

A Low-Power Compressive Sampling Time-Based Analog-to-Digital Converter

Praveen K. Yenduri, Aaron Z. Rocca, Aswin S. Rao, Shahrzad Naraghi, Michael P. Flynn, and Anna C. Gilbert

Abstract—This paper presents a low-power, time-based, compressive sampling architecture for analog-to-digital conversion. A random pulse-position-modulation (PPM) analog-to-digital conversion (ADC) architecture is proposed. A prototype 9-bit random PPM ADC incorporating a pseudo-random sampling scheme is implemented as proof of concept. This approach leverages the energy efficiency of time-based processing. The use of sampling techniques that exploit signal compressibility leads to further improvements in efficiency. The random PPM (pulse-position-modulation) ADC employs compressive sampling techniques to efficiently sample at sub-Nyquist rates. The sub-sampled signal is recovered using a reconstruction algorithm, which is tailored for practical hardware implementation. We develop a theoretical analysis of the hardware architecture and the reconstruction algorithm. Measurements of a prototype random PPM ADC and simulation, demonstrate this theory. The prototype successfully demonstrates a 90% reduction in sampling rate compared to the Nyquist rate for input signals that are 3% sparse in frequency domain.

Index Terms—Analog-to-digital conversion (ADC), compressive sampling (CS), low-power ADC, time-based ADC, time-to-digital converters.

I. INTRODUCTION

APPLICATIONS of low-power analog-to-digital conversion (ADC) include power constrained wireless environmental sensing, high energy physics, and biomedical applications such as massive-parallel access of neuron activity ([1]–[3]). We present a new low-power, compressive-sampling analog-to-digital converter which we call a random pulse-position-modulation (PPM) ADC. This random PPM ADC is one of the first ADCs that takes advantage of the combination of time-based analog-to-digital conversion techniques and compressive sampling. In addition, we discuss a new reconstruction algorithm called PRSreco (periodic random sampling reconstruction) and present theoretical upper-bounds for input

signal reconstruction error. This algorithm is tailored to make it viable for practical hardware implementation.

Technology scaling generally improves power consumption and speed, however, it poses a number of challenges in the design of ADCs. Scaling reduces the supply voltage, which in turn reduces the signal dynamic range. This has the direct effect of reducing the signal-to-noise ratio (SNR). One way to overcome the challenges of low-voltage design is to process signals in time domain. Technology scaling favors time domain processing since it reduces gate delays and thus improves time resolution. A wide variety of time-based ADCs that quantize time or frequency instead of voltage or current, have been proposed. These designs include simple architectures such as single-slope analog-to-digital conversion [4], pulse width modulation (PWM) ADC [5], asynchronous level crossing designs [6], voltage-controlled oscillator (VCO)-based $\Sigma\Delta$ modulators [7], and integrate and fire circuits [8], [9]. Continuous time DSPs are proposed in [10]. A continuous time level crossing ADC, such as [11] and [12] can be attractive for slow moving signals. However, the key advantages of these devices are lost if continuous time DSP is not available. Furthermore, sparse signals can be dominated by high-frequency content.

This work expands on the pulse position modulation ADC architecture developed in [13]. The PPM ADC is itself an elaboration of the PWM architecture in which a continuous-time comparator compares the input to a periodic ramp, to convert the input signal information to a time-domain representation (see Section III). A two-step time-to-digital converter (TDC) then converts the time domain information to digital domain. With the use of a two-step TDC, the PPM ADC achieves both high resolution and high dynamic range, along with low-power consumption. Another way to obtain an improvement in the power efficiency of an ADC is to reduce the sampling rate [14] since to a first order, power consumption is proportional to sampling frequency. We can achieve this by employing random sampling techniques that exploit the redundancy (i.e., compressibility or sparsity) of the input signal to reduce the sampling rates to below the Nyquist rate. We implement random sampling by introducing randomness into the reference ramp signal used by the PPM ADC. The proposed random PPM ADC lies at the intersection of time-based analog-to-digital conversion and compressive-sampling to improve energy efficiency in both ways.

Many compressive sampling (CS) ADC architectures and acquisition systems have been proposed in recent years. Some designs lack efficient hardware implementation of encoding or decoding (reconstruction) algorithms [15], [16]. Other designs focus on efficient compression but do not optimize power consumption [17], [18]. Some compressive sensing designs, such

Manuscript received March 07, 2012; revised June 30, 2012; accepted September 11, 2012. Date of publication October 26, 2012; date of current version December 05, 2012. This work is supported by the National Science Foundation (NSF) under Award 0910765. A. C. Gilbert is an Alfred P. Sloan Fellow and partially supported by NSF DMS 0547744 and DARPA ONR N66001-06-1-2011. This paper was recommended by Guest Editor G. Setti.

P. K. Yenduri, A. Z. Rocca, A. S. Rao, and M. P. Flynn are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: ypkumar@umich.edu; arocca@umich.edu; aswains@umich.edu; mpflynn@umich.edu).

S. Naraghi was with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA. She is now with Tektronix Inc., Santa Clara, CA 95051 USA (e-mail: shahrzad.naraghi@tektronix.com).

A. C. Gilbert is with Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: annacg@umich.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JETCAS.2012.2221832

as [19], are based around conventional high-speed ADCs. Also, none of the above designs exploit time-based conversion techniques to reduce power consumption. The random PPM ADC, thus occupies a unique position in the literature of compressive sensing ADCs.

The remainder of the paper is organized as follows. Section II briefly introduces compressive sampling and techniques that reduce the number of measurements needed to store and reconstruct a given input signal. The PPM and random PPM architectures are discussed in Section III. In contrast to [20], in which we present a mathematical model for random PPM, in this paper, we also discuss the motivation for and the design of the randomized ADC. A prototype random ADC, implemented as a custom complementary metal-oxide-semiconductor (CMOS) PPM ADC coupled to an field-programmable gate array (FPGA) is described. The hardware implementation of the random PPM ADC is described in Section IV. The problem of reconstructing the input signal from ADC output samples is introduced in Section V. In Section VI, we discuss the PRSreco algorithm for the recovery of input signals that satisfy the signal model presented in Section V-A. This algorithm falls under the general category of greedy pursuit methods that aim to minimize the norm of the reconstruction error, subject to the sparsity conditions of the input signal. The PRSreco is analyzed in Section VI-A. The error bound of the recovered signal is discussed in Section VI-A. Unlike [20], this paper contains a complete and detailed theoretical analysis of the hardware system and the algorithm. Appendixes A–E contain details about the mathematical modeling of the randomized sampling system along with lemmas and theorems that provide proof of correctness and run-time details of the algorithm.

The PRSreco algorithm can also be used for signal reconstruction in other randomized time based ADCs as the analysis in Section VI-A is easily extended. The algorithm is tailored to reduce computational cost and thus is viable for practical hardware implementation. Our analysis along with the numerical simulations and experimental results presented in Section VII, show that a random sampling time-based ADC exhibits much better performance than a nonrandom ADC operating at sub-Nyquist sampling rates.

II. COMPRESSIVE SAMPLING

The basic idea of compressive sensing or compressive sampling is to exploit redundancy (i.e., sparsity or compressibility) in an input signal in order to reconstruct it from a small set of observations. In other words, compressive sensing aims for “smart” sampling of signals to acquire only the “important” information. In this way, the signal sampling rate can be reduced from the Nyquist rate (i.e., twice the bandwidth) to a rate that is proportional to the actual amount of information present in the signal.

Let the signal of interest be represented by a vector x of length N . We say that x is sparse if it contains only a few nonzero components compared to the total length (N) of the signal. A compressible signal is one that is reasonably well approximated as a sparse signal. Let a vector $y = Ax$ represent linear measurements taken from x by the measurement system. The matrix, A , is called the measurement matrix and has a size $K \times N$,

where the number of measurements $K \ll N$. The reduction in the number of measurements that can be tolerated is proportional to the sparsity of the input signal x . The problem of recovering the signal x can be cast as that of solving an under-determined system of equations $Ax = y$. Solving for x based on y is an ill-posed problem in general; however, it is possible to do so under the assumption that the input signal x is sparse or compressible.

The assumption of sparsity is justified by the fact that real world signals are often sparse or compressible in some transform domain. For example, communication signals such as frequency shift keying (FSK) are sparse in the Fourier domain and natural images are often sparse in a wavelet domain. In other words, even if the input signal x is not sparse, it can be represented as $x = WX$ for some sparse vector X , where W denotes the sparsifying transform matrix. The net measurement matrix now changes to $B = AW$ for the system $BX = y$. In this paper, we assume that the input signal is sparse in the frequency (Fourier) domain. Hence, X is the DFT of x and W is the inverse discrete Fourier transform (IDFT) matrix. The sparse spectrum X can be expressed as the solution to the following optimization problem¹ [21]:

$$\arg \min \|X\|_0 \text{ such that } BX = y.$$

The above problem requires the solution of a nonconvex combinatorial problem, which is not practical [22]. Hence, the ℓ_0 -“norm” in the objective function is often replaced by its convex relaxation, the ℓ_1 -norm². That is

$$\arg \min \|X\|_1 \text{ such that } BX = y.$$

It has been shown that if the measurement matrix B satisfies the restricted isometric property (RIP), then the sparse vector X can be recovered exactly [23]. A matrix is said to satisfy RIP with parameters (s, ϵ) for $\epsilon \in (0, 1)$, if for all s -sparse³ vectors z

$$(1 - \epsilon)\|z\|_2 \leq \|Bz\|_2 \leq (1 + \epsilon)\|z\|_2.$$

There is no known algorithm that can verify if a given matrix is RIP other than the exponential time brute force algorithm. However, various results have been published about the RIP nature of the matrix B if it is drawn from certain distributions of random matrices. For example, in the cases where B is a random Gaussian matrix [24], a random Bernoulli matrix [25] or a random partial DFT matrix [24], B satisfies $\text{RIP}(s, \epsilon)$ with high probability, if the number of measurements $K > O(\epsilon^{-2} s \log^{O(1)} N)$ ($O(\cdot)$ refers to the Big-O notation [23]). Algorithms that carry out the ℓ_1 -minimization through linear programming to find X are typically referred to as the Basis Pursuit (BP) algorithms. Many of the proposed CS ADCs ([15], [16], [19]) use BP algorithms for reconstruction. However, BP algorithms are challenging to implement in hardware and are usually significantly slower when compared to greedy pursuit algorithms [26]–[28]. Greedy pursuit algorithms try to minimize

¹ $\arg \min \|X\|_0$ solves for X that has the smallest ℓ_0 -“norm,” where $\|X\|_0$ is defined as the number of nonzero elements in X .

²The ℓ_1 -norm of a vector is defined as the sum of the absolute values of its elements.

³An s -sparse vector has at most s nonzero elements.

