Implementation Models for Analog-to-Information Conversion via Random Sampling

Tamer Ragheb, Sami Kirolos, Jason Laska, Anna Gilbert,[*] Martin Strauss,[*], Richard Baraniuk, Yehia Massoud
Department of Electrical and Computer Engineering, Rice University, Houston, TX
[*] Departments of Mathematics and EECS, University of Michigan, Ann Arbor, MI

***Abstract*— We develop a framework for analog-to-information conversion based on the theory of information recovery from random samples. The framework enables sub-Nyquist acquisition and processing of wideband signals that are sparse in a local Fourier representation. We present the random sampling theory associated with an efficient information recovery algorithm to compute the spectrogram of the signal. Additionally, we develop a hardware design for the random sampling system that demonstrates a consistent reconstruction fidelity in the presence of sampling jitter, which forms the main source of non-ideality in a practical system implementation.**

I. Introduction

Achieving higher rates of data transmission with minimum bandwidth has motivated research in fields such as signal processing, communications, networking, and electronic circuits. Emerging applications such as radar detection and ultrawideband communications stress the performance of these sampling systems beyond their physical limits as the bandwidths in these applications increase. In many cases, these wideband signals are *Locally Fourier Sparse* (LFS) in the sense that at each point in time the signals 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. The standard approach of data compression is based on sampling the signal with a rate higher than Nyquist rate, then finding the most compact representation of the signal in the digital domain. This leads to a large overhead in the amount of sampled data in the case of sparse signals that can be represented with a small number of coefficients [1], [2].

Recent theoretical advances (see Gilbert, et al. [3], for example) prove that it is possible to compress a signal directly in the sampling phase and reconstruct it by post-processing a small number of samples. 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, and the second is an efficient *information recovery algorithm* to compute the spectrogram of LFS signals, which we dub *sparsogram*. Finally, we introduce candidate system implementations for the random sampling framework to verify the success of the signal reconstruction as well as to investigate the reconstruction accuracy under the effect of sampling jitter, which represents the main source of non-ideality in the sampling system.

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). Suppose that $\mathbf{s}$ consists of a superposition of m pure tones plus noise. Gilbert et. al [3] have developed an algorithm that uses at most $m \cdot \text{poly}(1/\epsilon \cdot \log N)$ space and time and outputs a representation $\mathbf{r}$ which consists of m pure tones having MSE no more than $(1+\epsilon)$ times the energy of the noise, where ϵ is a user parameter. In the case that $\mathbf{s}$ contains no noise, the recovered signal is exact, up to precision issues. In either case, the algorithm succeeds with high probability. The algorithm selects a small sample set at random and, for each signal, with high probability, successfully finds a near-best representation. 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.

Let us describe the distribution on sample points as it is crucial to the implementation. We assume for simplicity that N is a prime number. We generate two uniformly random integers a and b from $[0, \ldots, N-1]$ with the stipulation that b is non-zero. For input parameter m, we generate the set $A = a + k \cdot b \mod N$ 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 which depends on the desired accuracy and success probability. Note that this distribution generates some sample points that are separated by distance 1. Figure 1 shows a randomly generated sample set for a signal of length 1024 which contains a single pure tone. The structure of this sample set has been optimized for the recovery algorithm which we describe below. Once we have specified the sample set, our algorithm proceeds in a greedy fashion. Given a

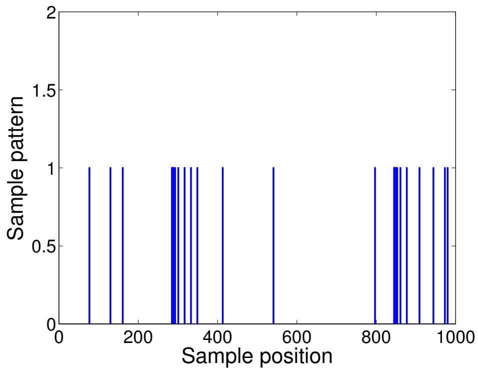


Fig. 1. A randomly generated sample pattern for a signal, consisting of a pure tone, of length 1024.

