Compressive Sampling for Streaming Signals with Sparse Frequency Content

Petros Boufounos
Mitsubishi Electric Research Laboratories
petrosb@merl.com

M. Salman Asif
School of ECE, Georgia Institute of Technology
sasif@gatech.edu

***Abstract*—Compressive sampling (CS) has emerged as significant signal processing framework to acquire and reconstruct sparse signals at rates significantly below the Nyquist rate. However, most of the CS development to-date has focused on finite-length signals and representations. In this paper we discuss a streaming CS framework and greedy reconstruction algorithm, the Streaming Greedy Pursuit (SGP), to reconstruct signals with sparse frequency content.**

Our proposed sampling framework and the SGP are explicitly intended for streaming applications and signals of unknown length. The measurement framework we propose is designed to be causal and implementable using existing hardware architectures. Furthermore, our reconstruction algorithm provides specific computational guarantees, which makes it appropriate for real-time system implementations. Our experimental results on very long signals demonstrate the good performance of the SGP and validate our approach.

I. Introduction

Compressive Sensing (CS) [1–3] has emerged as a powerful signal processing framework that enables signal acquisition at rates significantly below the Nyquist rate by exploiting the sparse structure of natural and man-made signals. The success of CS has driven both algorithm and hardware development to implement the theory, which places significant emphasis on randomized incoherent measurements at the acquisition stage and increased computation at the reconstruction stage.

Unfortunately, most of the CS results to date focus on finite-dimensional signals. Even when used to acquire streaming signals, such as audio, video and radio, the unstated assumption is that the signal is processed in finite-length blocks. Each block is compressively sampled and reconstructed using one of the known finite-dimensional algorithms. Such an approach can introduce significant blocking artifacts and input-output delay. Furthermore, existing finite-dimensional algorithms cannot provide hard guarantees on the computation time, often a critical requirement of streaming real-time systems. Thus, significant buffering of the input or excessive allocation of computation time for reconstruction might be required to ensure that the system satisfies timing requirements.

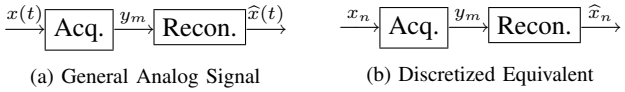
In this work we contribute a new CS framework with features explicitly designed for streaming signals. Specifically, we present a causal algorithm—inspired by the Compressive Sampling Matching Pursuit (CoSaMP) [4]—to process the streaming measurements. Our algorithm, which we term Streaming Greedy Pursuit (SGP), operates at a fixed input rate, computes the reconstruction with a fixed cost per input measurement and outputs the estimated streaming signal at a fixed output rate. The streaming framework we present explicitly avoids processing the signal in disjoint blocks and therefore avoids blocking artifacts. Furthermore, our framework provides strict computational guarantees and explicit tradeoffs between input-output delay, reconstruction performance and computation. These features make our framework very well-suited for real-time applications. In fact, we have already successfully used it in on-line reconstruction of high-speed periodic videos using a low-speed coded exposure camera [5].

Our goal in this paper is to introduce the measurement framework and provide a streaming reconstruction algorithm for signals that are sparse or compressible in the frequency domain. Still, the fundamental concepts can be immediately extended to other suitable sparse domains as well.

To our knowledge, the only other attempt to date to incorporate infinite-dimensional signals in CS is summarized in [6]. The authors assume the signal has a multi-band structure in the frequency domain and formulate an off-line finite-dimensional support estimation problem, the solution of which controls a linear reconstruction system. The authors further demonstrate exact and very efficient recovery under the multi-band signal assumption, assuming the estimated support does not change for the duration of the signal. Otherwise, whenever the multi-band signal support changes it needs to be re-estimated using a standard sparse estimation algorithm and the linear reconstruction system needs to be updated.

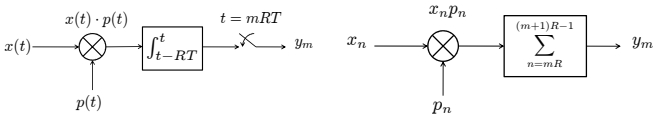
Our approach, instead, combines on-line support and signal estimation, similarly to standard CS algorithms and practice. Thus, we provide explicit causality and computation guarantees and can also incorporate CS extensions such as model-based CS or measurement quantization [7–9]. Furthermore, as we demonstrate in the experimental section, using continuous on-line support estimation our system can track signals with frequency drifts, such as linear frequency chirps or signals with local variations in their frequency support, which are globally wideband but locally narrowband.

