Title
Energy Efficient Acquisition and Inferencing for Low Power Physiological Sensing

Author
Charbiwala, Zainul

Publication Date
2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Energy Efficient Acquisition and Inferencing for Low Power Physiological Sensing

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Zainul Mohammed Charbiwala

2012

ABSTRACT OF THE DISSERTATION

Energy Efficient Acquisition and Inferencing for Low Power Physiological Sensing

by

Zainul Mohammed Charbiwala

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Mani B. Srivastava, Chair

Affordable, wearable, embedded, wireless medical sensor systems that enable continuous long term monitoring of physiological signals could revolutionize health care. Realizing this vision demands devices that are small, unobtrusive and low power. Effectively inferring health conditions begins by acquiring physiological signals of interest and decisions made about what signals are acquired, when, where and at what rate affect not only the energy efficiency of the sampling process but also that of other downstream components in the signal processing chain.

While the Nyquist sampling theorem provides for exact reconstruction from discrete-time samples, the prescribed rate is often wasteful for physiological sensing applications since it neither exploits the structure of signals fully nor does it take into account that many applications don't require full reconstruction at all. This dissertation illustrates how energy efficiency of the entire system can be improved by targeting just the signal acquisition process while being cognizant of the entire sensing information stack, from sampling, processing and communication to the top-level application inferences.

A key ingredient that makes optimizing the sensing stack worthwhile is that the sampling stage, which is usually abstracted away from the system, can now utilize sophisticated methods that have emerged in the past few years. Recent advances in sampling

and recovery techniques have demonstrated considerable rate reductions by employing stronger models of the phenomenon coupled with application-specific objectives (detection or control vs. reconstruction), which potentially translates to higher energy, processing and communications efficiency at the system level.

This research describes four major thrusts that span the processing chain from hardware to algorithms to inferences. First, recognizing that signal conditioning front-end circuits could account for a large portion of the energy expenditure in low power sensing, we demonstrate how prudently duty cycling them could increase device lifetime by threefold and reduce data rate by almost fourfold for an electrocardiography monitor. Then, we go on to show how one could further slash data rates using the new theory of compressed sensing. For a neural spike recorder, we exploit the fact that action potentials have both a structure and short term stability in their morphology. This meant that we could utilize historical signal information to optimize and adapt compressed sensing recovery, with only receiver-side modifications, doubling the compression ratio.

Third, since body area networks are prone to congestion and interference, we propose a rate control algorithm for the wireless channel so that the most important data from the most informative sensors gets delivered for maximum inference quality. Finally, we prove that compressed sensing could be utilized not only to compress signals but could also improve the robustness of sensor transmissions at low computational cost by viewing it as joint source-channel coding for wireless erasure channels.

The dissertation of Zainul Mohammed Charbiwala is approved.

Mario Gerla

Paulo Tabuada

William J. Kaiser

Mani B. Srivastava, Committee Chair

University of California, Los Angeles

2012

Table of Contents

List of Figures

List of Tables

Acknowledgments

This work would not have been possible without the input and effort of many. First and foremost, my advisor, mentor and friend, Mani B. Srivastava. Mani's unique perspective on life in research, his guidance and high standards have surely put me through my paces. For his unmistakable ability to make me productive in the most gentle of ways, I've crowned him my benevolent dictator. Mani has shaped me in more ways that he can imagine or I can express and I'll always be grateful to him. The second person to be credited for this work and, for who I am, is my wife. This document only exists because of her unconditional and unending support for my ludicrous ideologies about knowledge, experiences and changing the world. She has made many sacrifices so that I could follow my dream. I can't thank her enough for everything but I would hate to reveal to her that this is just the beginning. I'd also like to thank my parents, who've always been there for me and who've pampered me and believed in me. They've made me feel so special that even I've come to believe that I'm capable of anything.

A lot of my research grew out of discussions and collaboration and I'd like to thank my committee, all my colleagues at the Networked and Embedded Systems Lab, the Center for Embedded Networked Sensing and my co-authors on various publications. I'd like to place on record my gratitude, in particular (and in no particular order), to Jonathan Friedman, Benjamin Kuris, Vaibhav Karkare, Sarah Gibson, Dejan Markovic, Supriyo Chakraborty, Sadaf Zahedi, Younghun Kim, Ting He, Chatschik Bisdikian, Young H Cho, Paul Martin, Henry Herman, Kasturi Rangan Raghavan, Lucas Wanner, Greg Pottie, Thomas Schmid and Harris Teague. These individuals have contributed to this work and to me beyond what is within this manuscript. They've helped me grow intellectually and they've left me a small part of each one of them. I'd also like to thank Fe Asuncion for her assistance with every little thing I needed for my research and to Deeona Columbia and her team for being the best student officers ever.

I must add that this material is supported in part by the U.S. Army Research Laboratory and the U.K. Ministry Of Defense under Agreement Number W911NF-06-3-0001, the U.S. Office of Naval Research under MURI-VA Tech Award CR-19097-430345, the National Science Foundation under grant CCF-0820061, CNS-0910706, CNS-0905580, 0824275, and 0847088, the UCLA Center for Embedded Networked Sensing and through two Qualcomm Fellowships. Any opinions, findings and conclusions or recommendations expressed in this material are mine and do not necessarily reflect the views of the listed funding agencies.

