

GHz-Wide Sensing and Decoding Using the Sparse Fourier Transform

Haitham Hassanieh Lixin Shi Omid Abari Ezzeldin Hamed Dina Katabi

Massachusetts Institute of Technology

{haitham, lixin, abari, ezz, dina}@csail.mit.edu

Abstract— We present BigBand, a technology that can capture GHz of spectrum in realtime without sampling the signal at GS/s —i.e., without high speed ADCs. Further, it is simple and can be implemented on commodity low-power radios. Our approach builds on recent advances in the area of sparse Fourier transforms, which show that it is possible to reconstruct a sparse signal without sampling it at the Nyquist rate. To demonstrate our design, we implement it using 3 software radios, each sampling the spectrum at 50 MS/s, producing a device that captures 0.9 GHz — *i.e.*, 6× larger digital bandwidth than the three software radios combined. Finally, an extension of BigBand can perform GHz spectrum sensing even in scenarios where the spectrum is not sparse.

1 INTRODUCTION

The rising popularity of wireless communication and the potential of a spectrum shortage have motivated the FCC to take steps towards releasing multiple bands for dynamic spectrum sharing [1]. The government's interest in re-purposing the spectrum for sharing is motivated by the fact that the actual utilization of the spectrum is sparse in practice. For instance, Fig. 1 from the Microsoft Spectrum Observatory [2] shows that, even in urban areas, large swaths of the spectrum remain underutilized. To use the spectrum more efficiently, last year, the President's Council of Advisors on Science and Technology (PCAST) [3] has advocated dynamic sharing of much of the currently under-utilized spectrum, creating GHz-wide spectrum superhighways "that can be shared by many different types of wireless services, just as vehicles share a superhighway by moving from one lane to another."

Motivated by this vision, this paper presents BigBand, a technology that enables realtime GHz-wide spectrum sensing and reception using low-power radios, similar to those in WiFi devices. Making GHz-wide sensing (*i.e.* the ability to detect occupancy) and reception (*i.e.* the ability to decode) available on commodity radios enables new applications:

- In particular, realtime GHz sensing enables highly dynamic spectrum access, where secondary users can detect short sub-millisecond spectrum vacancies and leverage them, thereby increasing the overall spectrum efficiency [4].
- Further, a cheap low-power GHz spectrum sensing technology enables the government and the industry to deploy thousands or such sensors in a metropolitan area for large-scale realtime spectrum monitoring. This will enable a better understanding of spectrum utilization, identification and localization of breaches of spectrum policy, and a more-informed planning of spectrum allocation.
- Beyond sensing, the ability to decode signals in a GHz-wide band enables a single radio to receive concurrent transmissions from diverse parts of the spectrum. This would

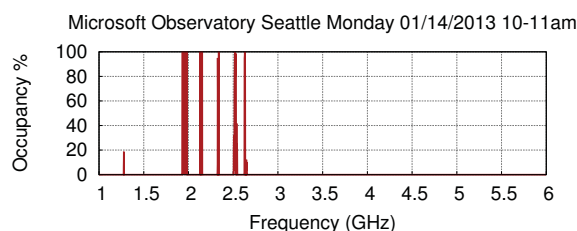


Fig. 1—Spectrum Occupancy: The figure shows the average spectrum occupancy at the Microsoft spectrum observatory in Seattle on Monday January 14, 2013 during the hour between 10 am and 11 am. The figure shows that between 1 GHz and 6 GHz, the spectrum is sparsely occupied.

enable future cell phones to use one radio to concurrently receive Bluetooth at 2.4 GHz, GSM at 1.9 GHz, and CDMA at 1.7 GHz.

Realtime GHz signal acquisition, however, is challenging. For example, existing methods for spectrum sensing, like those used in the Microsoft spectrum observatory [2], do not work in realtime. They rely on sequential hopping from one channel to the next, acquiring only tens of MHz at a time [5], [6]. As a result, each band is monitored only occasionally, making it easy to miss short lived signals (*e.g.*, radar).

The key difficulty in capturing GHz of bandwidth in realtime stems from the need for high-speed analog-to-digital converters (ADCs), which are costly, power hungry, and have a low bit resolution [7], [8]. Compare typical low-speed ADCs used in WiFi or cellular phones with the very high speed ADCs needed to capture GHz of bandwidth. A 100 MS/s ADC, like in Wi-Fi receivers, costs a few dollars, consumes a few milli Watts, and has a 12 to 16-bit resolution [8], [9], [10]. In contrast, a high speed ADC that can take multiple giga-samples per second may cost hundreds of dollars, consume multiple orders of magnitude more power, and have only 6 to 8-bits resolution [7], [8], [9].

In this paper, we explore how one can achieve the best of both worlds. Specifically, we would like to capture GHz of spectrum but using only a few ADCs that each samples the signal at tens of MS/s.

We introduce BigBand, a technology that can acquire GHz of signal using a few (3 or 4) low-speed ADCs. BigBand can do more than spectrum sensing – the action of detecting occupied bands. It can also decode the signal (*i.e.*, obtain the I and Q components). To achieve its goal, BigBand builds on advances in the area of sparse Fourier transform [11], [12], [13], which permit signals whose frequency domain representation is sparse to be recovered using only a small subset of their samples — *i.e.*, we can recover GHz of spectrum without sampling it at the Nyquist rate.

Some past work has proposed using compressive sensing

to acquire GHz signals at sub-Nyquist rate [14], [15], [16], [17]. BigBand builds on this work but differs from it substantially. Approaches based on compressive sensing require random sampling of the signal which cannot be done simply by using standard low-speed ADCs. It needs analog mixing at Nyquist rates [14], [16] and expensive processing to recover the original signal. Such a design is quite complex and could end up consuming as much power as an ADC that samples at the Nyquist rate [18], [19]. Like the compressive-sensing approaches, BigBand can acquire a wideband signal without sampling it at the Nyquist rate. Unlike compressive sensing, however, BigBand does not need analog mixing or random sampling and can work using commodity radio and standard low-speed ADCs. Further, it computes the Fourier transform of a sparse signal faster than the FFT, reducing baseband processing.

We have built a working prototype of BigBand using USRP software radios. Our prototype uses three USRPs, each of which can capture 50 MHz bandwidth to produce a device that captures 0.9 GHz –i.e., $6\times$ larger bandwidth than the digital bandwidth of the three USRPs combined. We have used our prototype to sense the spectrum between 2 GHz and 2.9 GHz, a 0.9-GHz stretch used by diverse technologies [2]. Our outdoor measurements reveal that, in our metropolitan area,¹ the above band has an occupancy of 2–5%. These results were verified using a spectrum analyzer in sync with similar measurements conducted at other locations [2]. We further use our prototype to decode 30 transmitters that are simultaneously frequency hopping in a 0.9 GHz band, hence demonstrating that BigBand decodes the signals, not only senses their power.