The paper is organized as follows. Section II provides some background on signal acquisition, CS, and CS hardware models. Section III establishes our streaming signal and acquisition model. Section IV describes the Streaming Greedy Pursuit (SGP) and Section V presents simulation results on its performance.



(a) General Analog Signal Acquisition System (b) Discretized Equivalent

Fig. 1. Basic Streaming Acquisition System



(a) Random demodulator (b) Discrete-time equivalent

Fig. 2. System model of the random demodulator

II. Background

A. Signal Acquisition

To introduce CS to streaming applications we consider the general streaming system depicted in Fig. 1(a). The signal $x(t)$ is acquired using an acquisition system ("Acq." in the figure) at an average rate of M measurements per time unit and subsequently reconstructed ("Recon." in the figure) using those measurements. In most classical signal acquisition systems the acquisition component is an analog-to-digital converter (ADC) which obtains linear measurements using a low-pass anti-aliasing filter followed by uniform time sampling and quantization. The reconstruction component is the linear bandlimited interpolation of the measurements y_m.

Assuming a suitable discrete representation x_n of the signal $x(t)$ using N coefficients per time period, the system can be discretized as shown in Figure 1(b). For example, in classical bandlimited sampling and interpolation $x_n = x(nT)$, where T is the signal Nyquist period. In this case the acquisition and reconstruction components are the identity (i.e., $m = n$ and $\widehat{x}_n = y_n = x_n$) and the implied reconstructed signal $\widehat{x}(t)$ is the bandlimited interpolation of $\widehat{x}_n$. More general cases with less trivial acquisition and reconstruction components are discussed in [10]. In this paper we assume that such a discretization exists and describes the acquisition system to sufficient accuracy. We examine this discretization for a fairly general hardware model in Sec. II-C.

Using the discretized model of the acquisition system, the Nyquist rate becomes a requirement that the rate M of the acquired measurements y_m is greater than or equal to the input rate N. Otherwise, the system is not invertible for all input signals and information is lost. However, with additional information on the signal structure it is possible to acquire a signal at a lower measurement rate $M \ll N$ and still reconstruct it.

B. Compressive Sensing

Compressive Sensing [1–3] demonstrates that a signal sparse or compressible in some basis can be efficiently sampled and reconstructed using very few linear measurements. The signal of interest, $\mathbf{x} \in \mathbb{R}^N$, is measured using the system

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{y}$ denotes the measurement vector and $\mathbf{A}$ an $M \times N$ measurement matrix with $M \ll N$. The signal $\mathbf{x}$ is assumed K-sparse in some basis $\mathbf{B}$, i.e., $\mathbf{B}^{-1}\mathbf{x}$ contains only K non-zero coefficients.

The measurement matrix $\mathbf{A}$ satisfies the *Restricted Isometry Property* (RIP) of order $2K$ if there exists a constant $\delta_{2K} < 1$ such that

$$(1 - \delta_{2K})\|\mathbf{x}\|_2^2 \le \|\mathbf{A}\mathbf{x}\|_2^2 \le (1 + \delta_{2K})\|\mathbf{x}\|_2^2, \tag{2}$$

for all $\mathbf{x}$ that are $2K$-sparse in the basis $\mathbf{B}$. If the RIP constant δ_{2K} is sufficiently small, then the signal can be exactly reconstructed using the convex optimization [3]

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{B}^{-1}\mathbf{x}\|_1 \ \text{ s.t. } \ \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{3}$$

or a greedy algorithm such as CoSaMP [4]. Furthermore, a small RIP constant provides robustness guarantees for recovery in the presence of measurement noise and for sampling signals that are not exactly sparse but can be well-approximated by a sparse signal [3, 4].

C. CS-based Analog-to-Digital Conversion

Significant work has been performed on hardware implementations of CS on streaming signals [11–13]. Such efforts focus on the hardware architectures that enable random projections. However, for most of these systems, the unstated assumption is that the signal is processed in disjoint finite-length blocks. Each block is compressively sampled and reconstructed using one of the known finite-dimensional algorithms.

