) \longrightarrow \frac{1}{\mu_T}, M \longrightarrow \infty, \tag{26c}$$

$$E\hat{z}(kT) = z(kT) \quad \text{if} \sum_{m=1}^{M} p_m(t) = \frac{1}{\mu_T}, \forall M, \tag{26d}$$

with $\mu_T = E[\tau_m]$, and $p_m(t)$ is the pdf for time t_m.

Proof. First of all, (5) gives

$$z(kT) = \int H(\psi)U(\psi)e^{i2\pi\psi kT}d\psi, \tag{27a}$$

and from (19), we get

$$\hat{z}(kT) = \int U(\psi)e^{i2\pi\psi kT} \underbrace{\int H(\eta)W(\psi-\eta)d\eta}_{\triangleq \widetilde{H}(\psi)} d\psi \tag{27b}$$

which implies that we can identify $\widetilde{H}(f)$ as the filter operation on the continuous-time signal $u(t)$, and (26a) follows.

From Lemma 1 and (22), we get

$$\begin{aligned} &\int \mathrm{E}[W(f)]y(f)df \\ &\quad = \int \mathrm{E}[\tau e^{-i2\pi f\tau}] \frac{1-\varphi_\tau(f)^M}{1-\varphi_\tau(f)} y(f)df \longrightarrow y(0) \end{aligned} \tag{28}$$

for any function $y(f)$ fulfilling the properties of Lemma 1. This gives

$$\begin{aligned} \mathrm{E}[\hat{z}(kT)] &= \iint H(\eta)U(\psi)e^{i2\pi\psi kT}\,\mathrm{E}[W(\psi-\eta)]d\psi\, d\eta \\ &\longrightarrow \int H(\psi)U(\psi)e^{i2\pi\psi kT}\, d\psi \\ &= z(kT), \end{aligned} \tag{29}$$

when $H(f)$ and $U(f)$ behave as requested, and (26c) follows.

Using the same technique as in the proof of Lemma 1, (26d) also follows. □

From the investigations in [22], it is clear that $\widetilde{H}(f)$, in (27b), is the AFT of the sequence $h(t_m)$ (cf. the AFT of $u(t_m)$ in step (2) of Algorithm 3).

Requiring that both $h(t)$ and $u(t)$ be in the Schwartz class is not, as indicated before, a major restriction. Though, some thought needs to be done for each specific case before applying the theorem.

Algorithm 3 can be investigated analogously.

Theorem 2. *The estimate given by Algorithm 3 can be written as*

$$\hat{z}(kT) = \widetilde{h} \star u(kT), \tag{30a}$$

[1] $h \in \mathcal{S} \Leftrightarrow t^k h^{(l)}(t)$ is bounded, that is, $h^{(l)}(t) = \theta(|t|^{-k})$, for all $k, l \geq 0$.

where $\widetilde{h}(t)$ *is given by the inverse Fourier transform of*

$$\widetilde{H}(f) = \frac{1}{2NT}\left(\sum_{n=0}^{N-1} H(f_n)e^{-i2\pi(f-f_n)kT}W(f_n - f) + \sum_{n=N}^{2N-1} H(-f_n)e^{-i2\pi(f-f_n)kT}W(-f_n - f)\right), \tag{30b}$$

and $W(f)$ *was given in* (20).

Proof. First, observe that real signals $u(t)$ and $h(t)$ give $U(f)' = U(-f)$ and $H(f)' = H(-f)'$, respectively, the rest is completely in analogue with the proof of Theorem 1, with one of the integrals replaced with the corresponding sum. □

Numerically, it is possible to confirm that the requirements on $p_m(t)$, in (26c), are true for Additive Random Sampling, since $p_m(t)$ then converges to a Gaussian distribution with $\mu = m\mu_T$ and $\sigma^2 = m\sigma_T$. A smooth filter with compact support is noncausal, but with finite impulse response (see e.g., the optimal filter discussed in Section 5).

A noncausal filter is desired in theory, but often not possible in practice. The performance of $\hat{z}_{BW}(kT)$ compared to the optimal noncausal filter in Table 2 is thus encouraging.