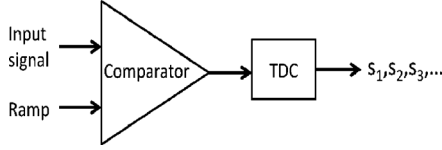


Fig. 1. Block diagram of the PPM ADC.

the ℓ_2 -norm of the error (defined as $BX - y$) subject to the condition that X is s -sparse

$$\min \|BX - y\|_2 \text{ such that } \|X\|_0 \leq s.$$

Conventional greedy pursuit algorithms such as those proposed in [27] and [28], require the matrix B to be RIP. The measurement matrix B associated with the PPM ADC, does not necessarily satisfy the RIP condition. If a new matrix B is constructed randomly for each input signal X , then the RIP condition on B can be relaxed [26], [29]. Hence, we impose the condition that B be a random matrix (newly constructed for each input signal X) and develop a new reconstruction algorithm, that falls under the category of greedy pursuit algorithms, but does not require matrix B to be RIP.

In this paper, a “measurement” of the input signal is a measurement of the amplitude of the input signal at some time point. To obtain a random measurement matrix B , the input signal is sampled at random time points. The measurement vector y represents the amplitude of the signal at those random time points. Random sampling can be broadly classified into two categories, random on-grid sampling where the sampling time points lie on a Nyquist grid and random off-grid sampling where the time points are continuously distributed in an interval and do not lie on a grid. Several fast sub-linear time algorithms can recover s -sparse signals from random on-grid samples [29]–[32]. These algorithms have a storage requirement and runtime of $O(s \log^{O(1)} N)$ (with the exception of [32], which samples the signal uniformly at Nyquist rate). However, PPM ADC produces off-grid samples. A specific case of random off-grid sampling is studied in [33] with a number of measurements, $K > O(sR^2 \log(4N/\epsilon))$, where R is the dynamic range of X and ϵ is a tolerance parameter. In this paper, we deal with reconstruction from signal-dependent,⁴ random, off-grid samples. Thus, the problem setting is different from [33], leading to an algorithm that offers different error guarantees and different conditions for recovery. We also take a nonconventional approach in proving the error guarantees. Unlike the random on-grid sampling techniques, our algorithm does not achieve a sub-linear run-time.

III. HARDWARE DESIGN

In this section, we describe the PPM ADC architecture and the design of the random PPM ADC design.

A. PPM ADC Architecture

A block diagram of the PPM ADC is shown in Fig. 1. The sampling procedure [13] is depicted in Fig. 2. A comparator

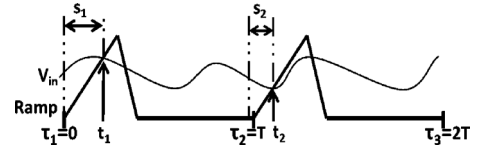


Fig. 2. Waveforms depicting the sampling procedure in the PPM ADC.

continuously compares the input signal with a repeating reference ramp. An output pulse is generated by the comparator at the time instants where the ramp voltage exceeds the input signal. The time elapsed between the beginning of the ramp and the instant the input signal crosses the ramp (i.e., s_1, s_2, \dots , as seen in Fig. 2) is measured and quantized by a two-step 9-bit time to digital converter (TDC). The simplest form of a TDC is a digital counter, however, to achieve a high resolution, one needs to have a very high counter frequency which in turn leads to a large energy consumption. On the other hand, delay line circuits [34] are more energy efficient for time measurement, however, the delay line must be long to measure long periods of time and can suffer from nonlinearity. As a compromise, the two-step TDC consists of a 5-bit counter which performs coarse quantization and a delay line TDC as the fine quantizer that resolves 4-bits. By combining a low frequency counter and a delay line TDC, the two-step TDC thus achieves both energy efficiency and a large dynamic range. Detailed implementation of the TDC is discussed in [13]. The output of the ADC is a sequence of time duration measurements s_i , which represent the relative position of the output pulse in every ramp period. Since the signal information is encoded into the position of the pulse, the ADC is called pulse position modulation ADC. The starting points of the ramps are given by $\tau_i = (i - 1)T$, where T is the period of the reference ramp signal. The crossover times are $t_i = \tau_i + s_i$. If we assume the slope of the ramp is a constant m , the signal amplitude at the crossover times is $y_i = ms_i$. In this way, from the output $\{s_i\}$ of the PPM ADC, we can calculate the sample set $\{(t_i, y_i), i = 1, 2, \dots\}$. Note that T is also the average sampling period of the ADC, because the ADC takes one sample within every interval of T seconds.

1) *Nonuniform Signal Dependent Sampling*: If we make the approximation that y_i are samples at uniform time points τ_i instead of the nonuniform t_i , we see harmonic distortion in the frequency spectrum of the recovered signal. Linear low-pass filtering is a straightforward conventional technique for constructing uniform samples from nonuniformly sampled information. According to [35] an oversampling factor of at least 8 is needed to use the traditional low pass filtering technique. Another approach is to use a time-varying iterative nonlinear reconstruction method, (as described in [13]) which allows the signal to be sampled closer to the Nyquist rate (see Appendix A for further details). The method still requires the sampling frequency to be above the Nyquist rate (the oversampling factor of 8 is brought down to 2). Further, a sufficient condition of $s_i < T/4$ is required to obtain a stable sampling set [36]. If this condition is relaxed, there is no guarantee that the algorithm converges. For sampling rates below the Nyquist rate, the method diverges. Our goal is to convert the PPM ADC into a compressive sampling ADC so that the signal can be recovered from

⁴The time points at which the signal is sampled depend on the signal, in contrast to being completely deterministic or completely random.

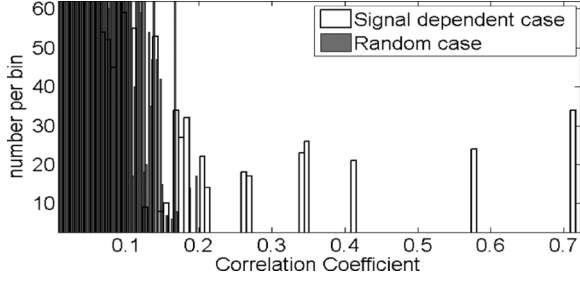


Fig. 3. Histogram of correlation coefficients between different pairs of columns of a signal dependent measurement matrix and a random measurement matrix (of size 15×40). The y -axis represents the number of correlation coefficients that fall in any particular bin of coefficient values.

samples acquired at sub-Nyquist rate. The sampling system and the reconstruction algorithm are co-designed to achieve this.

2) *A Regular PPM ADC at Sub-Nyquist Sampling Rate:* A straight-forward way to operate a PPM ADC as a compressive sampling ADC is to increase its average sampling period T (which is also the reference ramp period). The sampling frequency F can be brought down to a value $F < F_N$, where F_N is the Nyquist frequency of the input signal. We refer to this sampling architecture as the *regular* PPM ADC. We use the algorithm proposed in Section VI for reconstruction. However, since the time points t_i (calculated in Section III-A) are non-random and highly signal dependent, the resultant measurement matrix B is also nonrandom and thus disobeys the design rules of random sampling algorithms. In order to fit into the compressive sensing framework and to meet the criteria for successful signal reconstruction, we need to make some modifications to the PPM ADC sampling system. A random sampling scheme is introduced in the next section.

B. Random PPM ADC Design

Simple theoretical results from sparse approximation state that a low correlation between different columns of a measurement matrix indicates the possibility of better signal recovery [26]. Albeit crude, this is one of the elementary methods for evaluating a measurement matrix with respect to its reconstruction properties. Consider the following simple experiment to motivate the introduction of randomness into the PPM ADC. Let B^c be the measurement matrix that relates the DFT of the input signal to the samples obtained by the PPM ADC at sub-Nyquist rate with an average sampling period of T . As discussed, the time points t_i at which a signal is sampled by the PPM ADC are signal dependent. Now let B^r be the corresponding measurement matrix, when in each interval $[(i-1)T, iT]$ of length T , the signal is sampled at a time point t_i that is uniformly distributed on $[(i-1)T, iT]$ ⁵. Ideally, we want any measurement matrix B to be orthonormal ($B^H B = I$) so that the input signal can be easily recovered as $B^H y = B^H B X = X$. An orthonormal matrix is characterized by a zero correlation between any two columns of the matrix. Fig. 3 plots the histograms of correlation coefficients between different columns, for both the signal dependent B^c and the random B^r matrices. As can be seen from the figure, in the case of a signal dependent B^c several

⁵Note that in the actual implementation of the random PPM ADC, we do not have complete control over $t_i = \tau_i + s_i$, and so we randomize τ_i instead.

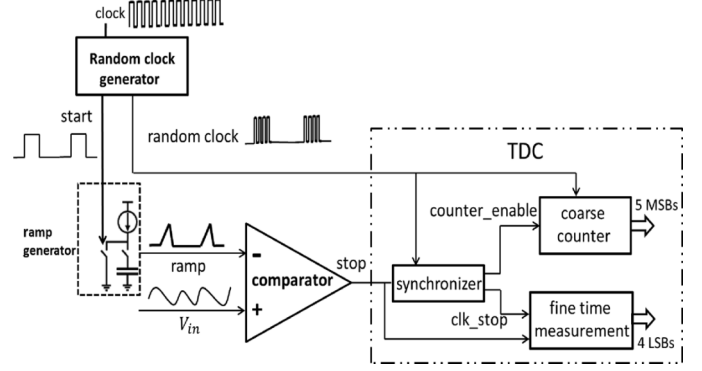


Fig. 4. Block diagram of the random PPM ADC also showing the building blocks of TDC block.

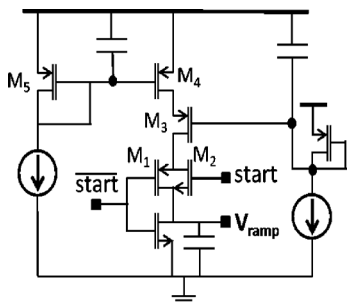
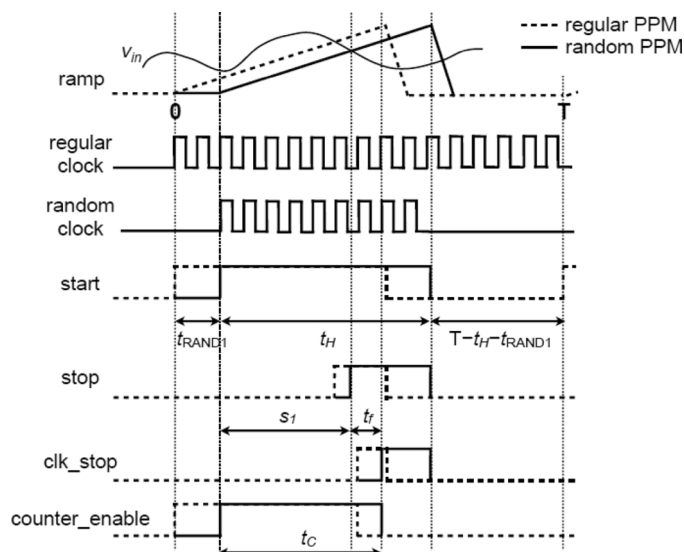
columns have high correlation coefficients. On the other hand for the random matrix B^r , coefficients are all distributed in the left region of the plot. Intuitively, since B^r achieves closer to zero correlation coefficients when compared to B^c , it is more “orthonormal” than B^c and is expected to lead to a better signal recovery.