Finally, I'd like to thank Prof. Richard Staba (UCLA Dept. of Neurology) for providing access to neural recordings for evaluating our systems, Hariprasad Chandrakumar and Neha Sinha for their inputs regarding op-amp design, Prof. Sudhakar Pamarti for sharing his perspectives on analog circuit duty-cycling, Abhishek Ghosh for his insight on switch parasitic capacitance and Prof. Justin Romberg for early comments on the CapMux architecture.

Vita

1995--1999	Bachelor of Engineering, Electronics and Telecommunication University of Mumbai Mumbai, India.
1999--2002	Master of Technology, Communication Engineering Indian Institute of Technology, Bombay Mumbai, India.
2002--2006	Chief System Architect Eisodus Networks Pvt. Ltd. Mumbai, India.
2006--2007	Henry Samueli Fellowship Univerity of California, Los Angeles Los Angeles, USA.
2007	Top rank in EE Preliminary Exam Univerity of California, Los Angeles Los Angeles, USA.
2009--2010	Qualcomm Research Fellowship, Univerity of California, Los Angeles Los Angeles, USA.
2010--2011	Dissertation Year Fellowship, Univerity of California, Los Angeles Los Angeles, USA.
2011--2012	Qualcomm Innovation Fellowship, Univerity of California, Los Angeles Los Angeles, USA.

Publications

Toward Quality of Information Aware Rate Control for Sensor Networks Fourth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks , April 2009.

Energy Efficient Sampling for Event Detection in Wireless Sensor Networks Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED) , August 2009.

Compressive Oversampling for Robust Data Transmission in Sensor Networks The 29th Conference on Computer Communications (INFOCOM) , March 2010.

Compressive Sensing of Neural Action Potentials Using a Learned Union of Supports Proceedings of the International Conference on Body Sensor Networks (BSN 2011) , May 2011.

Filters That Remember: Duty Cycling Analog Circuits for Long Term Medical Monitoring Wireless Health 2011 , October 2011.

CapMux: A Scalable Analog Front End for Low Power Compressed Sensing Submitted to The Third International Green Computing Conference, February 2012.

CHAPTER 1

Introduction

1.1 Energy Efficient Sampling for Physiological Sensing

As the threat of chronic illnesses overtakes that of communicable diseases, the traditional model of episodic care no longer suffices [ES10]. Symptoms related to chronic conditions, in particular, are hard to capture in clinical settings due to their ephemeral, and oftentimes, rare nature [RGF01]. Affordable, wearable, embedded, wireless medical sensor systems that enable continuous long term monitoring of physiological signals could help revolutionize health care [SO11]. Collecting data in this manner produces long-run, high-quality datasets, the analysis of which has already demonstrated potential in preventive medicine, stress inferencing [PRH11], and self-care [Dep05]. One of the key hurdles to realizing this vision is the availability of devices that are unobtrusive, inexpensive and low power [OOM, Shi10a] for higher user acceptance and, ultimately, early detection of conditions before they evolve into chronic illnesses.

Effectively inferring health conditions begins at the sensor sampling interface (Figure 1.1), where physiological signals of interest are discretized in time and amplitude. Decisions made about what signals are acquired, when, where and at what rate affect not only the energy efficiency of the sampling process but also that of other components in

Figure 1.1: The physiological sensing process begins with sampling at the sensor interface and ends at the health inference. Separate pathways may be present between blocks for feed forward and feedback control signals.

the signal processing chain. The sensor data is processed to either extract simple inferences directly on the sensor device or to remove redundancy for efficient transport to a back-end system for complex analytics. Physiological sensing applications today utilize the ubiquity, connectivity and processing power of smart phone technologies to provide a platform for performing more complex inferences, or as a bridge to a sufficiently capable computing engine that can use population-scale or personalized models to construct relevant inferences.

The famous Whittaker-Shannon-Nyquist [Whi15, Sha49, Nyq28] sampling theorem provides sufficiency conditions for exact reconstruction from discrete-time samples, prescribing a periodic acquisition rate of at least twice the signal bandwidth for reconstruction to be lossless. This rate is often wasteful for physiological sensing applications since it neither allows designers to exploit additional information about the structure of signals (beyond bandlimited-ness) nor does it take into account that full reconstruction may be unnecessary for some applications. This dissertation explores techniques that *enhance systemic energy efficiency by targeting the signal acquisition process while being cognizant of the entire sensing information chain, from sampling, processing and communication to eventual application inferences.*

This effort is similar in essence to works that empower the layers of the communication stack with information from other layers, except that our approach targets the 'sensing stack'. A key ingredient that makes optimizing the sensing stack worthwhile is that the sampling stage, which was previously abstracted away from the system, can utilize sophisticated methods that have recently emerged.