4.2. Illustration of Theorem 1

Theorem 1 shows that the originally designed filter $H(f)$ is effectively replaced by another linear filter $\widetilde{H}(f)$ when using Algorithm 2. Since $\widetilde{H}(f)$ only depends on $H(f)$ and the realization of the sampling times t_m, we here study $\widetilde{H}(f)$ to exclude the effects of the signal, $u(t)$, on the estimate.

First, we illustrate (26b) by showing the influence of the sampling times, or, the distribution of τ_m, on $\mathrm{E}[\widetilde{H}]$. We use the four different sampling noise distributions in (16a), (16b), (16c), (16d), using the original filter $h(t)$ from (14) with $T = 4$ seconds. Figure 2 shows the different filters $\widetilde{H}(f)$, when the sampling noise distribution is changed. We conclude that both the mean and the variance affect $|\mathrm{E}[\widetilde{H}(f)]|$, and that it seems possible to mitigate the static gain offset from $\widetilde{H}(f)$ by multiplying $\hat{z}(kT)$ with a constant depending on the filter $h(t)$ and the sampling time distribution $p_\tau(t)$.

Second, we show that $E\widetilde{H} \longrightarrow H$ when M increases, for a smooth filter with compact support, (26c). Here, we use

$$h(t) = \frac{1}{4d_f}\cos\left(\frac{\pi}{2}\frac{t - d_f T}{d_f T}\right)^2, \quad |t - d_f T| < d_f T, \tag{31}$$

where d_f is the width of the filter. The sampling noise distribution is given by (16b). Figure 3 shows an example of the sampled filter $h(t_m)$.

To produce a smooth causal filter, the time shift $d_f T$ is used. This in turn introduces a delay of $d_f T$ in the resampling procedure. We choose the scale factor $d_f = 8$ for a better view of the convergence (higher d_f gives slower convergence). The width of the filter covers approximately $2d_f T/\mu_T = 160$ nonuniform samples, and more than 140 of them are needed for a close fit also at higher frequencies. The convergence of the magnitude of the filter is clearly shown in Figure 4.

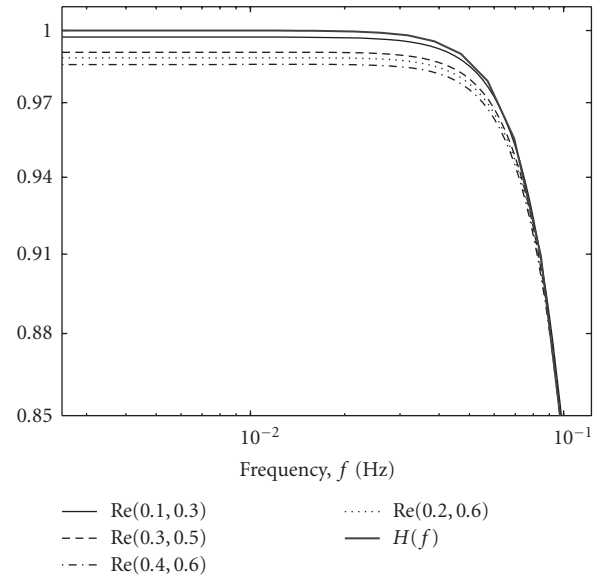

Figure 2: The true $H(f)$ (thick line) compared to $|\mathrm{E}[\widetilde{H}(f)]|$ given by (14) and Theorem 1, for the different sampling noise distributions in (16a), (16b), (16c), (16d), and $M = 250$.

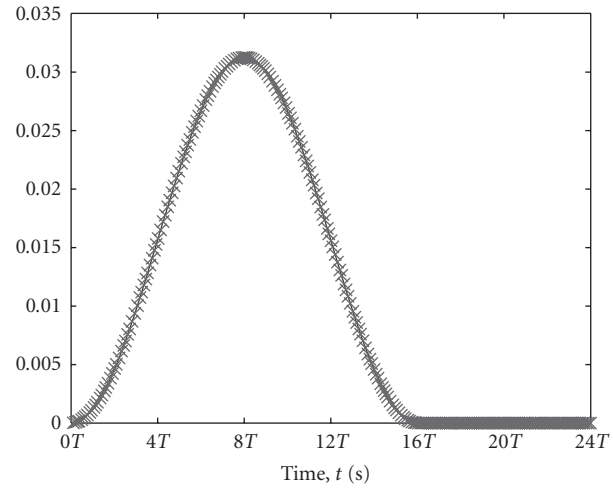

Figure 3: An example of the filter $h(t_m)$ given by (31), (16b), and $T = 4$.