Motivated by this observation, we introduce randomness into the PPM ADC system. We convert the ramp starting times τ_i (which are deterministic in regular PPM ADC) into random variables. More specifically, let $\tau_i - (i-1)T \sim \text{uniform}[0, T]$, $\forall i$. That is, in each interval $[(i-1)T, iT]$ of length T , the reference ramp has a random starting point this definition of τ_i as it is, may lead to overlap between adjacent ramps. For example, there is an overlap when both τ_1 and τ_2 are equal to T . See Appendix B for further details about how to avoid this. We call this architecture *random* PPM, as the ramp starting times are now randomly and independently chosen. As before, the crossover times are $t_i = \tau_i + s_i$ and the signal amplitude at the crossover times is $y_i = m s_i$, where m is the slope of the ramp.

IV. RANDOM PPM ADC IMPLEMENTATION

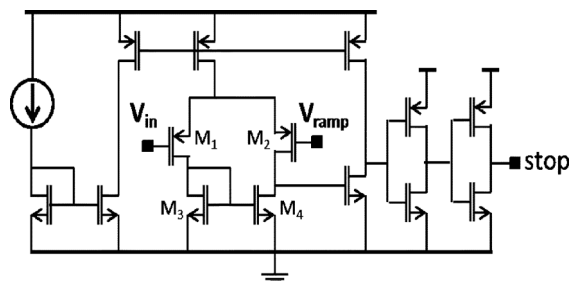
Fig. 4 shows a block diagram of the random PPM ADC. The individual blocks are explained in detail in the following sections. A random clock generator block produces two outputs, a *start* signal (which goes high at each τ_i) and a random clock. The reference ramp is generated only when the *start* signal is high. The comparator compares the output of the ramp generator with the input signal and generates a *stop* signal when the ramp voltage exceeds the input signal. The random clock acts as the time reference in the time-to-digital conversion (TDC) block. The two-step TDC (with a 5-bit coarse quantizer and a 4-bit fine quantizer) measures the time elapsed between the rising edges of the *start* and the *stop* signal. A synchronizer ensures correct alignment of the coarse and the fine time measurements. The ramp generator, comparator and the two-step TDC are implemented in 90-nm digital CMOS, while the random clock generator is implemented on an FPGA.

Fig. 5 shows some of the key timing signals and provides a comparison of operation between a regular PPM ADC and the random PPM ADC. The regular PPM ADC receives a regular periodic clock with period T_{clk} . The *start* signal of a regular



PPM ADC goes high at the beginning of each repetition period T . On the other hand, the random PPM receives a random clock, which consists of regular clock cycles only when the *start* signal is high. The *start* signal for the random PPM ADC goes high after a random time t_{RAND} in each interval (of length T). The *start* signal remains high only for a time t_H and then goes low for the rest of the interval. The time t_H is related to the slope m of the ramp such that the ramp covers the entire voltage range of the input signal in a time t_H . During the time $[t_{\text{RAND}}, t_{\text{RAND}} + t_H]$ when the *start* is high, the ramp is generated and when it crosses the input signal, the random PPM ADC makes one measurement (denoted as s_1 in the figure). This process repeats in every interval $[(i-1)T, iT]$, $i = 1, 2, \dots$. Therefore, the average sampling frequency of the ADC is $F = 1/T$.

The ramp generator circuit is shown in Fig. 6. Charging a capacitor with a constant current produces the ramp signal. Cascoded PMOS transistors M3 and M4 implement the current source while M1 and M2 are the digital switches that control capacitor charging. The switches are, in turn, controlled by the *start* signal. The capacitor discharge is achieved simply with a switch to ground [13].



The comparator is continuous time and is made up of two stages. The circuit is shown in Fig. 7. The first stage is a differential to single amplifier with a PMOS input pair. This is followed by an NMOS common source stage. The PMOS input pair operates in the sub-threshold region. This is done to minimize power consumption and to provide a larger input common mode range, which allows for a larger dynamic range in the ramp [13].

The random waiting times t_{RAND} 's (in Fig. 5) are produced using a linear feedback shift register (LFSR) system as shown in Fig. 8. Although the output of this system is only pseudo-random, with a large bit length sufficient randomness is achieved. The LFSR bit string is initialized with a nonzero seed. This bit string gives the number of regular clock cycles that the *start* signal initially remains low, i.e., $t_{\text{RAND}} = (\text{LFSR})T_{\text{clk}}$, where LFSR stands for the (integer) value of the bit string. The *start* signal then goes high and stays high for a time t_H (which is chosen such that mt_H is greater than the input signal voltage range; t_H is also chosen to be a multiple of T_{clk}). To complete the interval of length T , the *start* signal is kept low for an additional $T - t_H - t_{\text{RAND}}$ seconds, as shown in Fig. 5. Once a complete interval of length T has elapsed, the LFSR sequence is advanced to its next state and the same process is repeated with the new value of LFSR (thus, a new $t_{\text{RAND}} = (\text{LFSR})T_{\text{clk}}$). The random clock is produced by gating the *start* signal with the regular clock, as shown in Fig. 8. The rising edges of *start* and the random clock are thus synchronized. Note that two short bit length LFSR systems can be coupled to produce a pseudo-random sequence with sufficiently large period.

D. Time-to-Digital Converter

The two-step TDC [13] measures the time interval (s_1 in the Fig. 5) between the rising edges of the *start* signal (which is synchronous with the random clock), and the *stop* signal generated by the comparator. To enable correct alignment of the coarse and fine time measurements, the synchronizer block generates two additional signals, *clk_stop* and *counter_enable*. The *clk_stop* signal is set by the arrival of the second rising edge of the clock after the *stop* signal. The *counter_enable* signal is set by *start* and reset by *clk_stop*. A 5-bit counter (which is the coarse quantizer) measures t_c (in Fig. 5), which is the number of clock cycles elapsed while the *counter_enable* signal is high. The slope of the ramp is designed such that t_c is always less than 32 clock cycles. The fine TDC measures the time t_f between the *stop* signal and *clk_stop* signal rising edges. The overall TDC output is $s_1 = t_c - t_f$. The fine TDC consists of a 32-element delay line, spanning two full clock cycles (the fine TDC thus divides one clock cycle into 16 equal slices and resolves 4 LSBs).

V. RECONSTRUCTION PROBLEM

We now formulate the problem of reconstructing the input signal from samples collected by an ADC (regular PPM or random PPM). The samples are assumed to be collected at a sub-Nyquist rate. Let an N -length vector X represent the input signal in the Fourier domain. Let K ($K < N$) be the number of measurements taken by the ADC. Let (t_i, y_i) , $i = 1, \dots, K$ denote the measurements obtained from the output of ADC (see Section III). The time t_i is the i th time point at which the ADC samples the input signal and y_i is the signal amplitude at that time. Note that $K/N = F/F_N < 1$, where $F = 1/T$ is the average sampling frequency of the ADC and F_N is the Nyquist rate of the input signal. We relate the input signal X with the measurement vector y through the equation $BX = y$, where B is the measurement matrix. The goal is to solve for X from $BX = y$. Note that X is the N -point DFT of the time domain input signal x . The reconstruction is done in the frequency domain, as the input signal is assumed to be sparse in frequency domain as indicated in the following input signal model.

A. Signal Model

In this paper we focus only on a subset of band limited signals that are band limited to $[-W, W]$. The Nyquist rate of the input signal space is $F_N = 2W$. If the input signal is sampled at the Nyquist rate for a time of t_{Total} , then the number of samples $N = F_N t_{\text{Total}}$. We assume that the input signal is s -sparse or s -compressible in the frequency domain. A signal is called s -sparse in the frequency domain, if the DFT of the signal samples at Nyquist rate has only s nonzero terms. A signal is called s -compressible⁶ in the frequency domain, if the sorted list of its DFT coefficients has only s significant or dominant terms, compared to which the other terms are negligible. The input signal

⁶We call X , s -compressible, if it is well approximated as a s -sparse signal, $\|X - X_{(s)}\|_2 \leq C \cdot s^{-\alpha}$ for some constants C and $\alpha > 0$, where $X_{(s)}$ is the s -sparse signal that best approximates X .

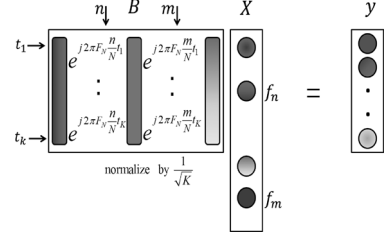


Fig. 9. Measurement matrix.

can be expressed as a linear combination of complex exponentials as follows:

$$x(t) \approx \sum_{m=1}^s c_m \exp(j2\pi f_m t)$$

where f_m , $m = 1, \dots, s$ are the s dominant frequencies which lie in the interval $[-W, W]$ and c_m are the corresponding coefficients. We further assume that the input signal is real, hence s is even and one set of frequencies is the negative of the other set. Some practical signals that are frequency-sparse include frequency-hopping communication signals, narrowband transmissions with an unknown carrier frequency that can lie anywhere in a wide band, communication to submarines, radar [37], and geophysical [38] signals such as slowly varying chirps, etc.

B. Measurement Matrix

To determine whether a successful signal recovery is possible from $BX = y$, we analyze the properties of the measurement matrix B , which is shown in Fig. 9. The matrix B can be intuitively constructed by making the observation that if f is the only frequency with a nonzero coefficient in the DFT X , i.e., in time domain $x(t) = \exp(j2\pi f t)$, then the samples of $x(t)$ at time points t_i are given by $\{\exp(j2\pi f t_i), i = 1, \dots, K\}$. Putting $f = (n/N)F_N$ (as in a N -point IDFT), the samples form the n th column of the measurement matrix, for $n = [-N/2 : N/2 - 1]$ (if N even) or $[(-N - 1)/2 : (N - 1)/2]$ (if N odd). Hence, for a given i and n , $B_{i,n} = \exp(j2\pi(n/N)F_N t_i)$. It is to be noted that B is not a sub-matrix of the N -point IDFT matrix, since t_i are nonuniform and do not lie on any Nyquist grid.

We now look at the correlations between different columns of the random PPM measurement matrix B . Let $C_{nm} = B_m^H B_n$ denote the correlation between the n th and m th columns of B .

Lemma 1: Let $I_f = [-N/2 : N/2 - 1]$ (if N even) or $[(-N - 1)/2 : (N - 1)/2]$ (if N odd). For $n, m \in I_f$ and $n \neq m$

$$|E(C_{nm})| \leq O\left(\frac{1}{N}\right). \quad (1)$$

See Appendix C for proof. The Lemma 1 provides a tight upper-bound, on the order of $1/N$, for the magnitude of expected correlation between different columns of B . A small expected correlation implies a better signal recovery, as discussed in Section III-B and illustrated in Fig. 3.