Finally, we have extended BigBand to perform spectrum sensing (not decoding) even when the spectrum utilization is not sparse. To do so, we leverage the idea that even if the spectrum itself is densely occupied, only a small fraction of the spectrum is likely to change its occupancy over short intervals of a few milliseconds. We build on this basic idea to sense densely occupied spectrum using sub-Nyquist sampling. We also evaluate our design empirically showing that it can detect frequency bands that change occupancy even when the spectrum is 95% occupied.

2 RELATED WORK

This paper builds on recent theoretical advances in sparse Fourier sampling [11], [12], [13], [20], [21]. In contrast to past work however, this paper focuses on the practical problem of realtime low-power GHz-wide spectrum acquisition, and presents the first practical system that adapts the sparse FFT algorithms to address this application. It also implements its design and empirically evaluates it, demonstrating that it addresses ADC speed, wireless channels, and radio related issues. The paper is also the first to adapt the sparse FFT for spectrum sensing in scenarios with a dense spectrum occupancy.

Our work is related to signal acquisition via digital and analog compressive sensing [14], [15], [16], [17], [22], [23], [24]. However, compressive sensing needs random sampling and analog mixing at Nyquist rates [14], [16], [24]. These approaches cannot be built using commodity radios and ADCs with regular sampling; they require a custom design and could

end up consuming as much power as an ADC that samples at the Nyquist rate [18], [19].

Our work is also related to theoretical work in signal processing on co-prime sampling [25], [26], [27]. In [25], [26], co-prime sampling patterns are utilized to sample sparse spectrum. These methods however require k ADCs with co-prime sampling patterns, where k is the number of occupied frequencies. In contrast, the analysis of our sparse FFT algorithm shows that our approach requires only a constant small number of ADCs [13]. Our approach is further implemented and shown to work in practice. In [27], co-prime sampling is used to sample linear antenna arrays [27]. This work however assumes the presence of a second dimension where signals can be fully sampled and cross-correlated and hence cannot be used for spectrum acquisition.

Also relevant to our work is the theoretical work on using multicoset sampling to capture the signals in a wideband sparse spectrum with a small number of low speed ADCs [28], [29]. However, in order to recover the original signals from the samples, these techniques require prior knowledge of the locations of occupied frequencies in the spectrum and hence are not useful for spectrum sensing. In contrast, our approach recovers both the locations of the occupied frequencies and the signals in these frequencies and thus can be used for both spectrum sensing and decoding.

Some proposals for test equipment reconstruct wideband periodic signals by undersampling [30], [31]. These approaches however assume that the signal is periodic –i.e., the same signal keeps repeating for very long time – which allows them to take one sample during each period until all samples are recovered and rearranged in the proper order. Though this requires one low speed ADC, it is only applicable to test equipment where the same signal is repeatedly transmitted [30].

Finally, there is also significant literature about spectrum sensing. Most of this work focuses on narrowband sensing [32], [33], [34]. It includes techniques for detecting the signal's energy [33], its waveform [32], its cyclostationarity [35], or its power variation [34]. In contrast, we focus on wideband spectrum sensing, an area that is significantly less explored. A recent system called QuickSense [36] senses a wideband signal using a hierarchy of analog filters and energy detectors. BigBand differs from QuickSense in that it can recover the signal (obtain the I and Q components) as opposed to only detecting spectrum occupancy. Second, for highly utilized spectrum (i.e. not sparse), the approach in [36] reduces to sequentially scanning the spectrum whereas BigBand's extension for the non-sparse case provides a fast sensing mechanism.

3 BIGBAND

BigBand is a receiver that can recover a sparse signal with sub-Nyquist sampling using low-power commodity radios. Thus, BigBand can do more than spectrum sensing – the action of detecting occupied bands. BigBand provides the details of the signals in those bands (I's and Q's of wireless symbols), which enables decoding those signals.

BigBand adapts the sparse FFT algorithm for spectrum acquisition using low speed ADCs. We use \mathbf{x} and $\hat{\mathbf{x}}$ to denote a time signal and its Fourier transform respectively. We also use the terms: the *value* of a frequency and its *position* in the spectrum to distinguish $\hat{\mathbf{x}}_f$ and f . BigBand discovers the

1. MIT campus, Cambridge MA, USA.

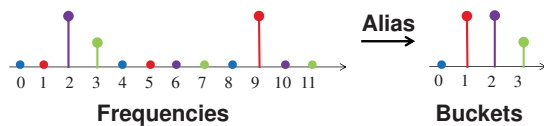


Fig. 2—Bucketization using aliasing filter: Sub-sampling a signal by $3\times$ in the time domain, results in the spectrum aliasing. Specifically, the 12 frequency will alias into 4 buckets. Frequencies that are equally spaced by 4 (shown with the same color) end up in the same bucket.

occupied frequency positions f and estimates their values $\hat{\mathbf{x}}_f$. Once $\hat{\mathbf{x}}$ is computed, it can recover the time signal \mathbf{x} and decode the wireless symbols. BigBand's design has three components: frequency *bucketization*, *estimation*, and *collision resolution*. Below we explain these components. (Further details of BigBand can be found in our technical report [37] and the complete analysis of the algorithm and proofs of correctness can be found in [13].)

3.1 STEP 1: Frequency Bucketization

BigBand starts by hashing the frequencies in the spectrum into buckets. Since the spectrum is sparsely occupied, many buckets will be empty and can be simply discarded. BigBand then focuses on the non-empty buckets, and computes the values of the frequencies in those buckets in what we call the estimation step.

So how can we hash frequencies into buckets? Recall the following basic property of the Fourier transform: *sub-sampling in the time domain causes aliasing in the frequency domain*. Formally, let \mathbf{x} be a time signal of bandwidth BW , and $\hat{\mathbf{x}}$ its frequency representation. Let \mathbf{b} be a sub-sampled version of \mathbf{x} , i.e., $\mathbf{b}_i = \mathbf{x}_{i/p}$ where p is the sub-sampling factor. Then, $\hat{\mathbf{b}}$, the FFT of \mathbf{b} is an aliased version of $\hat{\mathbf{x}}$, i.e.:

$$\hat{\mathbf{b}}_i = \sum_{m=0}^{p-1} \hat{\mathbf{x}}_{i+m(BW/p)}. \quad (1)$$

Thus, an aliasing filter is a form of bucketization in which frequencies equally spaced by an interval BW/p hash to the same bucket, i.e., frequency f hash to bucket $i = f \bmod BW/p$, as shown in Fig. 2. Further, the value in each bucket is the sum of the values of only the frequencies that hash to the bucket as shown in Eq. 1. Most importantly, aliasing is naturally done by sampling the signal using a low-speed ADC slower than the Nyquist rate.

Now that we hashed the frequencies into buckets, we can leverage the fact that the spectrum of interest is sparse and hence most buckets have noise and no signal. BigBand compares the energy (i.e., the magnitude square) of a bucket with the receiver's noise level and considers all buckets whose energy is below a threshold to be empty. It then focuses on the occupied buckets and ignores empty buckets.