In this section we describe one of the most successful architectures for compressive sensing—the random demodulator architecture and its variants [11]—with focus on how the analog continuous-time components of the system can be described using a discrete-time model. The demodulator has several advantages over alternative architectures. Specifically, it lends itself to a simple and robust hardware implementation, provides a regular sample output rate, has a straightforward discrete-time equivalent system for analysis purposes. Several variants of this architecture include parallel implementations of banks of modulators, random strobing and coded aperture implementations for imaging systems [13]. Our aim is to demonstrate that a discrete-time equivalent is a sufficient representation for the acquisition process.

The random demodulator, demonstrated in Fig. 2(a), first multiplies the continuous-time signal $x(t)$ with a random mixing pulse $p(t)$, generated as

$$p(t) = \sum_{n \in \mathbb{Z}} p_n p_o(t - nT), \tag{4}$$

where p_n is typically a random ± 1 sign sequence, T should be less than the Nyquist period corresponding to the bandwidth of $x(t)$ and $p_o(t)$ is typically a square pulse of width T. The product $x(t) \cdot p(t)$ is subsequently integrated over a period of RT and sampled at that rate to produce y_m at rate R times lower that the Nyquist rate of $x(t)$.

If $p_o(t)$ has support only in $0 \le t \le T$ the random demodulator is equivalent to the discrete-time system in Fig. 2(b)

where x_n is a discrete representation of $x(t)$. Specifically, x_n is samples of the convolution of $p_o(t)$ with $x(t)$ at rate T:

$$x_p(t) = x(t) * p_o(t) \Leftrightarrow X_p(f) = X(f) \cdot P_o(f) \quad (5)$$

$$x_n = x_p(nT), \quad (6)$$

where $X_p(f)$, $X(f)$, and $P_o(f)$ denote the continuous-time Fourier transforms of the respective signals. Under this discretization, the integrator is replaced by a summation of R consecutive values of the product $x_n p_n$:

$$y_m = \sum_{n=mR}^{(m+1)R-1} x_n p_n. \quad (7)$$

If $P_o(f)$ is non-zero in the support of $X(f)$, the discretization is invertible, and therefore recovering the discrete representation x_n from y_m is sufficient to reconstruct $x(t)$. The discretization preserves the sparsity structure of the signal, i.e., the support of the non-zero frequency components.

A simple extension of this hardware is the interleaving of L such demodulators, each using the same pulse width T but different random sequences p_n and each integrating and sampling with period LRT, i.e., summing over LR discrete-time coefficients. By staggering the sampling times of each demodulator, a constant rate of one measurement per R input coefficients is maintained but each measurement measures a larger signal length, thus providing more noise robustness.

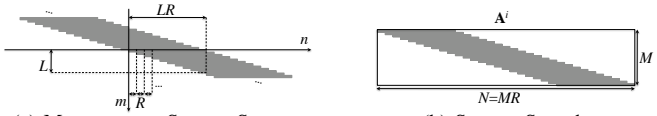
III. Streaming compressive sampling

In contrast to standard finite-length CS signal models, streaming signals do not have a pre-determined length. While it is possible to acquire the whole stream and post-process it, the storage and computational requirements for such an endeavor are often prohibitive. Furthermore, several applications often require on-line processing of such signals. Standard CS approaches perform on-line computation by processing the stream in finite-length blocks which are considered independent of each other. However, this approach ignores the continuity of the signal and introduces significant blocking artifacts. Instead, the algorithm we propose assumes a streaming signal and measurement model, which we describe in this section.

A. Signal model

Although frequency-domain sparsity is straightforward to define in finite-length signals, its infinite-dimensional extension can be elusive. The model we consider in this paper resorts to the well-defined concepts of sparsity and compressibility of finite-dimensional signals to describe snapshots of the infinite-dimensional signal. Thus we are able to exploit a wide literature of existing CS results and provide adaptivity to local changes in the signal sparsity structure even when the signal is not globally sparse.

In particular, we consider finite-length snapshots of x_n using a window v_n of length $N = MR$, where M is an integer and v_n is strictly positive on its support and zero outside it. Our model assumes that any such snapshot has a sparse or compressible discrete Fourier transform (DFT). In other words, if $\mathbf{x} \in \mathbb{R}^N$ denotes a length-N segment of the signal we assume that $\mathbf{FVx}$ is well approximated with few significant coefficients, where $\mathbf{F}$ denotes the DFT and $\mathbf{V}$ is a diagonal operator with the window coefficients on its diagonal.