5. APPLICATION EXAMPLE

As a motivating example, consider the ubiquitous wheel speed signals in vehicles that are instrumental for all driver assistance systems and other driver information systems. The wheel speed sensor considered here gives $L = 48$ pulses per revolution, and each pulse interval can be converted to a wheel speed. With a wheel radius of $1/\pi$, one gets 24ν pulses per second. For instance, driving at $\nu = 25$ m/s (90 km/h) gives an average sampling rate of 600 Hz. This illustrates the need for speed-adaptive downsampling.

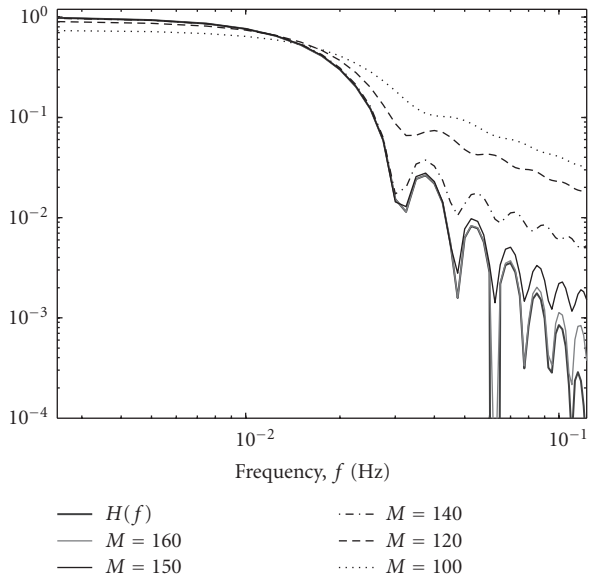

FIGURE 4: The true filter, $H(f)$ (thick line), compared to $|\mathrm{E}[\widetilde{H}(f)]|$ for increasing values of M, when $h(t)$ is given by (31).

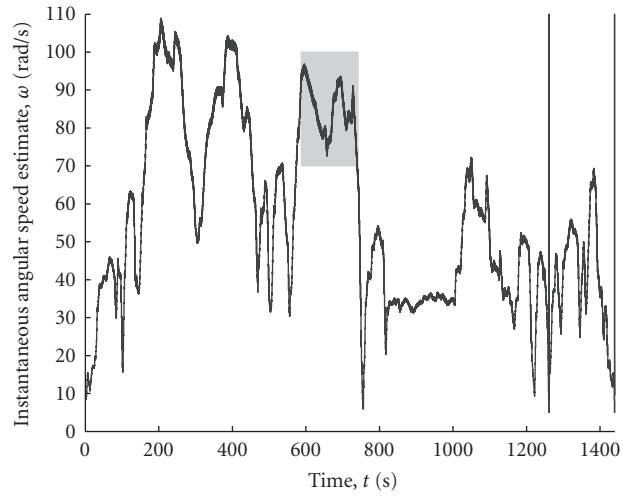

FIGURE 5: The data from a wheel speed sensor of a car. The data in the gray area is chosen for further evaluation. It includes more than 600 000 measurements.

Example 2. Data from a wheel speed sensor, like the one discussed above, have been collected. An estimate of the angular speed, $\omega(t)$,

$$\hat{\omega}(t_m) = \frac{2\pi}{L(t_m - t_{m-1})}, \tag{32}$$

can be computed at every sample time. The average intersampling time, t_M/M, is 2.3 milliseconds for the whole sampling set. The set is shown in Figure 5. We also indicate the time interval where the following calculations are performed. A sampling time $T = 0.1$ second gives a signal suitable for driver assistance systems.

We begin with a discussion on finding the underlying signal in an offline setting and then continue with a test of different online estimates of the wheel speed.

5.1. The optimal nonparametric estimate

For the data set in Example 2, there is no true reference signal, but in an offline test like this, we can use computationally expensive methods to compute the best estimate. For this application, we can assume that the measurements are given by

$$u(t_m) = s(t_m) + e(_m) \tag{33}$$

with

(i) independent measurement noise, $e(t_m)$, with variance σ^2, and

(ii) bounded second derivative of the underlying noise-free function $s(t)$, that is,

$$\left| s^{(2)}(t) \right| < C, \tag{34}$$

which in the car application means limited acceleration changes.

