Random Sampling for Analog-to-Information Conversion of Wideband Signals

Jason Laska, Sami Kirolos, Yehia Massoud, Richard Baraniuk
Department of Electrical and Computer Engineering
Rice University
Houston, TX

Anna Gilbert, Mark Iwen, Martin Strauss
Departments of Mathematics and EECS
University of Michigan
Ann Arbor, MI

***Abstract*— We develop a framework for analog-to-information conversion that enables sub-Nyquist acquisition and processing of wideband signals that are sparse in a local Fourier representation. The first component of the framework is a random sampling system that can be implemented in practical hardware. The second is an efficient information recovery algorithm to compute the spectrogram of the signal, which we dub the *sparsogram*. A simulated acquisition of a frequency hopping signal operates at $33\times$ sub-Nyquist average sampling rate with little degradation in signal quality.**

I. Introduction

Sensors, signal processing hardware, and algorithms are under increasing pressure to accommodate ever faster sampling and processing rates. In this paper we study the acquisition and analysis of wideband signals that are *locally Fourier sparse* (LFS) in the sense that at each point in time they are well-approximated by a few local sinusoids of constant frequency. Examples of LFS signals include frequency hopping communication signals, slowly varying chirps from radar and geophysics, and many acoustic and audio signals.

LFS signals are sparse in a time-frequency representation like the short-time Fourier transform (STFT). In discrete time, the STFT corresponds to a Fourier analysis on a sliding window of the signal

$$S(\tau,\omega) = \sum_{\nu=-\infty}^{\infty} \mathbf{s}(\nu)\,\mathbf{w}(\nu-\tau)\,e^{-j\omega\nu}; \qquad \text{(I.1)}$$

that is, $S(\tau,\omega)$ is the Fourier spectrum of the signal $\mathbf{s}$ localized around time sample τ by the N-point window $\mathbf{w}$. The *spectrogram* is the squared magnitude $|S(\tau,\omega)|^2$. The defining property of an LFS signal is that $S(\tau,\omega) \approx 0$ for most τ and ω.

While LFS signals are simply described in time-frequency, they are *wideband* when there is no a priori restriction on the frequencies of the local sinusoids. Hence, the requirements of traditional Nyquist-rate sampling at two times the bandwidth can be excessive and difficult to meet. As a practical example, consider sampling a frequency-hopping communications signal that consists of a sequence of windowed sinusoids with frequencies distributed between f_1 and f_2 Hz. The bandwidth of this signal is $f_2 - f_1$ Hz, which dictates sampling above the Nyquist rate of $2(f_2 - f_1)$ Hz to avoid aliasing. However the description of the signal at any point in time is extremely simple: it consists of just a single sinusoid. Surely we should be able to acquire such a signal with fewer than $2(f_2 - f_1)$ samples per second.

Fortunately, the past several years have seen several advances in the theory of sampling and reconstruction that address these very questions. Leveraging the theory of *streaming algorithms*, we introduce in this paper a new framework for *analog-to-information* conversion that enables sub-Nyquist acquisition and processing of LFS signals. Our framework has two key components. The first is a *random sampling system* that can be implemented in practical hardware. The second is an efficient *information recovery algorithm* to compute the spectrogram of LFS signal, which we dub the *sparsogram*.

This paper is organized as follows. We review the requisite theory and algorithms for random sampling in Section II and develop two implementations of random samplers in Section III. We conduct a number of experiments to validate our approach in Section IV and close with a discussion and conclusions in Section V.

II. Random Sampling and Information Recovery

To set the stage for the random sampling and information recovery algorithm, we start with a description of the problem in the discrete setting. Let $\mathbf{s}$ be a discrete-time signal of length N (not discrete samples of an analog signal but simply a vector of length N). We know that $\mathbf{s}$ is perfectly represented by its Fourier coefficients or spectrum. Suppose that we wish to use only m Fourier coefficients to represent the signal or spectrum. To minimize the MSE in our choice of m coefficients, the optimal choice of Fourier coefficients is the m largest Fourier coefficients in magnitude. Denote the optimal m-term Fourier representation of a signal $\mathbf{s}$ of length N by $\mathbf{r}_{\text{opt}}$. We assume that, for some M, we have

$$(1/M) \leq \|\mathbf{s} - \mathbf{r}_{\text{opt}}\|_2 \leq \|\mathbf{s}\|_2 \leq M.$$

Gilbert et. al [1] have developed an algorithm that uses at most $m \cdot (\log(1/\delta), \log N, \log M, 1/\epsilon)^{O(1)}$ space and time and outputs a representation $\mathbf{r}$ such that

$$\|\mathbf{s} - \mathbf{r}\|_2^2 \leq (1+\epsilon)\|\mathbf{s} - \mathbf{r}_{\text{opt}}\|_2^2,$$

with probability at least $1 - \delta$.