(a) Measurement System Support (b) System Snapshot

Fig. 3. Measurement system parameters.

The model is inspired by the success of the short-time Fourier transform and spectrogram methods in representing and analysing signals that exhibit some local structure but are globally wideband. Signals often encountered in practice, such as speech, chirps, FM and frequency hopping radio exhibit such behavior. Furthermore, the model encompasses stationary signals that exhibit a global sparsity structure.

B. Measurement Model

We consider a general measurement model with sufficient richness to describe several acquisition architectures suitable for streaming signals. Specifically, we measure the streaming signal x_n using a time varying linear system $a_{m,n}$ as

$$y_m = \sum_n x_n a_{m,n} = \langle \mathbf{x}, \mathbf{a}_m \rangle, \quad (8)$$

where y_m is the sequence of measurements, and $\mathbf{a}_m$ is sequence of infinite-dimensional measurement vectors. The system has an input rate of N coefficients per unit time for x_n and an output (measurement) rate of $M = N/R$ measurements per unit time, where R denotes the downsampling rate. For notational simplicity, in the remainder of this paper we assume that R is an integer[1].

We assume that the measurement system is causal with finite response length L. Thus $a_{m,n}$ has nonzero support in

$$n = mR, \ldots, (m + L)R - 1, \quad (9)$$

Figure 3(a) summarizes the relevant parameters of the measurement system $a_{m,n}$. The shaded area demonstrates the support of the system relative to the axes. The system coefficients are assumed zero in the unshaded area and can take any value in the shaded area, depending on the hardware modeled. With appropriate choice of the system parameters this model can accommodate several hardware architectures, such as the random demodulator and the bank of L interleaved random demodulators described in Sec. II-C, parallel non-interleaved random demodulators, the random sampler, periodic non-uniform sampling, the coded strobing and coded aperture camera, and several others.

[1]In several instances in this paper we assume rates and lengths are integer multiples of each other. Non-integer rates are straightforward to accommodate with appropriate use of rounding operators. However, the notation becomes cumbersome without adding any insight to the fundamental concepts.

C. System Snapshots

The Streaming Greedy Pursuit operates on snapshots of the system, updated at the measurement rate. The i^{th} snapshot of the signal, the measurements and the measurement system are denoted respectively using

$$\mathbf{x}^i = [x_{iR}, \ldots, x_{iR+N-1}]^T, \tag{10}$$

$$\mathbf{y}^i = [y_i, \ldots, y_{i+M-1}]^T, \text{ and} \tag{11}$$

$$\mathbf{A}^i = \begin{bmatrix} a_{i,iR} & \cdots & a_{i,iR+N-1} \\ \vdots & \ddots & \vdots \\ a_{i+M-1,iR} & \cdots & a_{i+M-1,iR+N-1} \end{bmatrix}, \tag{12}$$

where $N = RM$ is an appropriate length for the signal to be sparse under the model in Sec. III-A. We use $\mathbf{s} = \mathbf{F}\mathbf{x}$ to denote the corresponding frequency-domain representation of the signal snapshot, in which $\mathbf{F}$ is the $N \times N$ DFT matrix.

We further assume that $M \times N$ snapshots of the measurement matrix, pictorially shown in Fig. 3(b) satisfy the RIP for frequency sparse signals, i.e. $\mathbf{A}^i$ satisfies (2) with small δ_{2k} with $\mathbf{F}$ as the sparse basis. Although the RIP is not a necessary condition for recovery, it is a sufficient one and provides several robustness and reconstruction guarantees. It has already been shown that a length N snapshot of the random demodulator satisfies the RIP with very high probability as long as $M = O(K \log(N/K))$ [12]. It is also straightforward to show that if one length-N snapshot of the demodulator satisfies the RIP for frequency sparse signals, then a periodic replication of that length-N sequence p_n will also satisfy the RIP for every snapshot of the system as defined in (12).