3.2 STEP 2: Frequency Estimation

Next, for each of the occupied buckets we want to identify which frequencies created the energy in these buckets, and what are the values of these frequencies. If we can do that, we then have recovered a complete representation of the frequencies with non-zero signal values, i.e., we acquired the full signal in the Fourier domain.

Recall that our spectrum is sparse; thus, as mentioned earlier, when hashing frequencies into buckets many buckets are likely to be empty. Even for the occupied buckets, the sparsity of the spectrum means that many of these buckets will likely have a single non-zero frequency hashing into them, and only a small number will have a collision of multiple non-zero (or occupied) frequencies. In the next section, we present a mechanism to detect whether a bucket has a collision and resolve such collisions. In this section, we focus on buckets with a single non-zero frequency and estimate the value and the position of this non-zero frequency, i.e., $\hat{\mathbf{x}}_f$ and the corresponding f .

In the absence of a collision, the value of the occupied frequency is the value of the bucket. Said differently, the value of a bucket after aliasing, $\hat{\mathbf{b}}_i$ is a good estimate of the value of the occupied frequency $\hat{\mathbf{x}}_f$ in that bucket, since all other frequencies in the bucket have zero signal value (only noise).

Although we can easily find the value of the non-zero frequency in a bucket, we still do not know its frequency position f , since aliasing mapped multiple frequencies to the same bucket. To compute f , we leverage the *phase-rotation property* of the Fourier transform, which states that a shift in time domain translates into phase rotation in the frequency domain [38]. Specifically, we perform the process of bucketization again, after shifting the input signal by τ . Since a shift in time translates into phase rotation in the frequency domain, the value of the bucket changes from $\hat{\mathbf{b}}_i = \hat{\mathbf{x}}_f$ to $\hat{\mathbf{b}}_i^{(\tau)} = \hat{\mathbf{x}}_f \cdot e^{2\pi j f \cdot \tau}$. Hence, using the change in the phase of the bucket, we can estimate our frequency of interest and we can do this for all buckets that do not have collisions.

Two points are worth noting:

- First, recall that the phase wraps around every 2π . Hence, the value of τ has to be small to avoid the phase wrapping around for large values of f . In particular, τ should be on the order of $1/BW$ where BW is the bandwidth of interest. For example, to acquire one GHz of spectrum, τ should be on the order of a nanosecond.²
- Second, to sample the signal with a τ shift, we need a second low-speed ADC that has the same sampling rate as the ADC in the bucketization step but whose samples are delayed by τ . This can be achieved by connecting a single antenna to two ADCs using different delay lines (which is what we do in our implementation). Alternatively, one can use different delay lines to connect the clocks to the two ADCs.

3.3 STEP 3: Collision Detection and Resolution

We still need to address two questions: how do we distinguish the buckets that have a single non-zero frequency from those that have a collision? and in the case of a collision, how do we resolve the colliding frequencies?

3.3.1 Collision Detection

Again we use the *phase rotation* property of the Fourier transform to determine if a collision has occurred. Specifically, if the bucket contains a single non-zero frequency, i.e., no collision, then performing the bucketization with a time shift τ causes only a phase rotation of the value in the bucket but

² In fact, one can prove a looser version of this constraint where large τ are fine. Formally, for τ larger than $1/BW$, the FFT window size must be a non-integer multiple of τ .

the magnitude of the bucket does not change *-i.e.*, with or without the time shift, $\|\hat{\mathbf{b}}_i\| = \|\hat{\mathbf{b}}_i^{(\tau)}\| = \|\hat{\mathbf{x}}_f\|$. In contrast, consider the case where there is a collision between, say, two frequencies f and f' . Then the value of the bucket without a time-shift is $\hat{\mathbf{b}}_i = \hat{\mathbf{x}}_f + \hat{\mathbf{x}}_{f'}$ while its value with a time-shift of τ is $\hat{\mathbf{b}}_i^{(\tau)} = \hat{\mathbf{x}}_f \cdot e^{2\pi j f \tau} + \hat{\mathbf{x}}_{f'} \cdot e^{2\pi j f' \tau}$. Since the colliding frequencies rotate by different phases, the overall magnitude of the bucket will change. Thus, we can determine whether there is a collision or not by comparing the magnitudes of the buckets with and without the time-shift.³

3.3.2 Collision Resolution

To reconstruct the full spectrum, we need to resolve the collisions *-i.e.*, for each non-zero frequency in a collision we need to estimate its value $\hat{\mathbf{x}}_f$ and position f . We present two approaches for resolving collisions which may also be combined in case the spectrum is less sparse.

A. Resolving Collisions with Co-prime Aliasing Filters

One approach to resolve collisions is to bucketize the spectrum multiple times using aliasing filters with co-prime sampling rates. Co-prime aliasing filters guarantee (by the Chinese remainder theorem) that any two frequencies that collide in one bucketization will not collide in the other bucketizations. To better understand this point, consider the example in Fig. 3. The first time we bucketize, we use an aliasing filter that sub-samples the time signal by a factor of 3. In this case, the two frequencies labeled in red and blue collide in a bucket whereas the frequency labeled in green does not collide, as shown in the figure. The second time we bucketize, we use an aliasing filter that sub-samples by 4. This time the blue and green frequencies collide whereas the red frequency does not collide. Now we can resolve collisions by iterating between the two bucketizations. For example, we can estimate the green frequency from the first bucketization, where it does not collide. We subtract the green frequency from the colliding bucket in the second bucketization to obtain the blue frequency. We then go back to the first bucketization and subtract the blue frequency from the bucket where it collides to obtain the red frequency.

Thus, by using co-prime aliasing filters to bucketize and iterating between the bucketizations *-i.e.*, estimating frequencies from buckets where they do not collide and subtracting them from buckets where they do collide— we can recover the spectrum. This suggests that to capture a spectrum bandwidth BW , we can use two ADCs that sample at rates BW/p_1 and BW/p_2 where p_1 and p_2 are co-prime. For example, to recover a 1 GHz spectrum, we can use a 42 MHz ADC [?] along with a 50 MHz ADC. The combination of these two ADCs can capture a bandwidth of 1.05 GHz because $42 \text{ MHz} = 1.05 \text{ GHz}/25$ and $50 \text{ MHz} = 1.05 \text{ GHz}/21$, where 21 and 25 are co-prime. Note that we also repeat each of these co-prime bucketization with a time shift (as explained in §3.2, which requires a total of 4 low-speed ADCs.