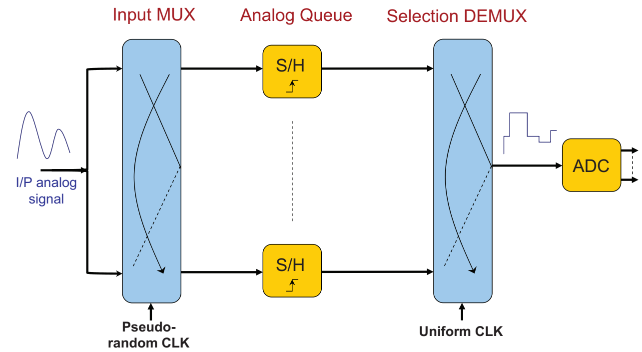


Fig. 2. RSADC block diagram. The input analog signal is fed to the input multiplexor. The MUX selects signal values with non-uniform timing and pushes them onto the analog queue which stores the values via sample and hold (S/H). Finally, these values are read with uniform timing from the selection DEMUX and sampled with a sub-Nyquist rate ADC.

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. The details may be found in [3].

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 (i.e., contains at least an η fraction of the energy) 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 recover 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). We note that this algorithm is considerably different from "Compressed Sensing" algorithms in that the algorithm runs much faster than those which solve an optimization problem [4], [5]. And, it comes with a different probabilistic failure guarantee than "Compressed Sensing" algorithms which are comparably efficient [6].

III. Random Sampling Analog-to-Digital Converters

The required samples for the random sampling recovery algorithm can be acquired with an “out of the box” analog-to-digital converter (ADC)implementation by setting the ADC clock to sample at predetermined non-uniform times. The sampling rate will be reduced compared to uniform sampling; however the recovery algorithm still requires that the minimum spacing between two samples is one over the Nyquist rate. For wideband signals this period is very small, complicating any hardware design. Thus, we propose here a novel hardware design for a *Random Sampling Analog-to-Digital Converter* (RSADC) that both achieves sub-Nyquist sampling rates and outputs an efficient representation of the signal. In the sequel we use N to also represent the Nyquist rate in Hz; the correspondence with Section II is that there N Nyquist-rate samples can be interpreted to be taken per second.

Our RSADC design is unique in that it converts non-uniformly spaced samples into uniformly spaced low-rate samples by buffering them in an analog queue. As the queue is filled, a conventional ADC pops off values at an average push rate of $(4m \log_2 N)/N$ Hz, resulting in a sub-Nyquist sampling of the signal. This design concept translates to four hardware blocks in sequence:

- **Input multiplexer:** It is driven by a *non-uniform* clock with edges predetermined by the recovery algorithm. The peak rate at which two adjacent clock values will change is the Nyquist rate. This device pushes non-uniformly selected values from the signal onto the analog queue.
- **Analog queue:** Built as an array of switched capacitors or *sample and hold* (S/H) devices, the analog queue buffers signal values before they are sampled. Due to the geometric progressions required by the recovery algorithm, the queue must be approximately $\log_2 N$ in length.
- **Selection demultiplexer:** It determines which element in the queue is to be sampled next and is driven by a sub-Nyquist-rate, uniform clock.
- **Low-rate ADC:** It is controlled by the same clock as the selection demultiplexer and it is used to digitize the sub-Nyquist random samples.

The RSADC design is shown in full in Figure 2. To understand the congestion the analog queue may endure, we refer to the example sample set presented in Figure 1. We are most concerned with closely spaced samples, and thus the queue length $\log_2 N$ guarantees we will have enough space to buffer all of the samples.