TABLE I
PERIODIC RANDOM SAMPLING RECONSTRUCTION (PRSRECO) ALGORITHM

PRSreco algorithm	
INPUT: N (signal length), s (sparsity), $(t_i, y_i), i = 1, 2, \dots, K$.	
OUTPUT: \tilde{X} (s -sparse approximation to X , length N)	
$\tilde{X} = \mathbf{0}$, residual $r^{(0)} = y$ $T = \text{supp}\{[B^H y]_{(2s)}\}$ $\tilde{X}_T = (B_T^H B_T)^{-1} B_T^H y$ (Least Squares) $r^{(0)} = r^{(0)} - B_T \tilde{X}_T$ for $i = 0, 1, 2, \dots$ $\tilde{X}^{(i+1)} = \tilde{X} + B^H r^{(i)}$ $\tilde{X} = [\tilde{X}^{(i+1)}]_{(s)}$ $r^{(i+1)} = y - B \tilde{X}$ until $\ r^{(i+1)}\ _2$ does not vary within a tolerance θ .	

VI. RECONSTRUCTION ALGORITHM

The random PPM ADC samples the signal at a rate proportional to its finite rate of innovation, defined as the number of degrees of freedom per unit time [39]. For the signal model considered in this paper, the rate of innovation is given by s , the number of frequencies present in the signal. Algorithms have been proposed in [39] that can recover the s frequencies and their coefficients by using only $2s$ consecutive uniform samples from the signal. However, these algorithms cannot be applied with the random PPM ADC as they require the samples to be uniformly spaced at Nyquist rate. Also, the measurement matrix B associated with random PPM ADC, is a signal-dependent nonuniform random Fourier matrix, and as such, does not necessarily satisfy the RIP assumed in [27] or the conditions assumed in [28]. This leads to the need to develop different algorithms with different theoretical analysis. A probabilistic approach is presented in Section VI-A.

We call the developed reconstruction algorithm, Periodic Random Sampling reconstruction (PRSreco). A pseudo-code for the PRSreco algorithm is presented in Table I. From Lemma 1, we see that correlations between different columns of B are small on average. Hence, $B^H y$ is a good approximation to the signal X . In particular, the largest components in $B^H y$ provide a good indication of the largest components in X . The algorithm applies this idea iteratively to reconstruct an approximation to the signal X . At each iteration, the current approximation induces a residual, which is the part of the signal that has not been approximated yet. The current approximation vector \tilde{X} is initialized to a zero vector and the residual is initialized to the measurement vector y . For a vector z , $\text{supp}(z)$ is defined as set of indices of the nonzero elements of z and $z_{(s)}$ stands for the best s -term approximation⁷ of z . For an index set $T \subset \{1, 2, \dots, N\}$, z_T stands for a sub-vector of z containing only those elements of z that are indexed by T . Similarly B_T stands for a sub-matrix of B containing only the columns of B indexed by T . The algorithm initially obtains an estimate for the dominant frequencies in the signal through least squares and then refines the estimate of the set of dominant frequencies and their coefficients in an iterative fashion.

⁷The best s -term approximation of a vector z can be obtained by equating all the elements of z to zero, except the elements that have the top s magnitudes.

The computationally intensive step of least squares is performed only once in the PRSreco algorithm. The least squares is implemented using the accelerated Richardson iteration [40] with runtime of $O(sK \log(2/e_t))$ where e_t is a tolerance parameter. The structure of the measurement matrix lends us to use the inverse NUFFT [41] with cardinal B-spline interpolation for forming the products of the form $B^H r$, in a runtime of $O(N \log N)$. Hence the total runtime of the algorithm is dominated by $O(IN \log N)$ where I is the number of iterations.

A. Analysis of Algorithm

Lemma 2 says that the estimators of coefficients of X in the PRSreco algorithm produce close to correct values and their second moments (variances) are bounded. The results of Lemma 2 and Lemma 1 are used to prove Theorem 3.

Lemma 2: If number of measurements $K = O(s/\epsilon^2)$ then for any s -sparse (or s -compressible) vector X , each estimate of the form $\tilde{X}_m = B_m^H B X$ for $m = 1, 2, \dots, N$, satisfies

$$\mathbb{E}(\tilde{X}_m) = X_m \pm O\left(\frac{1}{N}\right) \|X\|_1 \quad (2)$$

$$\text{Var}(\tilde{X}_m) \leq \frac{\epsilon^2}{s} \|X\|_2^2. \quad (3)$$

See Appendix D for proof.

Once the PRSreco algorithm gets an approximation \tilde{X} of X , it subtracts the contribution of the current approximation from the measurements and proceeds to recover the leftover signal $X - \tilde{X}$. As we move on to higher iterations of the algorithm, the energy in the leftover signal goes down, bringing down the upper-bound on the variance of the estimators (from Lemma 2 applied to $X - \tilde{X}$). Thus a better approximation is obtained for the signal X in each higher iteration until the required tolerance is reached or the algorithm converges. Please refer to Appendix E for further details. Theorem 3 offers an error guarantee for a signal recovered using the PRSreco algorithm and establishes the conditions on the sub-sampling ratio K/N that can be achieved using the random PPM ADC. If X is s -sparse and there is no noise in the measurements obtained from the random PPM ADC (operating at a sub-sampling ratio of K/N), then from Theorem 3, signal X can be recovered exactly. If the measurements are corrupted by some noise (e.g., quantization noise), the ℓ_2 -norm of the reconstruction error is bounded above by the ℓ_2 -norm of the noise.

Theorem 3: Let $y = BX + \xi$ be the time domain samples of signal X obtained by the random PPM ADC, where ξ is an arbitrary noise contamination in the measurements and B is the resultant measurement matrix of size $K \times N$. Let the phase⁸ of the time domain input signal $x(t + \phi)$ be uniformly distributed in $[0, 2\pi]$. Suppose $|X_{[s]}|^2 \geq 2\alpha \|X\|_2^2/s + |X_{[s+1]}|$ for some constant α and a given sparsity parameter s , where $|X_{[i]}|$ is the magnitude of the i th largest element of X . Given the error tolerance in reconstruction θ and $K = O(s \log N/\epsilon^2)$, with probability $> 1 - O(\epsilon^2)$ the algorithm produces an s -term estimate \tilde{X} of signal with the following property

$$\|X - \tilde{X}\|_2^2 \leq \max \left\{ \theta^2, \frac{\|X - X_{(s)}\|_2^2}{1 - \alpha} + \frac{c(B)\|\xi\|_2^2}{1 - \alpha} \right\} \quad (4)$$

⁸That is, the time $t = 0$ at which we start to observe the signal, is assumed to be random. This induces a probability distribution on the signal dependent s_i .

where $X_{(s)}$ is the best s -term approximation of X . The runtime of the algorithm is $O(IN \log N)$ where I = Number of iterations, with a gross upper bound of $I < \max(\log N, \log(\|X\|_2/\theta))$. The net storage requirement is $O(N) + O(sK)$. The constant $c(B)$ depends on the measurement matrix B .

See Appendix E for proof.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

The regular PPM and the random PPM sampling architectures (described in Section III), are implemented in hardware. The ADCs, combined with the reconstruction algorithm are also simulated in MATLAB. A series of experiments compares the performance of the PRSreco algorithm for both the sampling architectures. The signal-to-noise ratio⁹ (SNR), which is defined as the ratio between the signal energy and the reconstruction error, is used as the performance metric to evaluate the quality of the reconstructed signal. MATLAB simulation results are presented first and are followed by the experimental results from the hardware implementation.

A. Simulation Results

The finite time resolution t_r of the TDC block in the ADC induces some quantization into the measurements. For the simulation experiments to follow, the quantization is kept at 7 bits ($= \log_2(\text{ramp duration}/t_r)$, for a ramp duration of $0.25 \mu\text{s}$ and $t_r = 2 \text{ ns}$). This corresponds to a signal to quantization noise ratio of about 44 dB for an input sinusoid.

1) *Multitone Signals*: In the first experiment, we reconstruct multi-tone input signals, which are a linear combination of sinusoids. Each sinusoid has a random phase, comparable amplitude and its frequency is chosen randomly from the Nyquist grid. The Nyquist frequency is 3 MHz whereas the sampling frequency of the ADC is chosen to be 1 MHz, giving a sub-sampling ratio of 0.33. That is, $K/N = 0.33$, where K ($= 150$) is the number of measurements from the ADC and N ($= 450$) is the length of input signal, X . The input signal is corrupted by additive white Gaussian noise with varying power, sampled by the random PPM scheme and reconstructed using the PRSreco algorithm. The performance of the algorithm is evaluated by measuring the output SNR. The experiment uses the s -term Nyquist approximation as the benchmark performance, which is defined as the SNR obtained when the signal is sampled at Nyquist rate, quantized at the same quantization level as the ADC and then truncated, in frequency domain, to keep only the s dominant terms. The s -term Nyquist benchmark thus represents the best s -term approximation to the signal in frequency domain. Fig. 10(a) plots the mean (of 200 trials) reconstruction output SNRs for signals with nine tones (corresponding $s/N = 18/450$ and $s/K = 18/150$) and 17 tones ($s/N = 34/450$ and $s/K = 34/150$).

The experiment demonstrates that the random PPM achieves a close to the benchmark¹⁰ performance, owing to the good correlation properties of the measurement matrix (Lemma 1). The

⁹SNR(dB) = $20 \log(\|X\|_2/\|X - \tilde{X}\|_2)$, where X is the input signal and \tilde{X} is the output of the algorithm

¹⁰The benchmark considers the error in the amplitude of the s tones due to quantization and input noise

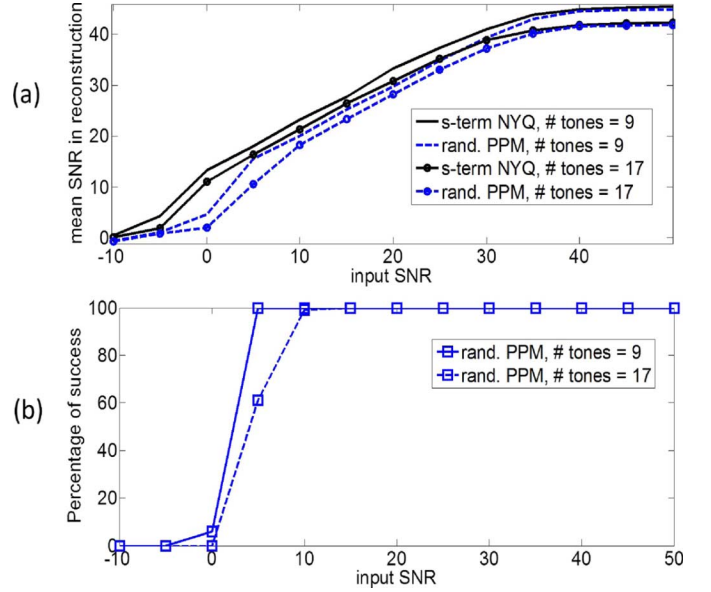


Fig. 10. (a) Mean output SNR versus input SNR and (b) success percentage (fraction of trials that succeed) versus input SNR for 9-tone and 17-tone signals. The s -term NYQ (Nyquist) benchmark represents the best s -term approximation to the signal in frequency domain. Success means the correct identification of the frequencies of all tones.

random PPM performance approaches the benchmark as the input SNR increases. Further more, as the number of tones increases (making the signal less sparse), the random PPM output SNR is unaffected relative to the benchmark.