IV. Streaming greedy pursuit

In contrast to the standard CoSaMP reconstruction algorithm, the Streaming Greedy Pursuit (SGP) is an on-line algorithm that receives one measurement per iteration and outputs the estimate of R consecutive signal values. In each iteration the algorithm operates once on the current snapshot of the system, which is updated after the iteration.

To simplify the presentation of the SGP, we first describe the streaming iteration loop in Sec. IV-A, followed by more detailed discussion of the signal estimation and refinement algorithm in Sec. IV-B.

A. Streaming iteration

The SGP operates by estimating the signal on a sliding window of length N over the streaming signal using M measurements. After each iteration, a new measurement is incorporated at the end of the measurement window and the oldest measurement is removed from the beginning. Similarly, R new signal samples (to be estimated) are included in the signal window and the oldest R are removed from the window and committed to the output.

In every i^{th} iteration, SGP maintains a working signal estimate of length N, denoted $\widehat{\mathbf{x}}^i$, a working measurement matrix of dimension $M \times N$, denoted $\widehat{\mathbf{A}}^i$, and a measurement vector of length M, denoted $\mathbf{y}^i$. The iteration is presented in Algorithm 1, where we use the subscript $(\cdot)_m$ notation to denote the m^{th} element of a vector and $(\cdot)_{\{m_1,\ldots,m_2\}}$ to denote a range of elements. Non-boldface letters denote the streaming signals. At each iteration, the working signal estimate from the previous iteration $\widehat{\mathbf{x}}^{i-1}$ is refined using a sequence of steps (step 2, with details shown in Algorithm 2) almost identical to a single CoSaMP iteration.

Algorithm 1 Streaming iteration for SGP

1: *Increase iteration count:* $i \leftarrow i + 1$

2: *Refine working estimate:*

$$\widetilde{\mathbf{x}}^i \leftarrow \text{Refine}(\widehat{\mathbf{x}}^{i-1}, \mathbf{y}^{i-1}, \mathbf{A}^{i-1}),$$

where Refine$(\cdot)$ is described in Algorithm 2.

3: *Update the weighted average of all the samples in the window and commit the estimate for the oldest samples:*

$$\overline{\mathbf{x}}^i \leftarrow \begin{bmatrix} \overline{\mathbf{x}}^{i-1}_{\{R+1,\ldots,N\}} \\ \mathbf{0}_R \end{bmatrix} + \mathbf{W}\widetilde{\mathbf{x}}^i,$$

$\mathbf{0}_R$ is a vector with R zeros, $\mathbf{W}$ is the diagonal weight matrix.

$$\widehat{x}_{Ri+j} = \overline{\mathbf{x}}^i_j, \quad j = 1, \ldots, R$$

where $\widehat{x}_n$ is the streaming signal estimate at the output.

4: *Slide working estimate window (circular shift in time):*

$$\widehat{\mathbf{x}}^i \leftarrow \begin{bmatrix} \widetilde{\mathbf{x}}^i_{\{R+1,\ldots,N\}} \\ \widetilde{\mathbf{x}}^i_{\{1,\ldots,R\}} \end{bmatrix}$$

5: *Slide working measurement window:*

$$\mathbf{y}^i \leftarrow \begin{bmatrix} \mathbf{y}^i_{\{2,\ldots,M\}} \\ y_{i+M} \end{bmatrix}$$

6: *Slide working measurement matrix:*

$$\mathbf{A}^i \leftarrow \left[\begin{array}{c|c} \mathbf{A}^{i-1}_{\{2,\ldots,M\},\{R+1,\ldots,N\}} & \mathbf{0}_{[M-1\times R]} \\ \hline a_{(i+M-1),\{iR,\ldots,iR+N-1\}} & \end{array}\right],$$

where $\mathbf{0}_{[m\times n]}$ denotes $m \times n$ zero matrix.

Any sample of x_n is included in M working snapshots of the system, i.e., M SGP iterations, and the SGP computes its estimate at every iteration. The final estimate of each sample can significantly improved using this estimate history: the final value for the estimate of any outgoing signal sample is computed as the weighted average over the M intermediate estimates. The weighted average is maintained in the vector $\overline{\mathbf{x}}^i$, where $\mathbf{W}$ is a diagonal weight matrix of choice (step 3). The oldest R samples in the current snapshot are finalized, committed to the output and removed from the vector.