Under these conditions, the work by [19] helps with optimally estimating $z(kT)$. When estimating a function value $z(kT)$ from a sequence $u(t_m)$ at times t_m, a local weighted linear approximation is investigated. The underlying function is approximated locally with a linear function

$$m(t) = \theta_1 + (t - kT)\theta_2, \tag{35}$$

and $m(kT) = \theta_1$ is then found from minimization,

$$\hat{\theta} = \arg\min_{\theta} \sum_{m=1}^{M} \left(u(t_m) - m(t_m)\right)^2 K_B(t_m - kT), \tag{36}$$

where $K_B(t)$ is a kernel with bandwidth B, that is, $K_B(t) = 0$ for $|t| > B$. The *Epanechnikov kernel*,

$$K_B(t) = \left(1 - \left(\frac{t}{B}\right)^2\right)_+, \tag{37}$$

is the optimal choice for interior points, $t_1 + B < kT < t_M - B$, both in minimizing MSE and error variance. Here, subscript + means taking the positive part. This corresponds to a noncausal filter for Algorithm 2.

This gives the optimal estimate $\hat{z}_{\mathrm{opt}}(kT) = \hat{\theta}_1$, using the noncausal filter given by (35)–(37) with $B = B_{opt}$ from [19],

$$B_{\mathrm{opt}} = \left(\frac{15\sigma^2}{C^2 MT/\mu_T}\right)^{1/5}. \tag{38}$$

In order to find B_{opt}, the values of σ^2, C, and μ_T were roughly estimated from data in each interval $[kT - T/2, kT + T/2]$ and a mean value of the resulting bandwidth was used for B_{opt}.

The result from the optimal filter, $\hat{z}_{\mathrm{opt}}(kT)$, is shown compared to the original data in Figure 6, and it follows a smooth line nicely.

For end points, that is, a causal filter, the error variance is still minimized by said kernel, (37), restricted to $t \in [-B, 0]$,

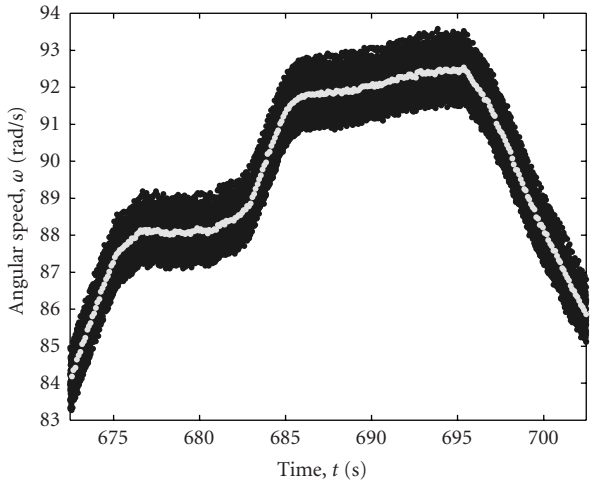

FIGURE 6: The cloud of data points, $u(t_m)$ black, from Example 2, and the optimal estimates, $z_{\text{opt}}(kT)$ gray. Only part of the shaded interval in Figure 5 is shown.

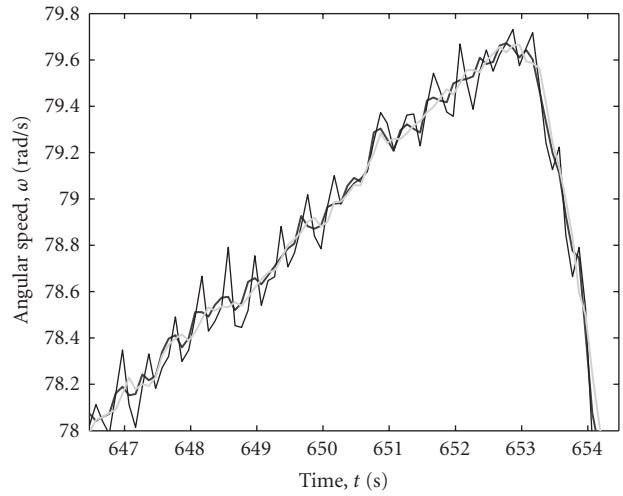

FIGURE 7: A comparison of three different estimates for the data in Example 2: optimal $z_{\text{opt}}(kT)$ (thick line), casual "optimal" $z_E(kT)$ (thin line), and causal Butterworth $z_{BW}(kT)$ (gray line). Only a part of the shaded interval in Figure 5 is shown.