The algorithm is randomized and the probability is over random choices made by the algorithm, not over the signal.

That is, it is a coin-flipping algorithm that uses coin-flips to choose the sample set upon which we observe the signal. For each signal, with high probability, the algorithm succeeds.

As a bonus, the running time of the algorithm is much less than N, the length of the signal. This is exponentially faster than the running time of FFT (which is $O(N \log N)$). One reason the algorithm is so much faster is that it does not read all the input; we sample exponentially fewer positions than the length of the signal and we assume that our algorithm can read $\mathbf{s}[t]$ for a t of its choice in constant time. The set of samples t where the algorithm observes the signal is chosen randomly (from a non-uniform distribution), and the set does not adapt to the signal.

Let us describe the distribution on sample points before we discuss how the information recovery algorithm uses these signal samples. Let us assume for simplicity that N is a prime number. We generate two uniformly random integers r and s from $[0, \ldots, N-1]$ with the stipulation that s is non-zero. For input parameter m, we generate the set $A = r + k \cdot s \mod N$ where $k = 0, \ldots, 4m$. The set A is a *random arithmetic progression* of length $4m$. For each point p in the arithmetic progression, we also sample at $(p + 2^l) \mod N$ where $l = 0, \ldots, \log_2 N$. For example, we observe the signal at a point p and at $p + N/2$. Thus, our sample set is

$$\mathcal{S} = \{(p + 2^l) \mod N | p \in A, l = 0, \ldots, \log_2 N\}$$

where A is a random arithmetic progression. To meet the accuracy and success probability guarantees stated above, we take multiple independent sample sets drawn from this distribution. The number of independent sample sets depends on the desired accuracy and success probability. Note that with low probability, this distribution generates samples points that are separated by distance 1. In other words, the probability that we choose consecutive sample points is low and can be kept under control.

Once we have specified the sample set, our algorithm proceeds in a greedy fashion. Given a signal $\mathbf{s}$, we set the representation $\mathbf{r}$ to zero and consider the residual $\mathbf{s} - \mathbf{r}$. We make progress on lowering $\|\mathbf{s} - \mathbf{r}\|$ by sampling from the residual, identifying a set of "significant" frequencies in the spectrum of the residual, estimating the Fourier coefficients of these "significant" frequencies, and subtracting their contribution, thus forming a new residual signal upon which we iterate. Below, we sketch each iteration of our algorithm. The details may be found in [1].

Each iteration of our algorithm proceeds as follows:

- SAMPLE from $\mathbf{s} - \mathbf{r}$ in $K \approx m$ correlated random positions, where $\mathbf{r}$ has L terms, in total time $(K + L) \log^{O(1)}(N)$. We take the given samples from the signal $\mathbf{s}$ on the sample set and sample from the representation $\mathbf{r}$ by performing an unequally-spaced fast Fourier transform.
- IDENTIFY a set of "significant" frequencies in the spectrum of $\mathbf{s} - \mathbf{r}$.
  - ISOLATE one or more modes of $\mathbf{s} - \mathbf{r}$. We generate a set $\{\mathbf{F}_k : k < K\}$ of K new signals from $\mathbf{s} - \mathbf{r}$ where $K \leq O(1/\eta)$ is sufficiently large, so that each ω that is η-significant in $\mathbf{s} - \mathbf{r}$ is likely to be $(1-\gamma)$-significant in some $\mathbf{F}_k$ for some small constant γ. To do this, we
    - PERMUTE the spectrum of $\mathbf{s} - \mathbf{r}$ by a random dilation σ, getting $\mathrm{perm}\sigma(\mathbf{s} - \mathbf{r})$. Observe that we can sample from the signal $\mathrm{perm}\mathbf{s}(\mathbf{s} - \mathbf{r})$ by modulating the (time-domain) samples of this signal thus keeping our time and memory requirements low.
    - FILTER $\mathrm{perm}\sigma(\mathbf{s} - \mathbf{r})$ by a filterbank of approximately m equally-spaced frequency-domain translations of the Boxcar Filter with bandwidth approximately N/m and approximately m common taps. Again, this procedure can be done on the the (time-domain) samples within the advertised time and memory constraints. We obtain m new signals, some of which have a single overwhelming Fourier mode, $\sigma\omega$, corresponding to significant mode ω in $\mathbf{s} - \mathbf{r}$.
    - We undo the above permutation, thereby making ω overwhelming instead of $\sigma\omega$.
  - GROUP-TEST each new signal to locate the one overwhelming mode, ω. Learn the bits of ω one at a time, least to most significant. For example, to learn the least significant bit, we project each $\mathbf{F}_k$ onto the space of even frequencies and the space of odd frequencies. Next, we estimate (via our samples) the energy of each projection and learn the least significant bit of ω. Observe that we have can carry out this projection as we have sampled our signal at a random point p and at $p + N/2$.
- ESTIMATE the Fourier coefficients of these "significant" frequencies by computing the Fourier coefficients of the sampled residual using an unequally-spaced fast Fourier transform algorithm and normalizing appropriately.
- ITERATE in a greedy pursuit.