The output SNR can be higher than the input SNR, as the algorithm (like any other greedy pursuit algorithm) only calculates the coefficients of the top s frequencies in the signal and thus inherently filters out the noise at other frequencies. This “denoising” effect decreases as the value of s increases. This explains the degradation in output SNR (of even the benchmark) when the number of tones is increased. After input SNR is high enough, we see a saturation in the output SNR. This can be attributed to the quantization noise in the measurements (which also gets “denoised” to some extent).

Fig. 10(b) plots the percentage of trials that achieve success in signal recovery. We call the reconstruction a success when the frequencies of all the tones in the input signal are correctly identified. The plot also confirms that mean output SNR is a good indicator of the quality of reconstruction, as it also captures (to some extent) the information about the percentage of success.

2) *Sub-Sampling Ratio*: The next experiment reconstructs a single tone signal (randomly chosen frequency, $s/N = 2/450$) with varying number of measurements and noise levels. The input signal is sampled by both the random PPM and the regular PPM sampling schemes and reconstructed using the PRSreco algorithm. The sub-sampling ratio is defined as the ratio between the sampling rate of the ADC and the Nyquist rate of the signal (which is twice the randomly chosen tone frequency), and can be computed as K/N . The sub-sampling ratio needed for at least 99% success (i.e., at least 99% of the total trials succeed in identifying the input signal frequencies correctly) is empirically determined for each input SNR level and is plotted in Fig. 11(c). We observe that at all SNR levels the random PPM ADC succeeds with far fewer measurements than the regular PPM. Fur-

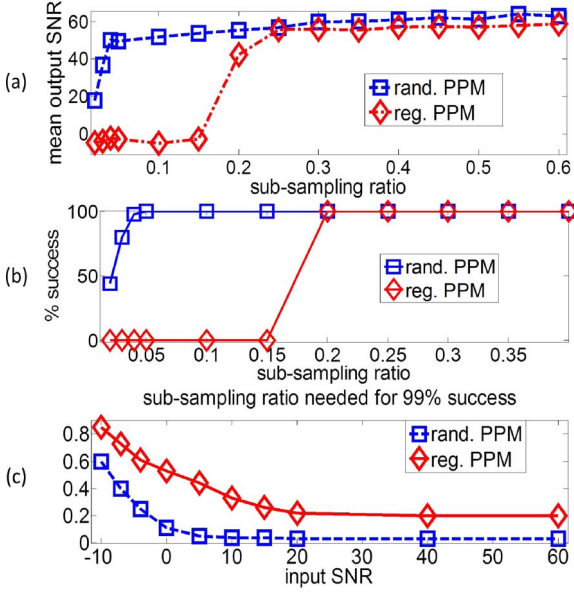


Fig. 11. Reconstruction of a single tone signal with varying number of measurements (a) with no noise (b) success percentage when no noise (c) sampling needed for 99% success, with noise.

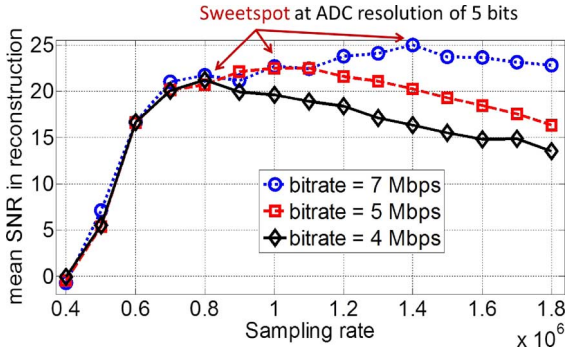


Fig. 12. Mean output SNR versus random PPM ADC sampling rate, for fixed bitrates of 4, 5, and 7 Mb/s.

ther, when the input SNR is high enough the sub-sampling ratio needed for success in the random PPM quickly falls to about 3%. This can also be seen in the no-noise (i.e., only quantization noise) case [Fig. 11(a) and (b)], where the regular PPM scheme breaks down when the sampling rate goes below 20% of Nyquist rate, whereas, the random scheme performs well enough for sampling rates as low as 3% of the Nyquist rate, indicating much better incoherence properties of the measurement matrix.

3) *Resolution Versus Sampling Rate:* In this experiment, we fix the bit-rate of random PPM ADC, that is the product of ADC quantization (resolution) and its sampling rate. For example, a bit-rate of 5 Mb/s can be achieved by choosing an ADC quantization of 5 bits and a sampling rate of 1 MHz. Fig. 12 displays the constant bitrate curves for bitrate values of 4, 5, and 7 Mb/s for a random 11-tone input signal with input SNR of 15 dB. Each curve plots the mean output SNR for varying sampling rate. A low sampling rate corresponds to high ADC resolution and vice-versa (since the bitrate is fixed for each curve). If the sampling rate is too low, resulting in a lack of enough measurements, the reconstruction error increases, degrading the output SNR. If the sampling rate is too high, the output SNR again degrades due to lack of sufficient resolution in each measurement.

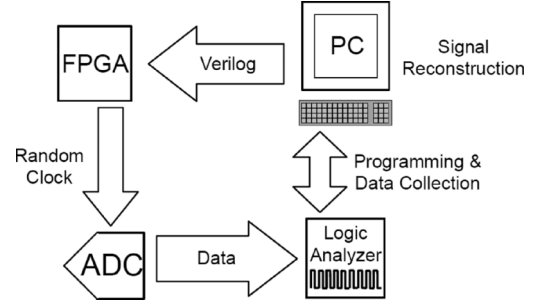


Fig. 13. Hardware setup for the random PPM ADC.

This trade-off results in a sweetspot where the SNR performance is the best. From Fig. 12, we observe that this sweetspot occurs when the ADC resolution is chosen to be about 5 bits.

B. Prototype and Measurement Results

We now present experimental results obtained with the prototype 9-bit random PPM ADC and 9-bit regular PPM ADC. The ramp generator, comparator and the two-step TDC, which are part of both the random PPM and the regular PPM ADCs, are implemented in 90 nm digital CMOS. The linear feedback shift register-based (LFSR-based) random clock generation block is implemented by programming Verilog code onto a field-programmable gate array (FPGA). The analog circuits operate with a 1 V supply, while the digital blocks operate at near-threshold from a 400 mV supply. The regular clock is a 64 MHz signal giving a $T_{clk} = 15.63$ nsec. The LFSR is 9 bits with taps at bin 5 and 9 resulting in a LFSR periodicity of 511. The entire evaluation setup of the random PPM ADC consists of four main blocks as displayed in Fig. 13, an FPGA, the ADC, a logic analyzer and a computer. The FPGA generates the *start* and the random clock signals, which are input to the PPM ADC. The ADC measurements are collected by the logic analyzer. The nonzero seed used to initialize the LFSR system is assumed to be known during reconstruction, so that the sequence of t_{RAND} 's can be calculated.

A single tone input signal is sampled both by the random PPM ADC prototype and the regular PPM ADC prototype, operating at various sampling rates, and reconstructed using the PRSreco algorithm. The results are displayed in Fig. 14. Also displayed for convenience is the compression loss (root mean square error of the reconstruction) on the right y-axis. Note that since the ADC resolution is fixed, the compression achieved by the sampling scheme only depends on the sub-sampling ratio K/N . As expected, the random PPM performs much better than the regular PPM which breaks down when the sub-sampling ratio is around 0.7, whereas the random PPM works well for sub-sampling ratios as low as 0.05. A compression ratio of 0.05 in the random PPM ADC and 0.7 in the regular PPM ADC, both result in the same compression loss of 0.77. Fig. 15 shows the reconstruction of a 5-tone signal with frequencies arbitrarily chosen from the Nyquist grid on [0, 1 MHz] (Nyquist rate = 2 MHz). The multi-tone signal was sampled with the random PPM ADC operating at a sampling frequency of about 173 KHz which leads to a sampling percentage of about 8.65%. The SNR of the recovered signal is 41.6 dB.

The measured power consumption of the PPM ADC system is 14 μ W (excluding digital post-processing). The analog and

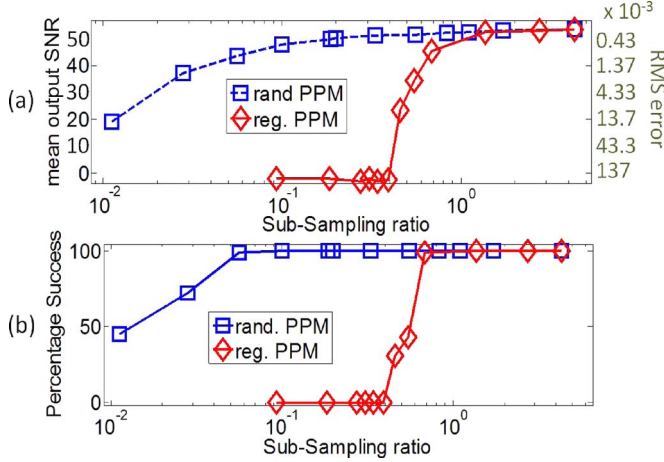


Fig. 14. Reconstruction of a single tone signal from samples collected by the regular and the random PPM ADC prototypes operating at varying sampling rates. The y-axis on the right displays the corresponding root mean square (rms) error.

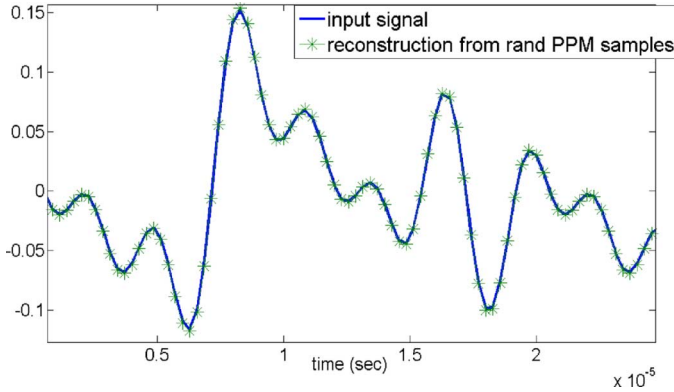


Fig. 15. Reconstruction of a five-tone signal from samples collected by the random PPM ADC with sampling rate at 8.65% of the Nyquist rate.

digital blocks each consume $7 \mu\text{W}$. For the random PPM ADC system, the expected improvement in the power by a factor of K/N (the sub-sampling ratio) is observed, however this does not include the power consumed by the random clock generator and the comparator, both of which operate all the time. The random clock signal is produced by the FPGA and the power consumption of the FPGA itself is not a good indication of the actual power needed since power is consumed by unnecessary circuitry in the FPGA. Implementing the random clock generation on the CMOS IC along with the rest of the compressive sensing ADC would only minimally increase the power consumption of the IC as the LFSR system only requires on the order of ten shift registers and a few gates. The PPM ADC itself uses approximately 50 registers and gates [13], therefore, the digital power consumption due to the addition of LFSR system, is expected to increase by only 6%–8%. An additional power reduction can be achieved by switching the continuous-time comparator off, when not in use. Thus, the power consumption of the random PPM ADC (with on-chip random clock generation) is estimated to be about $(2(K/N) + 0.07)7 \mu\text{W}$. For a random PPM ADC operating at 20% of the Nyquist sampling rate ($= 1 \text{ MHz}$), the estimated power consumption is $3 \mu\text{W}$.