At the end of each streaming loop iteration the system snapshot is updated by incorporating one new measurement (step 5), sliding the measurement system snapshot (step 6) and circularly shifting the working signal estimate (step 4). While the shifts in steps 3, 5, and 6 are linear, the shift in step 4 is circular because it maintains the signal support, as estimated in the frequency domain.

B. Signal Estimation and Refinement

The signal estimation and refinement step 2 of the recovery algorithm, summarized in Algorithm 2, is inspired by CoSaMP

[4]. Both CoSaMP and SGP refine the signal estimate using the unexplained residual to identify a candidate support for the signal estimate (steps 1–3 in Alg. 2), inverting the measurements in that support to obtain a new signal estimate (step 4), and truncating the signal estimate to have the required sparsity (step 5). The main differences with the standard CoSaMP iteration is (a) incorporating a virtual window in the signal estimation to improve the signal smoothness, even if the sampling system cannot impose a real window on the signal, and (b) reducing the number of support coefficients added to the candidate support in step 3, in order to control the computation cost for on-line performance.

To introduce the virtual smoothing window we consider the signal $\mathbf{x}$ actually measured versus the windowed signal $\mathbf{Vx}$ we would desire to measure, where $\mathbf{V}$ is a diagonal matrix. By rewriting the measurement system snapshot as

$$\mathbf{y} = \mathbf{Ax} = \mathbf{AV}^{-1}\mathbf{Vx} \equiv \mathbf{AV}^{-1}\mathbf{F}^{-1}\mathbf{s}, \tag{13}$$

we can introduce the virtual window by substituting $\mathbf{A}$ with $\mathbf{AV}^{-1}$, as long as the diagonal of $\mathbf{V}$ has no zero values.

The Refinement algorithm comprises the following steps:

1) **Residual and proxy computation.** Steps 1 and 2 in Alg. 2 compute the unexplained residual in the current working snapshot and form a proxy to estimate the DFT coefficients that best explain the residual.

2) **Support identification and merger:** Step 3 identifies the T largest coefficients (in magnitude) in the proxy merges their support set with the size-K support set of the DFT coefficients of the current signal estimate.

3) **Signal estimation:** Step 4 updates the DFT coefficients estimate by solving a least squares problem over the merged support. Note that the candidate signal support differs by at most $2T$ coefficients from the support in the previous iteration. Thus, a direct computation requires only $2T$ rank-one updates.

4) **Truncation:** Step 5 truncates the estimate of DFT coefficients and maintains its K largest coefficients. The signal is estimated using the inverse DFT of the truncated estimate $\widetilde{\mathbf{s}}$.

Note that, similar to CoSaMP, the SGP can incorporate further modifications, such as model-based CS [7] and saturation-aware CS [9] in the refinement algorithm.

Algorithm 2 Signal Refinement function: Refine($\widehat{\mathbf{x}},\ \mathbf{y},\ \mathbf{A}$)

Note: *Update for sparse* $\mathbf{s} = DFT(\mathbf{Vx})$*:*

$$\widehat{\mathbf{y}} = \widehat{\mathbf{A}}\mathbf{x} = \widehat{\mathbf{A}}\mathbf{V}^{-1}\mathbf{Vx} \equiv \widehat{\mathbf{A}}\mathbf{V}^{-1}\mathbf{F}^{-1}\mathbf{s}.$$

1: *Calculate residual:*

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\widehat{\mathbf{x}},$$

2: *Compute proxy:*

$$\mathbf{p} = \left(\mathbf{AV}^{-1}\mathbf{F}^{-1}\right)^{*}\mathbf{r},$$

where $\mathbf{F}$ denotes the DFT matrix, $\mathbf{V}$ is the virtual window and $(\cdot)^*$ denotes the conjugate transpose.

3: *Identify and merge support:*

$$\Omega = \operatorname{supp}\left(\mathbf{FV}\widehat{\mathbf{x}}\right) \cup \operatorname{supp}\left(\mathbf{p}|_T\right),$$

where $\operatorname{supp}(\cdot)$ denotes the support index set of a vector, and $\mathbf{p}|_T$ denotes truncation of the vector $\mathbf{p}$ to its T largest in magnitude coefficients.