Each component in this system exhibits some non-ideal behavior. For instance, both the input and selection MUX may encounter switching delay, or the switch capacitors will store noisy values as a result of feedthrough. The most egregious error in the RSADC is timing jitter. Each component described above will contribute jitter error, however the input multiplexer jitter is most detrimental to performance, because the errors it produces will propagate through the system. Furthermore, the input multiplexor operates at the fastest rates in the system

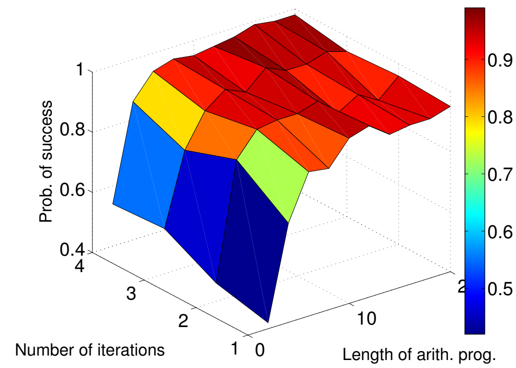


Fig. 3. With all other parameters fixed, the recovery probability is a function of the length of the arithmetic progressions and the number of greedy iterations of the algorithm.

and is thus susceptible to a more exaggerated timing error than the other components. This must be taken into consideration when budgeting the design.

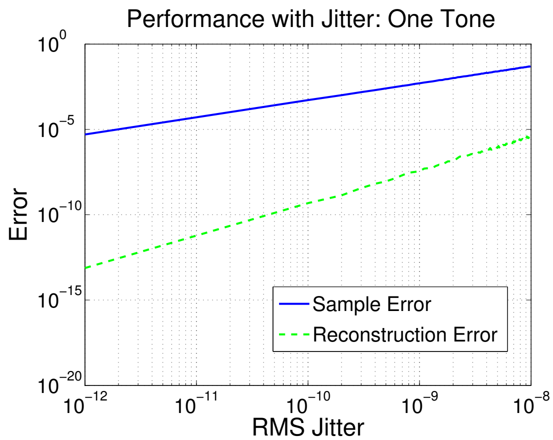
IV. Implementation Results

The recovery algorithm described in Section II has several adjustable parameters that may affect the number of samples required and the quality of reconstruction. Here we choose to focus on jitter effects and make modifications to the algorithm accordingly. A MATLAB implementation of the recovery algorithm has been developed and was used to determine which parameters for our experiments. We fixed a signal length of $N = 1024$, a relatively small number, so that we could efficiently run many experiments. We generated two types of LFS signals: a single pure tone (chosen uniformly at random) and a signal with two tones (both chosen independently and uniformly at random). The amplitudes of the tones in both experiments were chosen as -1 or 1 at random. There are several parameters that determine the structure of the sample set (e.g., the lengths of the arithmetic progressions and the number of them) as well as parameters that govern *how* the algorithm makes use of those samples. Figure 3 illustrates the different recovery probability values as a function of both the length of the arithmetic progressions and the number of greedy iterations of the algorithm. Similar empirical analysis provides us with the optimized parameters values given in Table IV. These values guarantee, with probability greater than 95%, that the reconstruction error will be zero to machine precision. All experiments for the remainder of this paper use these settings.

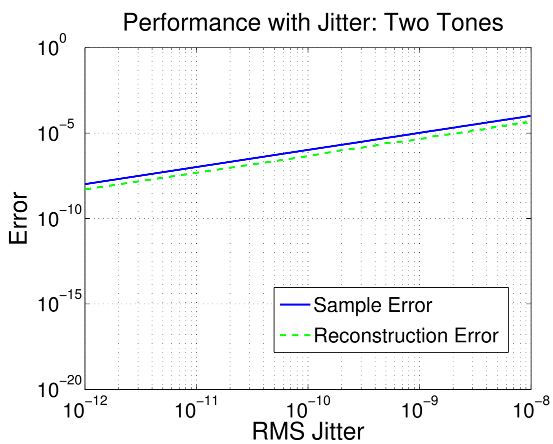
We first explore jitter effects on both the sampling and the reconstruction. We start by choosing an RMS jitter value σ. Each sample position from the set $\mathcal{S}$ is perturbed according to a Gaussian with variance σ^2; thus the resulting sample positions are $\widehat{\mathcal{S}} \sim \mathcal{N}(\mathcal{S}, \sigma^2)$. The experiments proceed by varying the timing jitter on each block from 10^{-12}s to 10^{-8}s. To understand the combined effect of each block, only the error induced by jitter on the output samples was measured. The system was tested on both a single sinusoid and a sum of two sinusoids using corresponding parameters shown in