B. Resolving Collisions without Co-prime Aliasing Filters

Co-prime aliasing filters are an efficient way to resolve collisions, but they are not necessary. Here, we show how to

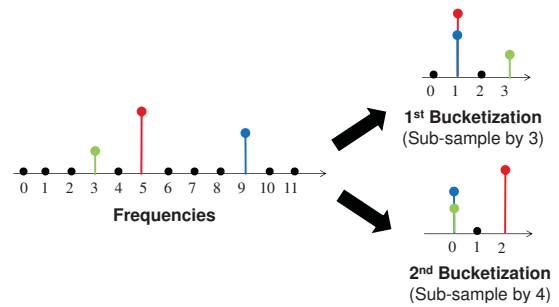


Fig. 3—Resolving collisions with co-prime filters: Using 2 co-prime aliasing filters, we ensure the frequencies that collide in one filter will not collide in the second. For example, frequencies 5 and 9 collide in the first filter. But frequency 5 does not collide in the second which allows us to estimate it and subtract it.

resolve collisions while still using ADCs that sample at the same rate. This means that one can use one type of ADCs for building the whole system.⁴

We use one type of aliasing filter. However, we perform it for more than twice using multiple different time shifts. To see how this can help resolve collisions, consider again the case where two frequencies f and f' collide in a bucket. If we use two time shifts τ_1 and τ_2 , we get three values for each bucket. For the bucket where f and f' collide, these values are:

$$\begin{aligned} \hat{\mathbf{b}}_i &= \hat{\mathbf{x}}_f + \hat{\mathbf{x}}_{f'} \\ \hat{\mathbf{b}}_i^{(\tau_1)} &= \hat{\mathbf{x}}_f \cdot e^{2\pi j f \tau_1} + \hat{\mathbf{x}}_{f'} \cdot e^{2\pi j f' \tau_1} \\ \hat{\mathbf{b}}_i^{(\tau_2)} &= \hat{\mathbf{x}}_f \cdot e^{2\pi j f \tau_2} + \hat{\mathbf{x}}_{f'} \cdot e^{2\pi j f' \tau_2} \end{aligned} \quad (2)$$

If we know the positions of f and f' , the above becomes an overdetermined system of equations where the only unknowns are $\hat{\mathbf{x}}_f$, $\hat{\mathbf{x}}_{f'}$. Since only few frequencies hash into each bucket, there is a limited number of possible values of f and f' . For each of these possibilities, the above over-determined system can be solved to find $\hat{\mathbf{x}}_f$, $\hat{\mathbf{x}}_{f'}$. Hence, we can solve overdetermined system for the possible (f, f') pairs and choose the pair that minimizes the mean square error. While the above does not guarantee that the solution is unique, in case multiple pairs (f, f') satisfy the equations, BigBand can detect that event and report to the user that the values of these frequencies remain unresolved.⁵ Our empirical results (in §7.3) show however that for practical spectrum sparsity (which is about 5%) 3 shifted bucketizations are enough to uniquely resolve the colliding frequencies.

We note that though this method requires more digital computation, we only need to do this for the few buckets that have a collision, and we know the number of collisions is small due to the sparsity of the spectrum.

We also note that this method can be combined with the co-prime approach to deal with less sparse spectrum. In this case, one uses this method to resolve collisions of two frequencies while iterating between the co-prime filters.

4 CHANNEL ESTIMATION AND CALIBRATION

The earlier description of BigBand assumes that the different ADCs can sample exactly the same signal at different time-

4. This makes it possible to build BigBand using only USRPs [39].

5. Note that theoretically, for a collision of k frequencies, $2k$ samples can guarantee a unique solution in the absence of noise.

3. Even if one occasionally falsely detects a collision when there is a single frequency, BigBand can still correct this error. This is because the collision resolution step described next will estimate the values of the presumed colliding frequencies to zero.

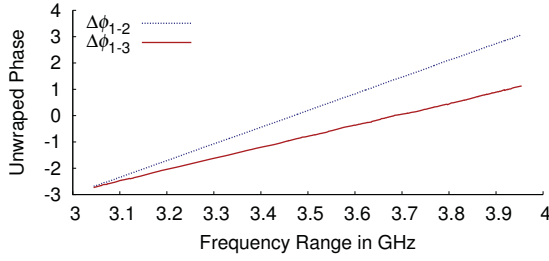


Fig. 4—Phase rotation vs frequency: The figure shows that the phase rotation between the 3 USRPs is linear across the 900 MHz frequency spectrum and can be used to estimate the time shifts.

shifts. However, because the signals experience different channels, they will be scaled differently and the ADCs will not be able to sample exactly the same signal.

To better understand this problem, let us consider the case where we resolve collisions without the co-prime sub-sampling. In this case, we will have 3 ADCs each sampling a signal that is delayed by a time shift. In this case, consider a non-zero frequency f whose value is $\hat{\mathbf{x}}_f$. If f hashes to bucket i and does not collide, then the value of the bucket at each of the ADCs can be written as:

$$\begin{aligned}\hat{\mathbf{b}}_i &= h_w(f) \cdot h_1(f) \cdot \hat{\mathbf{x}}_f \\ \hat{\mathbf{b}}_i^{(\tau_1)} &= h_w(f) \cdot h_2(f) \cdot \hat{\mathbf{x}}_f \cdot e^{2\pi j f \tau_1} \\ \hat{\mathbf{b}}_i^{(\tau_2)} &= h_w(f) \cdot h_3(f) \cdot \hat{\mathbf{x}}_f \cdot e^{2\pi j f \tau_2}\end{aligned}\quad (3)$$

where $h_w(f)$ is the channel on the wireless medium, $h_1(f), h_2(f), h_3(f)$ are the hardware channels on each of the radios, and $\cdot(f)$ indicates that these parameters are frequency dependent. We can ensure that $h_w(f)$ is the same in all three bucketizations by connecting the RF frontends to the same antenna. As a result, $h_w(f)$ cancels out once we take the ratios, $\hat{\mathbf{b}}_i^{(\tau_1)}/\hat{\mathbf{b}}_i$ and $\hat{\mathbf{b}}_i^{(\tau_2)}/\hat{\mathbf{b}}_i$ of the buckets. However, the hardware channels are different for the different bucketizations. We need to estimate them and compensate for them in order to perform frequency estimation and also resolve the collisions.

Furthermore, though it is simple to create time-shifts between the three ADCs as explained in §3.2, we need to know the values of these time-shifts τ_1, τ_2 in order to perform frequency estimation based on phase rotation. Hence, we also need a way to estimate these time-shifts.

4.1 Estimating the Channels and Time-Shifts

To estimate the channels and the time shifts, we divide the total bandwidth BW that BigBand captures into p consecutive chunks. We then transmit a known signal in each chunk, one by one. Since we only transmit in one chunk at a time, there are no collisions at the receiver after aliasing. We then use Eq. 3 to estimate the ratios $h_2(f) \cdot e^{2\pi j f \tau_1} / h_1(f)$ and $h_3(f) \cdot e^{2\pi j f \tau_2} / h_1(f)$ for each frequency f in the spectrum.