4: *Estimate DFT coefficients over the merged support:*

$$\mathbf{b} = \left(\widehat{\mathbf{A}}\mathbf{V}^{-1}\mathbf{F}^{-1}\Big|_{\Omega}\right)^{\dagger}\widehat{\mathbf{y}}$$

5: *Truncate DFT coefficients and compute the new estimate:*

$$\widetilde{\mathbf{s}} \leftarrow \mathbf{b}|_K$$

6: *Output:*

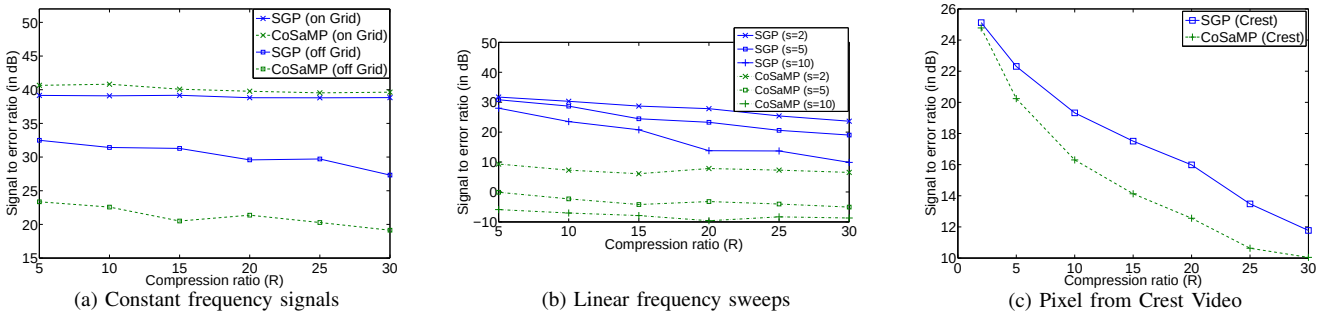
$$\widetilde{\mathbf{x}} \leftarrow \mathbf{V}^{-1}\mathbf{F}^{-1}\widetilde{\mathbf{s}}.$$

V. Numerical experiments

In this section we present some experimental results which demonstrate the performance of the SGP algorithm for various frequency-sparse signals. We also compare its performance with the standard CoSaMP algorithm operating on the entire signal at once.

On-Grid frequencies: This experiment examines the performance on signals consisting of sinusoids with frequencies exactly selected from the frequency grid. We compose the signal by selecting 10 uniformly random frequencies from the grid locations defined by an N-point DFT, where N is the length of the sliding window, and draw their amplitude from a normal distribution. The signals are acquired using a simple random demodulator using compression ratio R, with measurements corrupted by random normally distributed noise at 35dB SNR. We execute the SGP simulations for $M = 100$, $N = 100R$, $K = T = 12$ and different values of R over 1500 measurements, i.e., $1500R$ signal samples. We also compare CoSaMP to estimate the sparse DFT coefficients of the entire signal using the complete measurement set. Figure 4(a) plots the signal-to-reconstruction error ratio for different values of R, averaged over 50 simulations. Since the signal is perfectly sparse in the frequency domain, both SGP and CoSaMP estimate the signal with good accuracy in the presence of noise. As expected, in this simulation CoSaMP, being an off-line algorithm, performs better because it can see the whole measurement set.

Off-grid frequencies: In this experiment the streaming signal consists of 4 sinusoids with frequencies that do not lie on the N-point DFT grid. As with the previous simulation, the frequencies of the 4 sinusoids are randomly selected form a uniform distribution and their amplitude from a standard normal distribution. We execute the SGP simulations for $M = 120$, $N = 120R$, $K = T = 15$ and different values of R over 1500 measurements ($1500R$ samples). We use a Kaiser window (with $\beta = 5$) as the virtual window in Algorithm 2. Similarly, CoSaMP is used to estimate the DFT coefficients of the windowed sequence (using similar Kaiser window on the whole signal). The signal-to-reconstruction error ratio for different values of R, averaged over 50 simulations, is also presented in Fig. 4(a). Note that windowing makes the DFT coefficients of signal sparser but the samples near the end-points cannot be reconstructed reliably. This is one reason for the degradation in the reconstruction performance of CoSaMP. On the other hand, the SGP averages the reconstructed samples over multiple sliding windows, which improves performance.