VIII. CONCLUSION

We propose a new, low-power compressive-sampling analog to digital converter, called the random PPM ADC. It inherits the advantages of time to digital conversion and also exploits compressive sampling techniques, to improve the power efficiency of data conversion. An existing PPM ADC design is modified to achieve a 9-bit random PPM ADC, through the use of a random clocking technique. The new random design enables the reduction of the average sampling rate to sub-Nyquist levels and thus reduces the ADC power consumption by a factor close to the sub-sampling ratio. The random PPM performs much better than a regular PPM operating at a sub-Nyquist sampling rate, in terms of obtaining closer-to-benchmark output SNR and handling signals that are less sparse. The proposed reconstruction algorithm is not only faster (greedy pursuit versus basis pursuit inspired algorithms in the literature for compressive sampling ADCs) but also feasible for a hardware implementation. With on-chip reconstruction and a low power front-end, the random PPM ADC is attractive for power constrained applications such as wireless sensor networks, as it reduces both the power consumption and the amount of data that needs to be communicated by each sensor node.

APPENDIX A

RECONSTRUCTION ALGORITHM PROPOSED IN [13]

Let x be the input signal and y be the measurements from the PPM ADC. As discussed y is the set of nonuniform samples from x . We can represent the measurement as $y = Sx$ where S is the nonuniform sampling operator. Let operator P represent a low pass filter with cut-off frequency tuned to Nyquist frequency. The algorithm used is as follows:

$$\begin{aligned} x_0 &= y = Sx \\ x_1 &= Px_0 = PSx \\ x_{i+1} &= P(y - Sx_i) + x_i, \quad \text{for } i = 1, 2, \dots \end{aligned}$$

It is easy to see that $x_i = PS(\sum_{k=1}^i (I - PS)^k)x$, where I is the identity operator and $(I - PS)^0 = I$. $\text{Lt}_{i \rightarrow \infty} \sum_{k=1}^i (I - PS)^k = (PS)^{-1}$ and thus $\text{Lt}_{i \rightarrow \infty} x_i = (PS)(PS)^{-1}x = x$. When the PPM ADC is operated at Nyquist rate, applying low pass filter to nonuniform samples causes harmonic distortion which can be corrected through the iterations. However, below Nyquist rate, applying a low pass filter to nonuniform samples causes severe aliasing in the frequency domain which cannot be rectified through iterations. In other words, the operator PS cannot be inverted through the algorithm used.

APPENDIX B

ADDITIONAL DESIGN DETAILS OF RANDOM PPM

Assume that the duration of the reference ramp, that is the time for which the ramp is greater than zero in each period is given by cT for some $0 < c < 1$. For the original PPM, $c \leq 0.25$, so as to satisfy the stability condition for the reconstruction algorithm presented in Appendix A. Choosing $\tau_i - (i-1)T \sim \text{uniform}[0, T]$, $\forall i$ can cause overlap between adjacent ramps. For example, when $\tau_1 = T$ and $\tau_2 < T + cT$ there is an overlap. There are two ways to deal with this issue. The first is to adjust

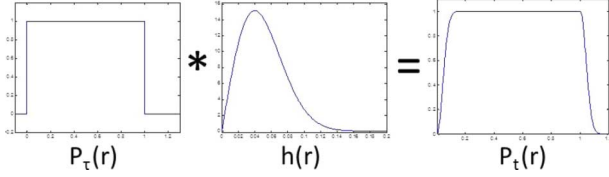


Fig. 16. Example pdfs of τ , s and $t = \tau + s$.

the distribution of τ_i as $\tau_i - (i-1)T \sim \text{uniform}[0, T - cT]$, $\forall i$. The implemented prototype random PPM ADC uses this adjustment. Another way is to employ a second sampling system. The sampling systems each produce a ramp in alternate periods and sample the input alternatively in each period. Thus, there is no overlap in the ramps and the original choice of distribution for τ_i can be maintained, that is $\tau_i - (i-1)T \sim \text{uniform}[0, T]$, $\forall i$. Note that τ_i are all still independently chosen. Also note that the net power consumption can be kept almost same as before since the two samplers would sample at half the original rate. Even though we do not actually use two sampling systems in the prototype random PPM ADC, we assume that the overlap between adjacent ramps is allowed for theoretical simplicity. The presented “mathematical framework” thus closely matches the implementation (actual or thought experiment).

The time points at which the random PPM ADC samples the signal are given by $t_i = \tau_i + s_i$. To further aid the analysis, we assume that the phase ϕ of the input signal $x(t + \phi)$, is uniformly distributed in $[0, 2\pi]$. This induces a probability distribution on s_i . The probability density function (pdf) of t_i can be obtained by convolving the pdfs of τ_i and s_i (since τ_i and s_i are independent for all i). Dropping the i for convenience, let $h(r)$, $0 \leq r \leq cT$ be the pdf of s (not to be confused with sparsity of the signal s). Recall that cT is the on-time of the reference ramp in each period. The pdf of τ is $p_\tau(r) = 1/T$, $0 \leq r \leq T$. Thus convolving the two it is easy to see that pdf of $t = \tau + s$ is given by

$$p_t(q) = \begin{cases} \frac{H(q)}{T}, & 0 \leq q \leq cT \\ \frac{1}{T}, & cT \leq q \leq T \\ \frac{1-H(q-T)}{T}, & T \leq q \leq T + cT \end{cases}$$

where $H(q)$ is the cumulative distributive function of the random variable s . An example of $h(r)$ and $p_t(r)$ is shown in the Fig. 16. These results are used in the proof of Lemma 1 in Appendix C.

APPENDIX C PROOF OF LEMMA 1

Proof:

$$\begin{aligned} \mathbb{E}(C_{nm}) &= \mathbb{E}(B_m^H B_n) = \mathbb{E}\left(\sum_{i=1}^K \frac{1}{K} \exp\left(j \frac{2\pi}{NT_N} (n-m)t_i\right)\right) \\ &= \sum_{i=1}^K \frac{1}{K} \mathbb{E}(\exp(\theta t_i)) \end{aligned}$$

where $\theta = j2\pi(n-m)/NT_N$ for convenience and $T_N = 1/F_N$. Now, using the distribution of t_1 derived in Appendix B, it can be proven that

$$\mathbb{E}(\exp(\theta t_1)) \leq (\exp(\theta T) - 1) \frac{\exp(\theta cT)}{\theta T}.$$

Now it is easy to obtain that

$$\mathbb{E}(\exp(\theta t_i)) = \exp(\theta(i-1)T) \mathbb{E}(\exp(\theta t_1)).$$

Hence

$$\begin{aligned} \mathbb{E}(C_{nm}) &\leq \frac{1}{K} \left(\sum_{i=1}^K \exp(\theta(i-1)T) \right) \mathbb{E}(\exp(\theta t_1)) \\ &= \frac{1}{K} \frac{\exp(\theta KT) - 1}{\exp(\theta T) - 1} \mathbb{E}(\exp(\theta t_1)) \\ &= \frac{\exp(\theta KT) - 1}{\theta KT} \exp(\theta cT). \end{aligned}$$

Now (using $|\exp(\theta cT)| = 1$ and then $NT_N \leq KT < (N+1)T_N$) we have

$$\begin{aligned} |\mathbb{E}(C_{nm})| &\leq \left| \frac{\exp(\theta KT) - 1}{\theta KT} \right| = \text{sinc}\left((n-m) \frac{KT}{NT_N}\right) \\ &\leq \text{sinc}\left((n-m) \left(1 + \frac{1}{N}\right)\right) \leq \frac{1}{(N+1)} \leq \frac{1}{N}. \end{aligned}$$

APPENDIX D PROOF OF LEMMA 2

Proof: For any vector X

$$\begin{aligned} \mathbb{E}(\tilde{X}_m) &= \mathbb{E}\left(\sum_{i=1}^N B_m^H B_i X_i\right) = \mathbb{E}\left(X_m + \sum_{i=1, i \neq m}^N C_{im} X_i\right) \\ &= X_m + \sum_{i \neq m} \mathbb{E}(C_{im}) X_i. \end{aligned}$$

The required result is now true from Lemma 1.

Now we will compute $\text{Var}(\tilde{X}_m) = \mathbb{E}(\tilde{X}_m^2) - (\mathbb{E}(\tilde{X}_m))^2$. But first, consider the following:

$$\begin{aligned} C_{im}^H C_{\ell m} &= \frac{1}{K^2} \sum_{n=1}^K e^{j2\pi(m-i)t_n/NT_N} \sum_{q=1}^K e^{j2\pi(\ell-m)t_q/NT_N} \\ &= \frac{1}{K^2} \sum_{n=1}^K e^{j(2\pi/NT_N)(m-i+\ell-m)t_n} \\ &\quad + \frac{1}{K^2} \sum_n \sum_{q \neq n} e^{j2\pi(m-i)t_n/NT_N} e^{j2\pi(\ell-m)t_q/NT_N}. \end{aligned}$$

After applying expectation, and using that t_n and t_q are independent for $n \neq q$ and also using the Lemma 1, we see that the second term above can be ignored as it is $O(1/N^2)$. Hence

$\mathbb{E}(C_{im}^H C_{\ell m}) \approx \mathbb{E}(C_{\ell i})/K$. We will use this in the expansion for $\mathbb{E}(\tilde{X}_m^2)$ as follows:

$$\begin{aligned} \mathbb{E}(\tilde{X}_m^H \tilde{X}_m) &= X_m^2 + \sum_{\ell=1, \ell \neq m}^N \mathbb{E}(C_{\ell m}) X_m^H X_\ell \\ &\quad + \sum_{i=1, i \neq m}^N \mathbb{E}(C_{mi}) X_i^H X_m + \sum_{i \neq m} \sum_{\ell \neq m} \frac{\mathbb{E}(C_{\ell i})}{K} X_i^H X_\ell \\ &\leq X_m^2 + \sum_{\ell=1, \ell \neq m}^N \mathbb{E}(C_{\ell m}) X_m^H X_\ell \\ &\quad + \sum_{i=1, i \neq m}^N \mathbb{E}(C_{mi}) X_i^H X_m + \sum_{i=1, i \neq m}^N \frac{1}{K} X_i^2 \end{aligned}$$

since, by Lemma 1, $\mathbb{E}(C_{\ell m}) \leq 1/N$, is negligible for $\ell \neq m$. We can similarly obtain the expansion

$$\begin{aligned} (\mathbb{E}(\tilde{X}_m))^2 &= X_m^2 + \sum_{\ell=1, \ell \neq m}^N \mathbb{E}(C_{\ell m}) X_m^H X_\ell + \sum_{i=1, i \neq m}^N \mathbb{E}(C_{mi}) X_i^H X_m \\ &\quad + \sum_{i=1, i \neq m}^N \sum_{\ell=1, \ell \neq m}^N \mathbb{E}(C_{mi}) \mathbb{E}(C_{\ell m}) X_i^H X_\ell. \end{aligned}$$

The last term can be ignored (assuming signal sparsity $s \ll N$). Now

$$\begin{aligned} \text{Var}(\tilde{X}_m) &= \mathbb{E}(\tilde{X}_m^H \tilde{X}_m) - (\mathbb{E}(\tilde{X}_m))^2 \\ &\leq \sum_{i=1, i \neq m}^N \frac{1}{K} X_i^2 \leq \frac{\epsilon^2}{s} \|X\|_2^2. \end{aligned}$$

APPENDIX E PROOF OF THEOREM 3

Proof: First we will show that the PRSreco algorithm succeeds in identifying the top s terms of the signal. We will then derive the error guarantee.