Now that we have the ratios, we need to compute $h_2(f)/h_1(f)$ for each frequency f , and the delay τ_1 . We can estimate this as follows: Both the magnitude and phase of the hardware channel ratio will be different for different frequencies. The magnitude differs with frequency because different frequencies experience different attenuation in the hardware. The phase varies linearly with frequency because all frequencies experience the same delay τ_1 , and the phase rotation of a frequency f is simply

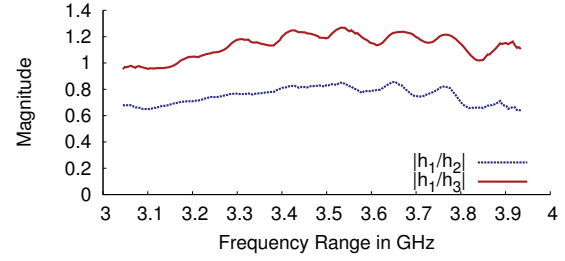


Fig. 5—Hardware channel magnitude: The relative channel magnitudes $|h_1(f)/h_2(f)|$ and $|h_1(f)/h_3(f)|$ are not equal to 1 and are not flat across the frequency spectrum. Hence, we need to compensate for these estimates to be able to detect and solve collisions.

$2\pi f \tau_1$. We can therefore plot the phase of the ratio as a function of frequency, and compute the delay τ_1 from the slope of the resulting line.

Fig. 4 shows the phase result of this estimation performed on the USRP software radios used in our implementation described in §6. As expected, the phase is linear across 900 MHz. Hence, by fitting the points in Fig. 4 to a line we can estimate the shifts τ_1, τ_2 and the relative phases of the hardware channels (*i.e.* $\angle h_1(f)/h_2(f)$ and $\angle h_1(f)/h_3(f)$). Fig. 5 also shows the relative magnitudes of the hardware channels on the USRPs (*i.e.* $|h_1(f)/h_2(f)|$ and $|h_1(f)/h_3(f)|$) over the 900 MHz between 3.05 GHz and 3.95 GHz. These hardware channels and time shifts are stable. For our implementation, we estimated them only once at the set up time.

5 SENSING NON-SPARSE SPECTRUM

We extend BigBand's algorithm to sense a non-sparse spectrum. The key idea is that although the spectrum might not be sparse, changes in spectrum usage are typically sparse, *i.e.*, over short intervals, only a small percentage of the frequencies are freed up or become occupied. This makes it possible to estimate the occupancy without sampling the signal at the Nyquist rate. We refer to sparse changes as differential sparsity, and call the extension that deals with such non-sparse spectrum D-BigBand. We note however that unlike in the case where the spectrum is sparse, in the non-sparse setting we only perform spectrum sensing but we cannot recover the I and Q components of the signal. Below we explain the two components of D-BigBand.

A. Frequency Bucketization: D-BigBand also bucketizes the spectrum using sub-sampling filters. However, since the spectrum is not sparse, it is very likely that all buckets will be occupied. Thus, D-BigBand tries to detect changes in the occupancy of frequencies that hash to each buckets. To do so, D-BigBand computes the average power of the buckets over two consecutive time windows TW by performing the bucketization multiple times during each time window.⁶ Since the changes in spectrum occupancies are sparse, only the average power of few buckets would change between the two time windows. D-BigBand can then focus only on the few buckets where the average power changes.

B. Frequency Estimation: Now that we know in which buckets the average power has changed, we need to estimate

6. The number of times D-BigBand can average is $= TW/T$ where T is the FFT window time.

which of the frequencies in the bucket is the one whose occupancy has changed. However, we can no longer use the phase rotation property to estimate these frequencies or resolve their collisions since the phase of the bucket now depends on all the frequencies that hash to the bucket and not just the frequency whose occupancy has changed. Thus, to estimate the changing frequencies we are going to use a different method which we refer to as voting. We repeat the bucketization multiple times while randomizing which frequencies hash to which buckets. After that, each bucketization votes for frequencies that hash to buckets where the power changed. Frequencies that get the most number of votes are picked as the ones whose occupancy has changed. To randomize the bucketizations, we simply use co-prime sub-sampling which as described in §3.3.2 guarantees that frequencies that hash together in one bucketization can not hash together in the other bucketizations. Further details of how and why this voting approach works can be found in the technical report [37].

As with any differential system, we need to initialize the state of spectrum occupancy. However, an interesting property of D-BigBand is that we can initialize the occupancy of each frequency in the spectrum to unknown. This is because, when we take the difference in power we can tell whether the frequency became occupied or it became empty. Specifically, a negative power difference implies that the corresponding frequency became empty, and a positive power difference implies that the corresponding frequency became occupied. Hence, once the occupancy of a frequency changes, we can tell its current state irrespective of its previous state. This avoids the need for initialization and prevents error propagation.

6 A USRP-BASED IMPLEMENTATION

A. Implementing BigBand: As a proof of concept, we implement BigBand using USRP N210 software radios [39]. Since the USRPs use the same ADCs, it is not possible to have co-prime sub-sampling rates. Thus, our implementation relies on resolving collisions without co-prime sub-sampling.

We use three USRP N210 radios with the SBX daughterboards, which can operate in the 400 MHz to 4.4 GHz range. The clocks of the three USRPs are synchronized using an external GPSDO clock [40]. In order to sample the same signal using the three USRPs, we connect the USRPs to the same antenna using a power splitter but with wires of different lengths in order to introduce small time-shifts. We also remove the analog low pass filters on the SBX daughterboards to allow the USRP's ADC to receive the entire bandwidth that its analog front-end circuitry is designed for. The analog circuitry of the USRP front-end can receive at most 0.9 GHz, which puts an upper bound on the digital bandwidth of the system. The three USRP ADCs each samples the signal at 50 MS/s.⁷ Thus, our implementation of BigBand captures a bandwidth $BW = 900$ MHz using only 150 MS/s.

B. Implementing D-BigBand: D-BigBand's frequency estimation relies on using different co-prime sub-sampling rates and hence we cannot implement D-BigBand directly on USRPs. Thus, to verify that D-BigBand can sense a non-sparse

⁷ In principle, the USRP ADC can sample up to 100 MS/s. However, the USRP digital processing chain cannot support this rate and hence the ADC sampling rate can be set to no higher than 50 MS/s.

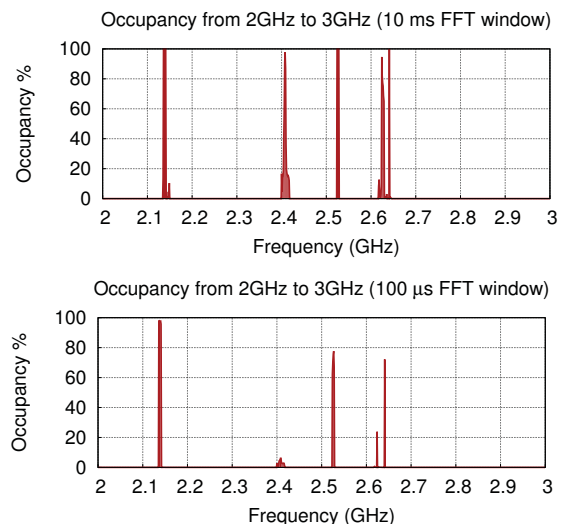


Fig. 6—Spectrum Occupancy: The figure shows the average spectrum occupancy at our geographical location on Friday 01/15/2013 between 1-2pm, as viewed at a 10 ms granularity (top) and 100 μ s granularity (bottom). It shows that the spectrum is sparsely occupied. Further, the sparsity increases when one computes the occupancy over shorter time windows.

spectrum, we use trace-driven experiments. To collect traces of one GHz of highly occupied spectrum, we use many USRPs to transmit and receive. Since we have a total of 20 USRPs, we divide them into 10 receivers and 10 transmitters and capture 250 MHz at a time. We repeat this 4 times at center frequencies that are 250 MHz apart and stitch them together in the frequency domain to capture the full 1 GHz spectrum. We then perform the inverse FFT to obtain a time signal sampled at 1 GHz. We now sub-sample this time domain signal using three co-prime rates: 1/21, 1/20, and 1/23 GHz. We run D-BigBand using these sub-sampled versions of the signal.