(a) Constant frequency signals (b) Linear frequency sweeps (c) Pixel from Crest Video

Fig. 4. Comparison between the performance of the SGP and CoSaMP.

Linear frequency sweeps: In this experiment, the frequencies of the streaming signal vary with time. The streaming signal consists of 4 sinusoids whose frequencies drift by $\pm s$ cycles over $1500R$ samples, which is the duration of our test signal. An example of such signal with $s = 5$ is shown on the top of Fig. 5. Since the frequency of the signal is varying slowly, the DFT of the signal over a small portion is sparse (as shown in STFT) but the DFT of the entire signal is fairly spread out. We tested SGP and CoSaMP for this signal with three different values of $s \in [2,\ 5,\ 10]$ and compression ratio R. The performance results are presented in Fig. 4(b). Since the SGP works on small sliding windows, every snapshot of the time-varying signal has sparse DFT, and the overlapping sliding window provides it with the ability to track the changing frequencies. On the other hand, since CoSaMP operates working on the entire signal, which is fairly wideband, its sparse approximation in the frequency domain is not as accurate.

Crest nearly periodic: The streaming signal in this experiment come from the time-series of pixels from a high-speed video of an oscillating Crest toothbrush, an example is shown on the bottom of Fig. 5. The period is roughly 16 samples and fluctuates over time due to variations in the movement of the brush. We used $M = 190$, $K = 20, T = 10$ and executed the SGP on 10000 samples in the presence of 35 db normally distributed noise. Similarly, CoSaMP was used to estimate 10000 samples at once using $10000/R$ measurements, for different values of R. The results are shown in Fig. 4(c). SGP outperforms CoSaMP in this case as well because of its ability to adapt to local frequency variations. Further results on high-speed video signals are available in [5].

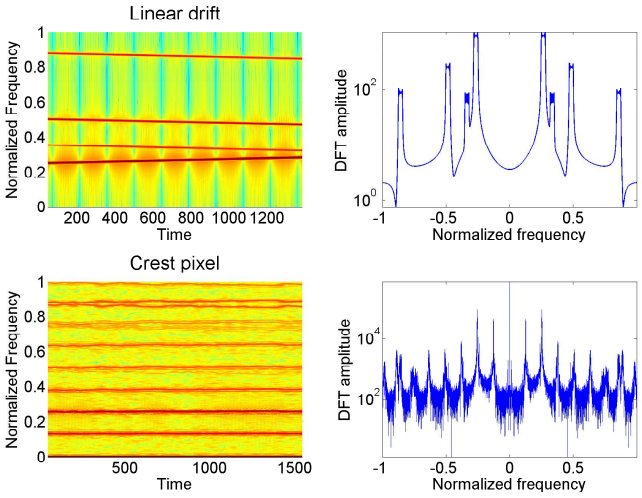


Fig. 5. Frequency content of signals with locally varying frequency content

REFERENCES

[1] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[3] E. Candès, "Compressive sampling," *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, vol. 3, pp. 1433–1452, 2006.

[4] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples.," *Appl. Comp. Harmonic Anal*, June 2008.

[5] M. S. Asif, D. Reddy, P. T. Boufounos, and A. Veeraraghavan, "Streaming compressive sensing for high-speed periodic videos." (submitted).

[6] M. Mishali and Y. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *Signal Processing, IEEE Transactions on*, vol. 57, pp. 993–1009, March 2009.

[7] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *preprint*, 2008.

[8] P. Boufounos, "Greedy sparse signal reconstruction from sign measurements," in *Asilomar Conference On Signals Systems and Computers*, 2009.

[9] J. N. Laska, P. Boufounos, M. A. Davenport, and R. G. Baraniuk, "Democracy in action: Quantization, saturation, and compressive sensing," *Preprint*, 2009.

[10] M. Unser and A. Aldroubi, "A general sampling theory for nonideal acquisition devices," *Signal Processing, IEEE Transactions on*, vol. 42, pp. 2915–2925, Nov 1994.

[11] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *Circuits and Systems, IEEE International Symposium on*, pp. 1959–1962, May 2007.

[12] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: Efficient sampling of sparse, bandlimited signals," *IEEE Trans. on IT*, vol. 56, pp. 520–544, Jan. 2010.

[13] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing photography for high-speed periodic events," *under review IEEE Trans. Pattern Anal. Mach. Intell.*