The above discussion assumes that the signal $\mathbf{s}$ is a discrete vector of length N. A naïve use of this algorithm would sample a wideband analog signal at Nyquist rate and then input this vector of discrete samples to the random sampling algorithm. This naïve use, however, does not take advantage of the strength of the algorithm, namely that we do not need to use all of the samples to recovery information about the signal. Instead, the algorithm and its analysis suggest that we can subsample the wideband analog signal at an average rate that is significantly lower than the Nyquist rate and use those samples to recover significant information about the spectrum of the signal in considerably less time than it would take to perform an FFT on all of the samples (taken at Nyquist rate). Application of the algorithm on local (Boxcar) windows of the signal yields the sparsogram coefficients.

III. IMPLEMENTATION

The random sampling algorithm of Section II suggests that we can build a *Random Analog to Digital Converter* (RADC)

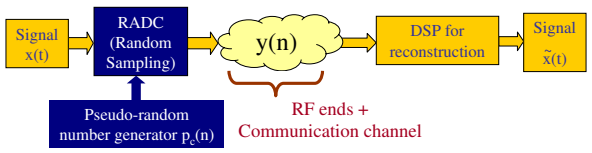

Fig. 1. Random Analog to Digital Converter (RADC) system.

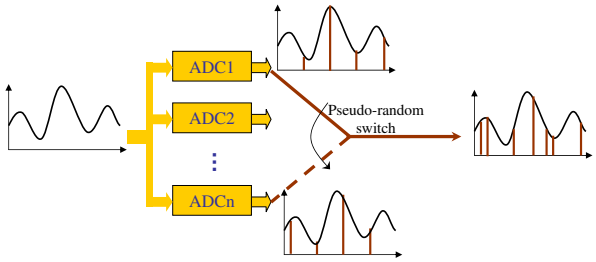

Fig. 2. First implementation of a RADC.

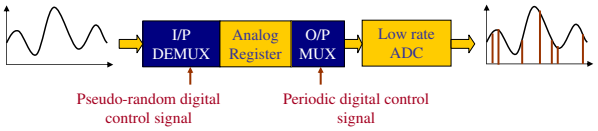

Fig. 3. Second implementation of a RADC.

for analog-to-information conversion that recovers significant information about LFS wideband signals from just a few measurements and with low computational complexity. There are, however, considerable practical challenges to implementing this vision. The RADC system shown in Figure 1 consists of a random sampler followed by a low-rate analog-to-digital converter (ADC). The random sampler is a challenging block to design, since the sampling pattern may call for consecutive samples that are closely space in time and almost at the Nyquist rate. In this section, we discuss two prototype implementations of the random sampler and discuss the different design issues and challenges in each implementation.

Our first prototype RADC implementation uses a parallel bank of low-rate ADCs that have equal shifts between their starting conversion points. This creates a shift in the samples that are produced from each of the parallel ADCs. We then control the switching mechanism among their outputs pseudo-randomly as shown in Figure 2. However, this approach would occupy a large chip area for the many ADCs. This implementation also faces the challenge of minimizing the jitter effect when controlling the switches [2]. The *aperture jitter* is the uncertainty in the switching time of the sampling switches, which typically leads to uncertainty in the instances of the time samples. Therefore, the time between consecutive samples in converting the analog to digital bits must be long enough to cover the aperture jitter. This in turn poses a limit on the maximum clock frequency as well as the maximum data rate [3]. For typical designs, the clock jitter should not exceed 10% of the clock period.