7 BIGBAND'S SPECTRUM SENSING RESULTS

7.1 Outdoor Spectrum Sensing

We collect outdoor measurements from the roof top of a 24 floor MIT building. We use BigBand to capture the signal between 2 GHz and 2.9 GHz over 30 minutes. We configure BigBand to compute the spectrum over an FFT window of size W . We report here results for $W = 10$ ms and $W = 100\mu$ s. We calculate the occupancy of a particular frequency as the percentage of the FFT windows during which the frequency was occupied (*i.e.*, the power at that frequency was at least twice the noise power).

Fig. 6 shows the fraction of time that each chunk of spectrum between 2 GHz and 2.9 GHz is occupied, as recovered by BigBand. These results were confirmed using a spectrum analyzer. The figure shows that the spectrum is sparsely occupied. In particular, the occupancy is about 5% when considered over FFT windows of 10 ms and drops to about 2%, when viewed over windows of 100 μ s. The figure shows that even frequencies that look 100% occupied over 10 ms windows, become less occupied when viewed over shorter intervals. This is because while these frequencies are occupied for some fraction of every 10 ms interval, there is a large number of

FFT Window	BigBand (900 MHz)	3 USRP Seq. Scan (150 MHz)	RFeye Scan (20 MHz)
1 μ s	1 μ s	48 ms	22.5 ms
10 μ s	10 μ s	48 ms	22.5 ms
100 μ s	100 μ s	48 ms	—
1 ms	1 ms	54 ms	—
10 ms	10 ms	114 ms	—

TABLE 1—Scanning time: BigBand is multiple orders of magnitude faster than other technologies. This allows it to perform real-time sensing to take advantage of even short term spectrum vacancies.

shorter windows within each 10 ms where these frequencies are not occupied. For example, the WiFi band around 2.4 GHz seems fully utilized when checked over 10 ms windows; yet if one views it over windows that are 100 times shorter (*i.e.*, 100 μ s), one would discover that the medium is almost always idle. In contrast, the band around 2.1 GHz which is used by cellular technologies is occupied even at very short time scales.

The above implies that the spectrum is sparser at finer time intervals, and provides more opportunities for fine-grained spectrum reuse. This result motivates the need for fast spectrum sensing schemes to exploit these short-term vacancies.

Finally, we note that measurements collected in other locations or on different dates show similar results to those in Fig. 6 but may differ slightly in which frequencies are occupied. Measurements from higher parts of the spectrum are qualitatively similar but have significantly higher sparsity (we omit the figures for lack of space).

7.2 BigBand vs. Spectrum Scanning

Most of today’s spectrum sensing equipment relies on scanning. Even expensive, power hungry spectrum analyzers typically capture a 100 MHz bandwidth in one shot, and end up scanning to capture a larger spectrum [6]. The performance of sequentially scanning the spectrum depends mainly on how fast the device can scan a GHz of bandwidth. In the absence of fast scanning, the system can miss radar and other highly dynamic signals. Here, we compare how fast it would take to scan the 900 MHz bandwidth using three techniques: state-of-the-art spectrum monitors like the RFeye [5], which is used in the Microsoft spectrum observatory, 3 USRPs sequentially scanning the 900 MHz, or 3 USRPs using BigBand.

Table 1 shows the results for different FFT window sizes. In all cases, BigBand takes exactly the time of the FFT window to acquire the 900 MHz spectrum. The 3 USRPs combined can scan 150 MHz at a time and hence need to scan 6 times to acquire the full 900 MHz. For FFT window sizes lower than 10 ms, the scanning time is about 48 ms. Hence, the USRPs spend very little time actually sensing the spectrum, which will lead to a lot of missed signals. Of course, state of the art spectrum monitors can do much better. The RFeye node has a fast scanning mode of 40 GHz/second [5]. It scans in chunks of 20 MHz and thus will take 22.5 ms to scan 900 MHz. Note that RFeye has a maximum resolution of 20 kHz, and hence does not support FFT windows larger than 50 μ s.

Thus, BigBand, which uses off-the-shelf components, is much faster than even expensive scanning based solutions, allowing it to detect short-term spectrum vacancies.

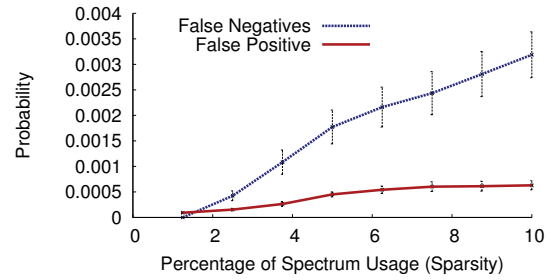


Fig. 7—False negatives and positives as a function of spectrum sparsity: BigBand’s false positive and false negative rates are extremely low.

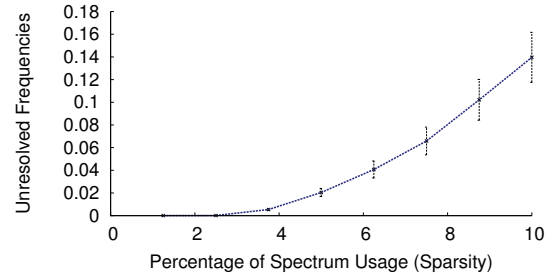


Fig. 8—Unresolved frequencies as a function of spectrum sparsity: BigBand cannot resolve around 2% of the frequencies with 5% spectrum occupancy, and stays below 14% even when spectrum occupancy grows as large as 10%.

7.3 BigBand’s Sparsity Range

The primary motivation of BigBand is to be able to sense sparse spectrum. In this section, we verify the range of sparsity for which BigBand works. We run our experiments between 3.05 GHz and 3.95 GHz because this band is effectively empty (see Fig. 1), and hence enables us to perform controlled experiments. We vary the sparsity in the 3.05 GHz to 3.95 GHz range between 1% and 10% by transmitting from 5 different USRPs. Each USRP transmits a signal whose bandwidth is at least 1 MHz and at most 20 MHz. We randomize the bandwidth and the center frequencies of the signals transmitted by the USRPs. For each sparsity level, we repeat the experiment 100 times with different random choices of bandwidth and center frequencies. We run BigBand over a 1 ms FFT window. We consider three metrics:

- **False Negatives:** The fraction of occupied frequencies that BigBand incorrectly reports as empty.
- **False Positives:** The fraction of empty frequencies that BigBand incorrectly reports as occupied.
- **Unresolved Frequencies:** The fraction of total frequencies that BigBand cannot resolve due to unresolved collisions.