but $K_B(t) = (1 - t/B)_+, t \in [-B, 0]$ is optimal in MSE sense. Fan and Gijbels still recommend to always use the Epanechnikov kernel, because of both performance and implementation issues. [19] does not include a result for the optimal bandwidth in this case. In our test we need a causal filter and then choose $B = 2B_{\text{opt}}$ in order to include the same number of points as in the noncausal estimate.

5.2. Online estimation

The investigation in the previous section gives a reference value to compare the online estimates to. Now, we test four different estimates:

(i) $\hat{z}_E(kT)$: the casual filter given by (35), (36), the kernel (37) for $-B < t \leq 0$ and $B = 2B_{\text{opt}}$;
(ii) $\hat{z}_{BW}(kT)$: a causal Butterworth filter, $h(t)$, in Algorithm 2; the Butterworth filter is of order 2 with cutoff frequency $1/2T = 5$ Hz, as defined in (14),
(iii) $\hat{z}_m(kT)$: the mean of $u(t_m)$ for $t_m \in [kT - T/2, kT + T/2]$;
(iv) $\hat{z}_n(kT)$: a nearest neighbor estimate;

and compare them to the optimal $z_{\text{opt}}(kT)$. The last two estimates are included in order to show if the more clever estimates give significant improvements. Figure 7 shows the first two estimates, $z_E(kT)$ and $z_{BW}(kT)$, compared to the optimal $z_{\text{opt}}(kT)$.

Table 2 shows the root mean square errors compared to the optimal estimate, $\hat{z}_{\text{opt}}(kT)$, calculated over the interval indicated in Figure 5. From this, it is clear that the casual "optimal" filter, giving $\hat{z}_E(kT)$, needs tuning of the bandwidth, B, since the literature gave no result for the optimal choice of B in this case. Both the filtered estimates, $z_E(kT)$ and $z_{BW}(kT)$, are significantly better than the simple mean, $z_m(kT)$. The Butterworth filter performs very well, and is also much less computationally complex than using the Epanechnikov kernel. It is promising that the estimate from Algorithm 2, $\hat{z}_{BW}(kT)$, is close to $\hat{z}_{\text{opt}}(kT)$, and it encourages future investigations.

TABLE 2: RMSE from optimal estimate, $\sqrt{E[|\hat{z}_{\text{opt}}(kT) - \hat{z}_*(kT)|^2]}$, in Example 2.

Casual "optimal" $\hat{z}_E(kT)$	Butterworth $\hat{z}_{BW}(kT)$	Local mean $\hat{z}_m(kT)$	Nearest neighbor $\hat{z}_n(kT)$
0.0793	0.0542	0.0964	0.3875

6. CONCLUSIONS

This work investigated three different algorithms for downsampling non-uniformly sampled signals, each using interpolation on different levels. Two algorithms are based on existing techniques for uniform sampling with interpolation in time and frequency domain, while the third alternative is truly non-uniform where interpolation is made in the convolution integral. The results in the paper indicate that this third alternative is preferable in more than one way.

Numerical experiments presented the root mean square error, RMSE, for the three algorithms, and convolution interpolation has the lowest mean RMSE together with frequency-domain interpolation. It also has the lowest standard deviation of the RMSE together with time-domain interpolation.

Theoretic analysis showed that the algorithm gives asymptotically unbiased estimates for noncausal filters. It was also possible to show how the actual filter implemented by the algorithm was given by a convolution in the frequency domain with the original filter and a window depending only on the sampling times.

In a final example with empirical data, the algorithm gave significant improvement compared to the simple local mean estimate and was close to the optimal nonparameteric estimate that was computed beforehand.

Thus, the results are encouraging for further investigations, such as approximation error analysis and search for optimality conditions.