The second prototype implementation employs an analog register (a group of capacitors) and an analog demultiplexer/multiplexer (digitally controlled commutators) as shown in Figure 3. This implementation for the random sampler requires a successive low-rate ADC. This implementation focuses on eliminating the unnecessary storage of the analog signal. The demultiplexer (input commutator) is controlled by a pseudo-random digital control signal with average rate corresponding to the measurement rate, while the output multiplexer is controlled by a periodic signal with the same rate (measurement rate). The storage control signal is pseudo-random in order to sample the signal at pseudo-random time instances. The read signal is periodic to give the ADC enough time to complete its process. The length (storage capacity) of the analog register is a function of the probability of having close sampling times and the average rate of the measurement. We need the highest frequency of the pseudo-random clock to be greater than or equal to the Nyquist rate of the signal to be able to capture very close samples (although, these close samples occur with low probability).

Maximal-Length Linear Feedback Shift Registers (MLFSR) are good candidates for generating the pseudo-random control sequences for the input commutator [4]. The MLFSR has the benefit of providing a random sequence of 0s and 1s with balanced average, while offering the possibility of regenerating the same sequence knowing its initial seed. This feature allows the decoder to re-generate the pseudo-random sequence in order to use it for the information recovery of the sparsogram. The MLFSR is reset to its initial state every time frame, which is the period of time that is captured from the simulations and fed to the frame-based reconstruction algorithm.

IV. Experimental Results

In our first experiment, we consider a frequency-hopping signal submerged in additive white Gaussian noise (AWGN). We make a small number of samples in each window and return the frequency present (and its negation since the signal is real-valued). Figure 4 shows the sparsogram generated by our algorithm. For the sake of comparison, we also use FFTW 3.1 [5] to produce the spectrogram using Nyquist-rate samples (FFTW 3.1 is the state-of-the-art FFT implementation). This approach needs 100% of the Nyquist samples and runs in time $O(N \log N)$. In contrast, the RADC/sparsogram needs only 3% as many samples; in other words, we are sampling at an average of $33\times$ sub-Nyquist. Furthermore, the sampling algorithm runs in comparable time to FFTW 3.1.

Next, we generate 1000 instances of signals of the form $\mathbf{s} = \varphi_\omega + \nu$ where φ_ω is a single unknown frequency and ν is AWGN. We vary the signal SNR and the number of samples (or percentage) we use to recover the unknown frequency. Since the RADC/sparsogram is a randomized algorithm, there is some probability that it will fail; we cannot guarantee that the algorithm succeeds with probability one but we can control its success probability by increasing the average sampling rate. In Figure IV, we show how the sampling rate determines the

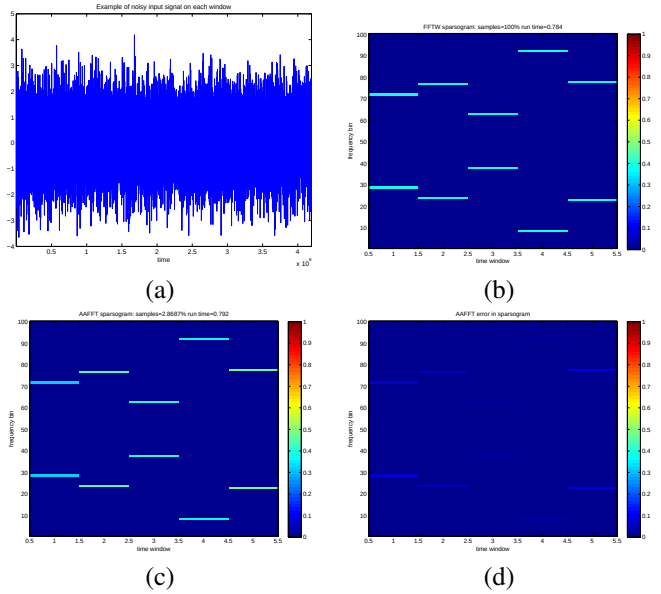

Fig. 4. Sparsogram of a frequency hopping signal in noise. (a) Test signal. (b) FFTW-based spectrogram. (c) Random sampling sparsogram. (d) Difference between (b) and (c).

probability of correctly identifying the unknown frequency. Figure IV(a) shows that the number of samples we need does not depend linearly on the length of the signal (for fixed SNR). As we increase the signal length or increase the band of the signal, the number of samples grows sublinearly with the highest frequency. Figure IV(b) shows, for a fixed signal length and SNR, how the sampling rate affects the success probability. For this bandwidth and SNR, a sampling rate of roughly 2% is sufficient to guarantee finding the uknown frequency 90% of the time. With 3% sampling rate, we recover the unknown frequency all of the time.