Let us begin with signal X exactly s -sparse. For simplicity let us assume that $X_i, i = 1, \dots, s$ are the nonzeros. There exists a $\beta < 1$ such that $|X_i|^2 \geq \beta \|X\|^2/s$. Let $0 < 2\alpha \leq \beta$. For $i = 1, \dots, s$, using the Chebyshev inequality we have

$$\mathbb{P}_r(|\tilde{X}_i^{(1)} - X_i|^2 \geq \frac{\alpha \|X\|^2}{s}) \leq \frac{\text{Var}(\tilde{X}_i^{(1)})}{\frac{\alpha \|X\|^2}{s}} \leq \frac{\frac{\epsilon^2 \|X\|^2}{s}}{\frac{\alpha \|X\|^2}{s}} = \frac{\epsilon^2}{\alpha}$$

[using (3) from Lemma (2)]. Hence

$$\mathbb{P}_r(\tilde{X}_i^{(1)} \text{ good}) = \mathbb{P}_r(|\tilde{X}_i^{(1)} - X_i|^2 \leq \frac{\alpha \|X\|^2}{s}) \geq 1 - \frac{\epsilon^2}{\alpha}.$$

Let $|X_{\min}|$ be the smallest nonzero in X . Again using Chebyshev inequality, for each of $X_i, i > s$ we have

$$\begin{aligned} \mathbb{P}_r(|\tilde{X}_i^{(1)}|^2 \leq |X_{\min}| - \frac{\alpha \|X\|^2}{s}) \\ \geq \mathbb{P}_r(|\tilde{X}_i^{(1)}|^2 \leq (\beta - \alpha) \frac{\|X\|^2}{s}) \\ \geq 1 - \frac{\epsilon^2}{\beta - \alpha} = 1 - \frac{\epsilon^2}{\alpha} \end{aligned}$$

(since $2\alpha \leq \beta$). Now, define Bernoulli random variables z_i as indicators of failure of the i th coefficient estimator. That is

$$\mathbb{P}_r(z_i = 0) = 1 - \frac{\epsilon^2}{\alpha} = 1 - \mathbb{P}_r(z_i = 1)$$

for all $i = 1, \dots, N$. Let $Z_1 = \sum_{i=1}^s z_i$ and $Z_2 = \sum_{i=s+1}^N z_i$. We have

$$\mathbb{P}_r\left(Z_1 > \frac{s}{4}\right) \leq \frac{\mathbb{E}(Z_1)}{\frac{s}{4}} \leq \frac{\frac{s\epsilon^2}{4}}{\frac{s}{4}} = \frac{4\epsilon^2}{\alpha}.$$

Hence

$$\begin{aligned} \mathbb{P}_r\left(\text{No. of good estimators among } \tilde{X}_1^{(1)}, \tilde{X}_2^{(1)}, \dots, \tilde{X}_s^{(1)} \geq \frac{3s}{4}\right) &\geq 1 - \frac{4\epsilon^2}{\alpha}. \end{aligned}$$

Note that the factor $1/4$ is chosen as an example to simplify the presentation of the proof. Now let us move on to the *second* iteration of the algorithm. More than $3s/4$ estimators which were good in the first iteration are still good in the second iteration. This is because the estimator $\tilde{X}_i^{(2)}$ depends on the same random correlations (between B_i and other columns of B) as the estimator $\tilde{X}_i^{(1)}$ from the first iteration. Put the current approximation $\tilde{X} = [\tilde{X}^{(1)}]_{(s)}$ as defined in the PRSreco algorithm (see Table I). Now for those coefficients whose estimators were not good in the first iteration we have

$$\mathbb{P}_r\left(|\tilde{X}_i^{(2)} - X_i|^2 \geq \frac{\alpha \|X - \tilde{X}\|^2}{s}\right) \leq \frac{\frac{\epsilon^2 \|X - \tilde{X}\|^2}{s}}{\frac{\alpha \|X - \tilde{X}\|^2}{s}} = \frac{\epsilon^2}{\alpha}$$

like before, using the (3) from Lemma (2) applied to $X - \tilde{X}$. Now define a new $Z_1^{(2)}$ for these estimators. Note that $\mathbb{E}(Z_1^{(2)}) \leq s\epsilon^2/4\alpha$ (since there are less than $s/4$ terms in the definition of $Z_1^{(2)}$). Now as before we have

$$\mathbb{P}_r\left(Z_1^{(2)} > \frac{s}{4^2}\right) \leq \frac{4\epsilon^2}{\alpha}.$$

Hence by the end of second iteration number of good estimators among the $i = 1, \dots, s$ is $\geq 3s/4 + (3/4)s/4$ with a net probability $\geq (1 - 4\epsilon^2/\alpha)^2$. Going on this way at k th iteration, number of good estimators $\geq (1 - (1/4)^k)s$, with probability $\geq (1 - 4\epsilon^2/\alpha)^k$. Similar statements can be obtained about Z_2 , i.e., about the estimators with $i > s$. Hence, after sufficient number of iterations (say I), all the estimators are good which implies that all the nonzero terms will be identified by the algorithm with

$$\mathbb{P}_r(\text{Success}) \geq \left(1 - \frac{4\epsilon^2}{\alpha}\right)^{2I} \approx \left(1 - \frac{8I\epsilon^2}{\alpha}\right) = 1 - O(\epsilon^2)$$

after absorbing some constants along with number of iterations I into the number of measurements. If I is the sufficient number of iterations at which all estimators are good, then $(1/4)^I N < 1 \Rightarrow I = 0.5 \log N = o(\log N)$. Hence, an increase in number of measurements by a factor of $\log N$ is required. Note that the above is a gross lower bound for the success probability. In reality since all the estimators are highly dependent, the probability that they will be good together is higher than the product of the individual success probabilities, which is the gross lower bound produced by the above theory.

Now let the signal X be s -compressible (hence not exactly s -sparse). We start with $K = O(s/\epsilon^2)$ as before. Again for simplicity let the first s elements of X be the top s terms. For $i > s$ we assume that $|X_i|^2 \leq \gamma \|X\|^2/s$ for some $\gamma < 1$. Let X_{\min}^h be the smallest coefficient in the head ($i = 1, \dots, s$) of X . Similarly let X_{\max}^t be the largest coefficient in the tail ($i > s$) of X . All the above arguments hold again except that for $i > s$ the probabilities will involve γ in the following manner. For example in the first iteration

$$\begin{aligned} \Pr \left(|\tilde{X}_i - X_i|^2 \leq |X_{\min}^h| - \frac{(\alpha) \|X\|^2}{s} - |X_{\max}^t| \right) \\ \geq \Pr \left(|\tilde{X}_i - X_i|^2 \leq \left(\frac{1}{\beta} - \alpha - \gamma \right) \frac{\|X\|^2}{s} \right) \\ \geq 1 - \frac{\beta \epsilon^2}{1 - (\alpha + \gamma)\beta} = 1 - \frac{\epsilon^2}{\alpha} \end{aligned}$$

(assuming $0 < 2\alpha \leq \beta - \gamma$). Repeating the arguments from above we show that the algorithm succeeds in identifying the top s -terms.

Now let us prove the error guarantee. Let us assume that $\xi = 0$ for the moment. Let us say the algorithm correctly identifies the position of top s terms in I iterations. For any $k > I$, at iteration $k + 1$, $|\tilde{X}_i^{(k+1)} - X_i|^2 < \alpha \|X - \tilde{X}^{(k)}\|^2/s$ for $i = 1, \dots, s$. Summing up the s inequalities, we get

$$\|\tilde{X}^{(k+1)} - X_{(s)}\|^2 \leq \alpha \|X - \tilde{X}^{(k)}\|^2$$

where $X_{(s)}$ is the best s -term approximation to X . Now

$$\begin{aligned} \|X - \tilde{X}^{(k+1)}\|^2 &\leq \|X - X_{(s)}\|^2 + \|X_{(s)} - \tilde{X}^{(k+1)}\|^2 \\ &\leq \|X - X_{(s)}\|^2 + \alpha \|X - \tilde{X}^{(k)}\|^2. \end{aligned}$$

This implies

$$\begin{aligned} \|X - \tilde{X}^{(k+1)}\|^2 \\ \leq \frac{1 - \alpha^{k-I}}{1 - \alpha} \|X - X_{(s)}\|^2 + \alpha^{k-I} \|X - \tilde{X}^{(k-I)}\|^2. \end{aligned}$$

For k large enough we have

$$\|X - \tilde{X}\|^2 \leq \frac{1}{1 - \alpha} \|X - X_{(s)}\|^2.$$

This is consistent with (4).