ACKNOWLEDGMENTS

The authors wish to thank NIRA Dynamics AB for providing the wheel speed data, and Jacob Roll, for interesting discussions on optimal filtering. Part of this work was presented at EUSIPCO07

REFERENCES

[1] A. Aldroubi and K. Gröchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *SIAM Review*, vol. 43, no. 4, pp. 585–620, 2001.

[2] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, New York, NY, USA, 1998.

[3] T. A. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 577–591, 1984.

[4] A. I. Russell and P. E. Beckmann, "Efficient arbitrary sampling rate conversion with recursive calculation of coefficients," *IEEE Transactions on Signal Processing*, vol. 50, no. 4, pp. 854–865, 2002.

[5] A. I. Russell, "Efficient rational sampling rate alteration using IIR filters," *IEEE Signal Processing Letters*, vol. 7, no. 1, pp. 6–7, 2000.

[6] T. Saramäki and T. Ritoniemi, "An efficient approach for conversion between arbitrary sampling frequencies," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '96)*, vol. 2, pp. 285–288, Atlanta, Ga, USA, May 1996.

[7] F. J. Beutler, "Error-free recovery of signals from irregularly spaced samples," *SIAM Review*, vol. 8, no. 3, pp. 328–335, 1966.

[8] F. Marvasti, M. Analoui, and M. Gamshadzahi, "Recovery of signals from nonuniform samples using iterative methods," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 872–878, 1991.

[9] A. I. Russell, "Regular and irregular signal resampling," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2002.

[10] F. Marvasti, Ed., *Nonuniform Sampling: Theory and Practice*, Kluwer Academic Publishers, Boston, Mass, USA, 2001.

[11] P. J. Ferreira, "Iterative and noniterative recovery of missing samples for 1-D band-limited signals," in *Sampling: Theory and Practice*, F. Marvasti, Ed., chapter 5, pp. 235–282, Kluwer Academic Publishers, Boston, Mass, USA, 2001.

[12] B. Lacaze, "Reconstruction of stationary processes sampled at random times," in *Nonuniform Sampling: Theory and Practice*, F. Marvasti, Ed., chapter 8, pp. 361–390, Kluwer Academic Publishers, Boston, Mass, USA, 2001.

[13] H. G. Feichtinger and K. Gröchenig, "Theory and practice of irregular sampling," in *Wavelets: Mathematics and Applications*, J. J. Benedetto and M. W. Frazier, Eds., pp. 305–363, CRC Press, Boca Raton, Fla, USA, 1994.

[14] Y. C. Eldar, "Sampling with arbitrary sampling and reconstruction spaces and oblique dual frame vectors," *Journal of Fourier Analysis and Applications*, vol. 9, no. 1, pp. 77–96, 2003.

[15] K. Yao and J. B. Thomas, "On some stability and interpolatory properties of nonuniform sampling expansions," *IEEE Transactions on Circuits and Systems*, vol. 14, no. 4, pp. 404–408, 1967.

[16] M. Bourgeois, F. Wajer, D. van Ormondt, and F. Graveron-Demilly, "Reconstruction of MRI images from non-uniform sampling and its application to intrascan motion correction in functional MRI," in *Modern Sampling Theory*, J. J. Benedetto and P. J. Ferreira, Eds., chapter 16, pp. 343–363, Birkhäuser, Boston, Mass, USA, 2001.

[17] S. R. Dey, A. I. Russell, and A. V. Oppenheim, "Precompensation for anticipated erasures in LTI interpolation systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 325–335, 2006.

[18] P. J. Ferreira, "Nonuniform sampling of nonbandlimited signals," *IEEE Signal Processing Letters*, vol. 2, no. 5, pp. 89–91, 1995.

[19] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications*, Chapman & Hall, London, UK, 1996.

[20] A. Papoulis, *Signal Analysis*, McGraw-Hill, New York, NY, USA, 1977.

[21] M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[22] F. Eng, F. Gustafsson, and F. Gunnarsson, "Frequency domain analysis of signals with stochastic sampling times," to appear in *IEEE Transactions on Signal Processing*.

[23] F. Gunnarsson and F. Gustafsson, "Frequency analysis using non-uniform sampling with application to active queue management," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04*, vol. 2, pp. 581–584, Montreal, Canada, May 2004.

[24] C. Gasquet and P. Witkomski, *Fourier Analysis and Applications*, Springer, New York, NY, USA, 1999.