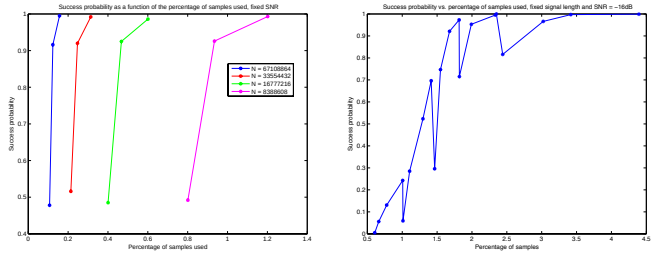

Fig. 5. (a) Success probability vs. sample rate for varying bandwidth, fixed SNR signal. (b) Success probability vs. sample rate for fixed bandwidth, fixed SNR signal.

While these sampling rates are impressive, the algorithm is also computationally efficient. Figure IV shows that the run time of the algorithm varies linearly with the number of samples and *not* with the Nyquist rate, while any FFT algorithm must grow (essentially) linearly with the Nyquist rate.

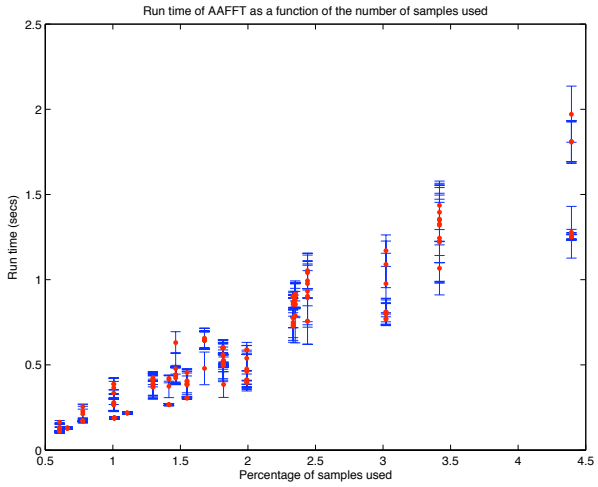

Fig. 6. Run time (y-axis) of the RADC/sparsogram algorithm grows sublinearly with the number of samples (x-axis).

V. Conclusions

In this paper we have taken some significant steps towards moving beyond Nyquist-rate analog-to-digital conversion by developing a new scheme for analog-to-information conversion of locally Fourier sparse signals. The performance of our system (sub-Nyquist sampling and linear information recovery) definitely motivates further study. Some of the many avenues for continued research, include sparsograms with smoother time windows (Hamming, Hanning, etc.), sampling schemes for signals sparse in the wavelet domain, and actual implementations of the RADCs.

Acknowledgments

All authors were supported by DARPA ONR grant N66001-06-1-2011. JL and RB were supported by NSF CCF-0431150, ONR N00014-02-10353, AFOSR FA9550-04-1-0148, and the Texas Instruments Leadership University Program. SK and YM were supported by NSF CARRER 0448558 and funds from Texas Instruments. ACG and MJS were supported by NSF DMS 03546000. MJS and MI were supported by NSF DMS 0510203. ACG is an Alfred P. Sloan Fellow.

For more information on random sampling and compressive sensing, see the website dsp.rice.edu/cs.

References

[1] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, "Improved time bounds for near-optimal sparse Fourier representation via sampling," in *Proc. SPIE Wavelets XI*, San Diego, 2005.

[2] S. R. Duncan, V. Gregers-Hansen, and J. P. McConnell, "A Stacked Analog-to-Digital Converter Providing 100 dB of Dynamic Range," in *IEEE International Radar Conference*, May 2005, pp. 31–36.

[3] N. D. Dalt, "Effect of Jitter on Asynchronous Sampling with Finite Number of Samples," *IEEE Transactions on Circuits and Systems II*, vol. 51, pp. 660–664, Dec. 2004.

[4] T. Sato, R. Sakuma, D. Miyamori, and M. Fukaset, "Performance Analysis of Wave-Pipelined LFSR," in *IEEE International Symposium on Communications and Information Technology, ISCIT*, Oct. 2004, pp. 694–699.

[5] M. Frigo and S. Johnson, "The design and implementation of FFTW3," *Proc. IEEE 93 (2)*, pp. 216–231, 2005.