TABLE I
ALGORITHM PARAMETERS.

Number of Tones	Arithmetic Progressions	Arithmetic Progression Length	Greedy Iterations	Total Samples
1	3	2	1	25
2	6	18	1	259



(a)



(b)

Fig. 4. Jitter effect on the system performance. Blue solid lines represent the error on the sample values due to jitter ($\|x(\mathcal{S})-x(\widehat{\mathcal{S}})\|_2$). Green dashed lines represent error of the reconstructed signal from the samples with jitter ($\|x-\widehat{x}\|_2$). (a) Performance for a single toned signal and corresponding parameters. (b) Performance for a dual toned signal and corresponding parameters.

Table IV. In order to isolate effects of jitter, we chose single-sinusoid and double-sinusoid signals on which the algorithm performs well and fixed them for all trials.[1] For each RMS jitter value, 100 trials were performed, each with a new jitter realization. Finally, the samples $x(\widehat{\mathcal{S}})$ were input to the MATLAB recovery algorithm which outputs $\widehat{x}$, and we computed $\|x(\mathcal{S})-x(\widehat{\mathcal{S}})\|_2$ to judge the "quality" of the output samples and $\|x-\widehat{x}\|_2$ for the reconstruction.

Figure 4 displays the results of our experiments in (a) the single tone case and (b) the two tone case. In both cases, the error follows the same trend as the error on the samples. In fact, the error on the reconstruction is lower than the sample error in both cases. This is because the reconstruction algorithm attempts to fit the samples to a sparse model, which if it is the correct model, is in effect denoising the signal. We also notice that the denoising is less prominent in the two tone case. Both experiments verify that the reconstruction performance will be proportional to the noise on the samples; however the tested cases actually perform better than the theoretical guarantees given in [3].

[1]Additional experiments confirm that most signal choices work comparably well and, by the nature of the algorithm, the full algorithm—which takes somewhat more samples than our bare bones algorithm here—will recover *each* signal with high probability since the bare bones algorithm recovers *most* signals. A full experimental study is the subject of future work.

V. Conclusions

The theory of information recovery from random samples provides a new avenue for advanced analog-to-digital conversion of wideband Locally Frequency Sparse (LFS) signals at sub-Nyquist rates. Our hardware design achieves a consistent reconstruction fidelity in the presence of sampling jitter, which forms the main source of non-ideality in a practical system implementation. As an added bonus, the back-end signal recovery algorithm runs in sub-linear time (fewer computations than the number of Nyquist samples). Current and future work will extend not only the theory of random sampling and the practice of random-sample analog-digital converters but will also test and validate the approach on a broad range of LFS signals.

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 was 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. Gerosa, A. Xotta, A. Bevilacqua, and A. Neviani, "An a/d converter for multimode wireless receivers, based on the cascade of a double-sampling $\sigma\delta$ modulator and a flash converter," *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 10, pp. 2109 – 2124, October 2006.

[2] A. Olyaei and R. Genov, "Focal-plane spatially oversampling cmos image compression sensor," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 1, pp. 26 – 34, January 2007.

[3] 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.

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Info. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[6] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: Fast algorithms for Compressed Sensing," in *Proc. 39th ACM Symposium on Theory of Computing*, San Diego, Jun. 2007, to appear.