Now let $\xi = Bn$ for some vector n . we have $y = B(X + n)$. Following the arguments as before, we have

$$\|X + n - \tilde{X}\|^2 \leq \frac{\|X + n - (X + n)_{(s)}\|^2}{1 - \alpha} \leq \frac{\|X + n - X_{(s)}\|^2}{1 - \alpha}$$

(since $(X + n)_{(s)}$ is the best s -term approximation to $X + n$). Now

$$\begin{aligned} \|X - \tilde{X}\|^2 \\ \leq \|X + n - \tilde{X}\|^2 + \|n\|^2 \leq \frac{\|X + n - X_{(s)}\|^2}{1 - \alpha} + \|n\|^2 \\ \leq \frac{\|X - X_{(s)}\|^2}{1 - \alpha} + \frac{(2 - \alpha)\|n\|^2}{1 - \alpha}. \end{aligned}$$

We will have (4) by putting $\|n\|_2 \leq c\|\xi\|_2$. This is true for some $c(B) < 1/\sigma_{\min}(B)$ where the denominator is the smallest singular value of B . ■

REFERENCES

- [1] J. A. Michaelson, J. E. Ramstad, D. T. Wisland, and O. Sorasen, *Low-Power Sensor Interfacing and MEMS for Wireless Sensor Networks*, 1st ed. New York: InTech, 2011.
- [2] F. Shahrokhi, K. Abdelhalim, D. Serletis, P. L. Carlen, and R. Genov, "The 128-channel fully differential digital integrated neural recording and stimulation interface," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 149–161, Jun. 2010.
- [3] E. Delagnes, D. Breton, F. Lugiez, and R. Rahmanifard, "A low power multi-channel single ramp ADC with up to 3.2 GHz virtual clock," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 5, pp. 1735–1742, Oct. 2007.
- [4] T. Fusayasu, "A fast integrating ADC using precise time-to-digital conversion," in *IEEE Nucl. Sci. Symp. Conf. Rec.*, 2007, vol. 1, pp. 302–304.
- [5] A. H. Reeves, "Electrical signaling system," U.S. Patent 2 272 070, Feb. 3, 1942.
- [6] J. Mark and T. Todd, "A non-uniform sampling approach to data compression," *IEEE Trans. Commun.*, vol. 29, no. 1, pp. 24–32, Jan. 1981.
- [7] M. Z. Straayer and M. H. Perrott, "A 10-bit 20 MHz 38 mW 950 MHz CT Sigma Delta ADC with a 5-bit noise-shaping VCO-based quantizer and DEM circuit in 0.13 μ CMOS," in *VLSI Symp. Dig. Tech. Papers*, Jun. 2007, pp. 246–247.
- [8] A. A. Lazar and L. T. Toth, "Perfect recovery and sensitivity analysis of time encoded bandlimited signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 2060–2073, Oct. 2004.
- [9] A. A. Lazar and E. A. Pnevmatikakis, "Reconstruction of sensory stimuli encoded with integrate-and-fire neurons with random thresholds," *EURASIP J. Adv. Signal Process.*, pp. 1–15, 2009.
- [10] M. Kurchuk and Y. Tsvividis, "Signal-Dependent Variable-Resolution clockless A/D conversion with application to CT-DSP," *IEEE Trans. Circuits Syst.*, vol. 57, no. 5, pp. 982–991, May 2010.
- [11] B. Schell and Y. Tsvividis, "A continuous-time ADC/DSP/DAC system with no clock and with activity-dependent power dissipation," *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp. 2742–2481, Nov. 2008.
- [12] Y. Li, "A 0.5 v signal-specific continuous-time level-crossing ADC with charge sharing," in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2011, pp. 381–384.
- [13] S. Naraghi, M. Courcy, and M. P. Flynn, "A 9 b 14 μ W 0.06 mm² PPM ADC in 90 nm digital CMOS," in *IEEE Int. Solid-State Circuits Conf.*, 2009, vol. 54, pp. 168–169.
- [14] B. Murmann, "A/D converter trends: Power dissipation, scaling and digitally assisted architectures," in *IEEE Custom Integr. Circuits Conf. (CICC)*, 2008, pp. 105–112.
- [15] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2009, pp. 1959–1962.
- [16] C. Luo and J. H. McClellan, "Compressive sampling with a successive approximation ADC architecture," in *Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2011, pp. 3920–3923.
- [17] S. R. Becker, "Practical compressed sensing: Modern data acquisition and signal processing," Ph.D. dissertation, California Inst. Technol, Pasadena, 2011.
- [18] M. Mishali, Y. C. Eldar, and A. J. Elron, "Xampling: Signal acquisition and processing in union of subspaces," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4719–4734, Oct. 2011.
- [19] F. Chen, A. P. Chandrakasan, and V. Stojanovic, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," in *Custom Integr. Circuits Conf. (CICC)*, 2010, pp. 1–4.

- [20] P. K. Yenduri, A. C. Gilbert, M. P. Flynn, and S. Naraghi, "Rand PPM: A low power compressive sampling analog to digital converter," in *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 5980–5983.
- [21] M. Fornasier and H. Rauhut, *Compressive Sensing*. New York: Springer, 2011.
- [22] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [23] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [24] M. Rudelson and R. Vershynin, "Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements," in *40th Annu. Conf. Inf. Sci. and Syst. (CISS)*, Princeton, NJ, Mar. 2006, pp. 207–212.
- [25] A. E. Litvak, A. Pajor, M. Rudelson, and N. Tomczak-Jaegermann, "Smallest singular value of random matrices and geometry of random polytopes," *Adv. Math.*, vol. 195, no. 2, pp. 491–523, 2005.
- [26] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
- [27] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmonic Anal.*, vol. 26, pp. 301–321, Apr. 2008.
- [28] T. Blumensath and M. E. Davis, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, pp. 629–654, Sep. 2008.
- [29] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, "Improved time bounds for near-optimal sparse Fourier representation via sampling," presented at the SPIE Wavelets XI, San Diego, CA, 2005.
- [30] M. Iwen, "A deterministic sub-linear time sparse Fourier algorithm via non-adaptive compressed sensing methods," in *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2008, pp. 20–29.
- [31] P. K. Yenduri and A. C. Gilbert, "Continuous fast Fourier sampling," presented at the Conf. Sampling Theory Applicat. (SAMPTA), Marseille, France, 2009.
- [32] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Simple and practical algorithm for sparse fourier transform," in *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2012, pp. 1183–1194.
- [33] S. Kunis and H. Rauhut, "Random sampling of sparse trigonometric polynomials, II. Orthogonal matching pursuit versus basis pursuit," *Found. Comput. Math.*, vol. 8, no. 6, pp. 737–763, Nov. 2008.
- [34] S. Szczepanski, P. Dudek, and J. V. Hatfield, "A high-resolution CMOS time-to-digital converter utilizing a vernier delay line," *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, Feb. 2000.
- [35] F. Marvasti, *Nonuniform Sampling Theory and Practice*. Norwell, MA: Kluwer, 1990.
- [36] F. Marvasti, M. Analoui, and M. Gamshadzi, "Recovery of signals from nonuniform samples using iterative methods," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 872–878, Apr. 1991.
- [37] R. G. Baraniuk and P. Steeghs, "Compressive radar imaging," in *IEEE Radar Conf.*, Apr. 2007, pp. 128–133.
- [38] F. J. Herrmann, D. Wang, G. Hennenfent, and P. Moghaddam, "Curvelet-based seismic data processing: A multiscale and nonlinear approach," *Geophysics*, vol. 73, no. 1, pp. A1–A5, Feb. 2008.
- [39] J. Berent, P. Dragotti, and T. Blu, "Sampling piecewise sinusoidal signals with finite rate of innovation methods," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 613–625, Feb. 2010.
- [40] A. Björck, *Numerical Methods for Least Squares Problems*, 17 ed. Philadelphia, PA: SIAM, 1996.
- [41] G. Steidl, "A note on fast Fourier transforms for nonequispaced grids," *Advances Comput. Math.*, vol. 9, pp. 337–352, Nov. 1998.

Praveen K. Yenduri received the B.Tech. degree in electronics and communications from Indian Institute of Technology (IIT), Guwahati, India, in 2007. He graduated with the accolade of being the institute topper and was the recipient of the prestigious President of India Gold medal. He received the M.S.E degree in electrical engineering and the Ph.D. degree in signal processing, from the University of Michigan, Ann Arbor, in 2008 and 2012, respectively.

During summer 2006, he worked with Medical Imaging Group at ETH Zurich as a research intern. His interests include compressive sampling and signal reconstruction, structured random sampling, wideband cognitive radios, and image processing.

Dr. Yenduri has received several awards, including the Electronic, Electrical, and Computer Engineering Department fellowship (2007–2008), IIT merit scholarship (2003–2007), and the Best Graduate Student Instructor (GSI) award (2011–2012).

Aaron Z. Rocca received the B.S.E. and M.S.E. degrees in electrical engineering and physics from the University of Michigan, Ann Arbor, in 2010 and 2011, respectively.

He is currently with Baker-Calling, Inc., a University of Michigan microphone start up. His design interests include data converters, RF communication, and low-noise audio amplifier circuits.

Aswin S. Rao was born in Trivandrum, India, in 1985. He received the B.Tech. degree from College of Engineering, Trivandrum, India, in 2006, and the M.S. degree in electrical engineering specializing in analog circuits from University of Michigan, Ann Arbor, in 2011. His research interests during this time included digital PLL design and compressive sensing.

He was with Analog Devices Inc., Bangalore, India from 2006 to 2010 as a Digital Design Engineer in the Digital Signal Processors IC Design Group. He is currently with Texas Instruments Inc., Dallas, TX, as an Analog Design Engineer in the Power Management Group.

Shahrazad Naraghi received the B.Sc. degree from Sharif University of Technology, Tehran, Iran, in 2001, the M.Sc. degree from University of Waterloo, Waterloo, ON, Canada, in 2004, and the Ph.D. degree from University of Michigan, Ann Arbor, in 2009, all in electrical engineering.

She is currently an Analog Circuit Design Engineer at Tektronix Component Solutions, Santa Clara, CA. Previously, she held IC design positions at Cirrus Logic, Austin, TX, and IBM T. J. Watson Research Center, Yorktown Heights, NY.

Dr. Naraghi has been an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS since 2010.

Michael P. Flynn received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1995.

From 1988 to 1991, he was with the National Microelectronics Research Centre in Cork, Ireland. He was with National Semiconductor, Santa Clara, CA, from 1993 to 1995. From 1995 to 1997, he was a Member of Technical Staff with Texas Instruments, Dallas. During the four-year period from 1997 to 2001, he was with Parthus Technologies, Cork, Ireland. He joined the University of Michigan, Ann Arbor, in 2001, and is currently Professor. His technical interests are in RF circuits, data conversion, serial transceivers, and biomedical systems.

Dr. Flynn is a 2008 Guggenheim Fellow. He received the 2011 Education Excellence Award and the 2010 College of Engineering Ted Kennedy Family Team Excellence Award from the College from Engineering at the University of Michigan. He received the 2005–2006 Outstanding Achievement Award from the Department of Electrical Engineering and Computer Science at the University of Michigan. He received the NSF Early Career Award in 2004. He received the 1992–1993 IEEE Solid-State Circuits Pre-doctoral Fellowship. He is an Associate Editor of the IEEE JOURNAL OF SOLID-STATE CIRCUITS and serves on the Technical Program Committees of the International Solid State Circuits Conference (ISSCC) and the Symposium on VLSI Circuits. He formerly served on the program committee of the Asian Solid-State Circuits Conference was Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS from 2002 to 2004.

Anna C. Gilbert received the S.B. degree from the University of Chicago, Chicago, IL, and the Ph.D. degree from Princeton University, Princeton, NJ, both in mathematics.

In 1997, she was a postdoctoral fellow at Yale University and AT&T Labs-Research. From 1998 to 2004, she was a member of technical staff at AT&T Labs-Research, Florham Park, NJ. Since then she has been with the Department of Mathematics, University of Michigan, Ann Arbor, where she is now a Professor. Her research interests include analysis, probability, networking, and algorithms. She is especially interested in randomized algorithms with applications to harmonic analysis, signal and image processing, networking, and massive datasets.

Dr. Gilbert has received several awards, including a Sloan Research Fellowship (2006), an NSF CAREER award (2006), the National Academy of Sciences Award for Initiatives in Research (2008), the Association of Computing Machinery (ACM) Douglas Engelbart Best Paper award (2008), and the EURASIP Signal Processing Best Paper award (2010).