Fig. 7 shows that BigBand’s false positives and false negatives rates are extremely low. The probability of false positive stays below 0.0005 even when 10% of the spectrum is occupied. The probability of false negative is less than 0.002 when the spectrum occupancy is less than 5%, and stays within 0.003 even when the spectrum occupancy goes up to 10%.

Fig. 8 shows that the fraction of unresolved frequencies is less than 0.03 when the spectrum usage is below 5%. This number increases as the spectrum usage increases, but stays below 0.14 when 10% of the spectrum is used. Unresolved frequencies increase as spectrum usage increases because the

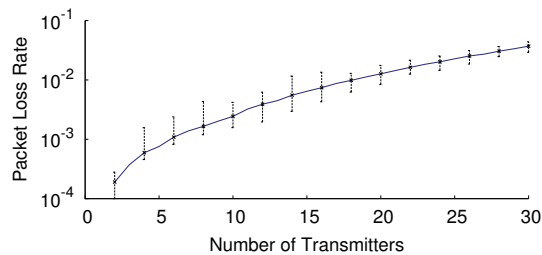


Fig. 9—BigBand’s Packet Loss as a function of the number of simultaneous transmitters: BigBand can decode as many as 30 transmitters spread across a 900 MHz wide band, while keeping the packet loss less than 3.5%.

probability of collision increases. Note however that in contrast to false positive and false negatives, BigBand knows which exact frequencies it could not resolve and reports these frequencies with the label “not-estimated”. Thus, unresolved frequencies show lack of information as opposed to errors. The application can decide how to treat unresolved frequencies. For dynamic spectrum access, it can simply avoid the unresolved frequencies.

We also note that real-world spectrum measurements, for instance, in the Microsoft observatory, and in this paper, reveal that actual spectrum usage is 2–5%. In this regime, BigBand’s unresolved frequencies are less than 0.03. Further, if the occupancy is high, one may use D-BigBand, which deals with high occupancies (see results in §9.)

8 BIGBAND’S DECODING RESULTS

8.1 Decoding Multiple Transmitters

In this section, we verify that BigBand can concurrently decode a large number of transmitters from diverse parts of the spectrum. All the transmitters in our implementation use the same technology, but the result naturally generalizes to transmitters using different technologies.

We use 10 USRPs to emulate up to 30 transmitters hopping in a spectrum of 0.9 GHz. At any given time instant, each device uses 1 MHz of spectrum to transmit a BPSK signal. Similar to the Bluetooth frequency hopping standard, we assume that there is a master that assigns a hopping sequence to each device that ensures that no two devices hop to the same frequency at the same time instant. Note however, that the hopping sequence for different devices allows them to hop to frequencies that get aliased to the same bucket at a particular time instant, and hence collide in BigBand’s aliasing filters. Like in Bluetooth, each transmitter hops 1, 3, or 5 times per packet, depending on the length of the packet.

Fig. 9 shows the packet loss rate versus the number of devices hopping in the spectrum. It shows that BigBand can decode the packets from 30 devices spanning a bandwidth of 900 MHz with a packet loss rate less than 3.5%. Decoding all these transmitters without BigBand would either require a wideband 0.9 GHz receiver, or a receiver with 30 RF-frontends, both of which would be significantly more costly and power-hungry.

8.2 Signal-to-Noise Ratio

It is expected that BigBand will have more noise than a narrowband receiver since it can capture a much larger

ADC Quantization	BigBand vs Narrowband RX	
	mean	max
8 bits	-2.73 dB	-2.78 dB
14 bits	-5.68 dB	-5.89 dB

TABLE 2—Reduction in SNR at different quantization levels

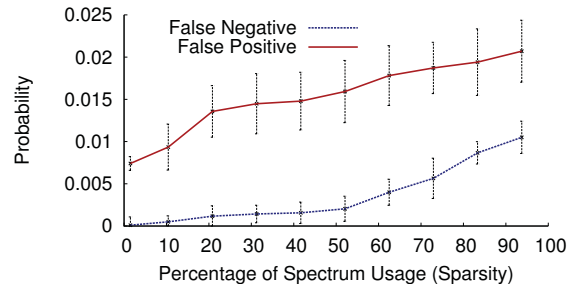


Fig. 10—D-BigBand’s effectiveness as a function of Spectrum Sparsity: Over a band of 1 GHz, D-BigBand can reliably detect changes in spectrum occupancy even when the spectrum is 95% occupied, as long as the change in spectrum occupancy is less than 1% every ms.

bandwidth. This section aims to shed insight on this issue. We note three types of noise: thermal noise, quantization noise and ADC jitter noise [19]. BigBand has higher thermal noise due to bucketization. Specifically, since in our implementation, the 900 MHz bandwidth is aliased into 50 MHz, it is expected that the thermal noise would increase by $18\times$ (12.5 dB). However, quantization noise and ADC jitter noise do not alias, and hence do not increase. The overall increase in noise depends on how the thermal noise compares to these other types of noise.

To understand the impact of thermal noise and quantify the SNR performance of BigBand we compare it with a 50 MHz narrowband receiver that uses the same USRP hardware. We transmit a 10 MHz signal, receive it on BigBand and the narrowband receiver, and compare the resulting SNR. We connect BigBand and the narrowband receiver to the same antenna and ensure that both receivers’ rx-gains are set properly so that the received signal amplitude spans the same range on both receivers. We run it for different receive signal strengths and measure the SNR on each. We repeat the measurements for the ADC quantization set to 8 bits and 14 bits to better understand the interaction between thermal noise and quantization noise.

Table 2 shows the mean and max reduction in SNR of a signal received on BigBand relative to the narrowband USRP. The result shows that at 8 bit quantization, the reduction is a little less than 3 dB which means that the 12 dB increase in thermal noise only translates to 3 dB reduction in SNR due to quantization and jitter noise. At a quantization of 14 bits, the SNR reduction becomes 6 dB which means that the ADC jitter noise is still significantly higher than thermal noise. Though this reduction in SNR is significant compared to narrowband receivers, one would require using 18 such receivers to capture in realtime the same 900 MHz bandwidth as BigBand which is not practical in terms of cost and bulkiness.

9 D-BIGBAND’S SENSING RESULTS

In this section, we evaluate D-BigBand’s ability to sense changes in spectrum occupancy independent of sparsity. We implement D-BigBand as described in §6. We vary the percentage of total occupied frequencies in the spectrum between

1% (sparse) to 95% (almost fully occupied). We then change the number of frequencies that change occupancy every 1 ms by up to 1% (*i.e.*, 10 MHz), and evaluate D-BigBand's accuracy in identifying the frequencies that change occupancy.

As a function of spectrum occupancy, Fig. 10 shows the false positives (*i.e.*, frequencies whose occupancy has not changed, but D-BigBand erroneously declared as changed) and false negatives (*i.e.*, frequencies whose occupancy has changed, but D-BigBand erroneously declares as unchanged). We see that D-BigBand robustly identifies changes in occupancy, with both the false positive and the false negative probabilities remaining under 0.02 even for a spectrum occupancy of 95%.

10 CONCLUSION

This paper presents BigBand, a system that enables GHz-wide sensing and decoding using commodity radios. Empirical evaluation demonstrates that BigBand is able to sense the spectrum stably and dynamically under different sparsity levels; we also demonstrate BigBand's effectiveness as a receiver to decode GHz-wide sparse signals. We believe that BigBand enables multiple applications that would otherwise require expensive and power hungry devices, *e.g.* realtime spectrum monitoring, dynamic spectrum access, concurrent decoding of multiple transmitters in diverse parts of the spectrum.

Acknowledgments: We thank Hariharan Rahul, Keith Winstein and Piotr Indyk for all their help, support and insightful comments. This work is funded by NSF.

REFERENCES

- [1] FCC, Second Memorandum Opinion & Order 10-174, <http://www.fcc.gov/encyclopedia/white-space-database-administration>.
- [2] Microsoft Spectrum Observatory, <http://spectrum-observatory.cloudapp.net>.
- [3] PCAST: Realizing the full potential of government held spectrum to spur economic growth, 2012, http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_spectrum_report_final_july_20_2012.pdf.
- [4] T. Baykas et al., "Developing a standard for TV white space coexistence," *Wireless Comm, IEEE*, vol. 19, no. 1, 2012.
- [5] RFeye Node, <http://media.crfs.com/uploads/files/2/crfs-md00011-c00-rf-eye-node.pdf>.
- [6] Tektronix Spectrum Analyzer, <http://tek.com>.
- [7] Y. M. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan, and S. C. Wang, "A 40 gs/s 6b adc in 65 nm cmos," in *IEEE Solid-State Circuits Conf. (ISSCC)*, 2010.
- [8] B. Murmann, "A/d converter trends: Power dissipation, scaling and digitally assisted architectures," in *IEEE Custom Integrated Circuits Conf. (CICC)*, 2008.
- [9] DigiKey, ADCs, <http://www.digikey.com/>.
- [10] Y. Tanet et al., "A 2.4 ghz wlan transceiver with fully-integrated highly-linear 1.8 v 28.4 dbm pa, 34 dbm t/r switch, 240 ms/s dac, 320 ms/s adc, and dpll in 32 nm soc cmos," in *IEEE Symposium on VLSI Technology and Circuits*, 2012.
- [11] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Simple and practical algorithm for sparse FFT," in *SODA*, 2012.
- [12] —, "Nearly optimal sparse fourier transform," in *STOC*, 2012.
- [13] B. Ghazi, H. Hassanieh, P. Indyk, D. Katabi, E. Price, and L. Shi, "Sample-Optimal Average-Case Sparse Fourier Transform in Two Dimensions," *Allerton*, October 2013.
- [14] J. Laska, W. Bradley, T. Rondeau, K. Nolan, B. Vigoda. Compressive sensing for dynamic spectrum access networks: Techniques tradeoffs. *DySPAN*, 2011.
- [15] M. Rashidi, K. Haghighi, A. Panahi, and M. Viberg, "A NLLS based sub-nyquist rate spectrum sensing for wideband cognitive radio," in *DySPAN*, 2011.
- [16] J. Yoo, S. Becker, M. Loh, M. Monge, E. Candès, and A. E-Neyestanak, "A 100MHz–2GHz 12.5x subNyquist rate receiver in 90nm CMOS," in *IEEE RFIC*, 2012.
- [17] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk, "Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, 2010.
- [18] O. Abari et al., "Performance trade-offs and design limitations of analog-to-information converter front-ends," in *ICASSP*, 2012.
- [19] O. Abari, F. Lim, F. Chen, and V. Stojanovic, "Why analog-to-information converters suffer in high-bandwidth sparse signal applications," *IEEE Transactions on Circuits and Systems I*, 2013.
- [20] A. Gilbert, M. Muthukrishnan, and M. Strauss, "Improved time bounds for near-optimal space fourier representations," in *SPIE*, 2005.
- [21] S. Pawar and K. Ramchandran, "Computing a k-sparse n-length discrete fourier transform using at most 4k samples and o(k log k) complexity," *arXiv preprint arXiv:1305.0870*, 2013.
- [22] J. Yoo et al., "A compressed sensing parameter extraction platform for radar pulse signal acquisition," *JETCAS*, 2012.
- [23] M. Mishali and Y. Eldar, "From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.
- [24] —, "Wideband Spectrum Sensing at Sub-Nyquist Rates," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 102–135, 2011.
- [25] X.-G. Xia, "On estimation of multiple frequencies in undersampled complex valued waveforms," *Signal Processing, IEEE Transactions on*, vol. 47, no. 12, pp. 3417–3419, 1999.
- [26] —, "An efficient frequency-determination algorithm from multiple undersampled waveforms," *Signal Processing Letters, IEEE*, vol. 7, no. 2, pp. 34–37, 2000.
- [27] P. Vaidyanathan and P. Pal, "Sparse Sensing With Co-Prime Samplers and Arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, 2011.
- [28] C. Herley and P. Wong, "Minimum rate sampling and reconstruction of signals with arbitrary frequency support," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1555–1564, 1999.
- [29] R. Venkataramani and Y. Bresler, "Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2173–2183, 2000.
- [30] N. Tzou, D. Bhatta, S.-W. Hsiao, H. W. Choi, and A. Chatterjee, "Low-Cost Wideband Periodic Signal Reconstruction Using Incoherent Undersampling and Back-end Cost Optimization," in *IEEE Inter. Test Conference*, 2012.
- [31] A. Silva, "Reconstruction of Undersampled Periodic Signals," 1986, mIT Technical Report.
- [32] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 1, 2009.
- [33] P. Bahl et al., "White space networking with wi-fi like connectivity," in *ACM SIGCOMM*, 2009.
- [34] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to Share: Narrowband-Friendly Wideband Networks," in *ACM SIGCOMM*, 2008.
- [35] S. S. Hong and S. R. Katti, "DOF: a local wireless information plane," in *ACM SIGCOMM*, 2011.
- [36] S. Yoon, L. E. Li, S. Liew, R. R. Choudhury, K. Tan, and I. Rhee, "Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access wireless networks," in *IEEE INFOCOM*, 2013.
- [37] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "BigBand: GHz-Wide Sensing and Decoding on Commodity Radios," May 2013, mIT-CSAIL-TR-2013-009.
- [38] R. Lyons, *Understanding Digital Signal Processing*, 1996.
- [39] Ettus Inc. USRP, <http://ettus.com>.
- [40] Jackson Labs, Fury GPSDO, <http://jackson-labs.com/>.