Recent advances in sampling and recovery techniques have demonstrated considerable rate reductions by employing stronger models of the phenomenon coupled with application-specific objectives (detection or control vs. reconstruction). For example, spatiotemporal correlation in the acquired signals can be used to adapt the sampling rate such that only as much data is collected as there is information present (for entropy reduction) [SKG]. Furthermore, the sampling process can also exploit the fact that many

application objectives can be met without reconstructing the signal completely. For instance, event detection in the presence of Gaussian noise can be performed by acquiring just enough data to compute a sufficient statistic [Kay98, MW95] and linear control objectives can be met without continuously sampling system state [AT08]. Acquiring the signal frugally and prudently in this way not only alleviates burden at the sampling stage but also potentially translates to higher energy, processing and communications efficiency at the system level.

Our research seeks to capture these efficiencies by investigating the sampling block in depth, while focusing on the system level impacts of applying prudent sampling strategies. We explore techniques such as compressive sensing and model-based learning for adaptive sampling and quantify the benefits that accrue from them, while taking the end-to-end application into consideration.

Since the sampling stage is fundamentally connected to the continuous-valued physical world, sampling is an inherently analog domain process and realizing rate reductions from these sophisticated sampling techniques typically involves some pre-sampling analog circuits to condition and process the signal beyond the usual anti-alias filtering. In our platform designs, we leverage analog circuit advances to create custom sampling interfaces that interact with signals in order to derive the most information at the lowest practical rates.

As an overarching goal, we seek to systematically and jointly, optimize and manage the multiple phases of the entire physical-to-cyber, information-acquisition, processing, communication and inference pipeline for specific application objectives related to physiological sensing while satisfying their low power and system resource constraints.

1.2 Contributions of this Dissertation

This dissertation explores the span of the sensing chain from hardware to algorithms to inferences (Figure 1.2) and reports on six major contributions. The following sections

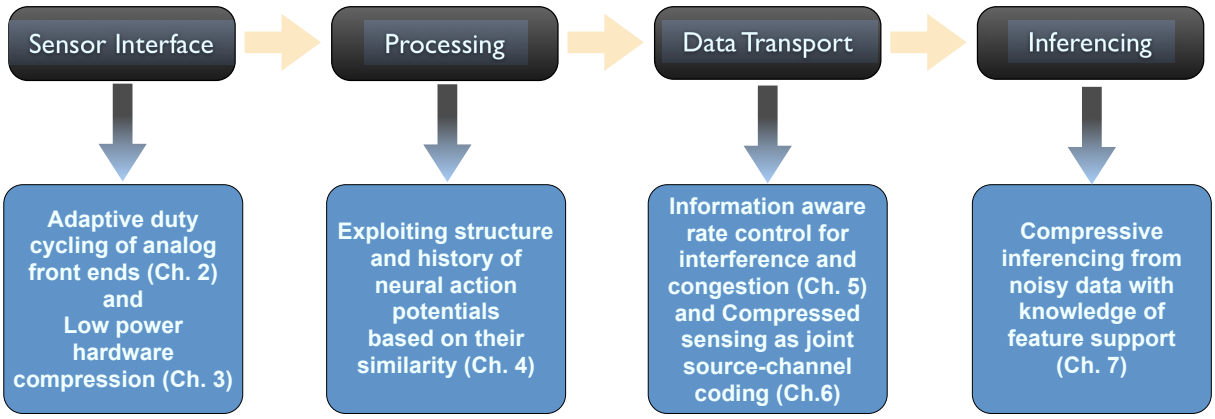

Figure 1.2: An overview of the research conducted in this dissertation. Each block of the sensing process has been studied in depth to show how prudent sampling can be designed to enhance its performance.

introduce our research into sampling optimization techniques and provide an overview of the work described in detail in later chapters. We begin in Section 1.3 by focusing on an electrocardiography (ECG) monitor designed for recording heart conditions to manage stress and to detect abnormal changes in heart rhythm (arrhythmia). Since both these conditions can occur at any time, a long-term continuous monitoring system is essential. However, low battery lifetime typical of these devices [OOM, Shi10a] has resulted in reduced acceptance especially for users casually monitoring their physiological status. Note that this user demographic is the primary target for smarter health care because the principal purpose is to detect early onset so that it can be treated before it evolves into a chronic condition [RGF01].

Signal conditioning front-end circuits that transduce weak electrical signals picked up at the skin interface could account for a large portion of the overall energy expenditure in a low power monitoring system. A common mechanism for reducing average power consumption is to use duty-cycling to switch circuits to a power saving mode when they are not needed. Duty-cycling cannot be used as-is with front-end circuits because they contain filters with large settling times resulting in circuit wake up delay that is too high to be useful. In Section 1.3, with full details in Chapter 2, we illustrate how we save the state of the filter between duty-cycle instances, reducing the wake up delay by over

three orders of magnitude. Through an additional signal specific adaptation, we increase overall device lifetime by threefold and reduce data rate by almost fourfold.

Many physiological signals have structure to them and some even repetition. Scientists have been able to utilize this structure to compress the signal waveforms for more efficient transport, storage and analysis. However, a key problem that remains is that these signals are usually first sampled, rather wastefully, at the Nyquist rate and only then parametrized and compressed. Compressed sensing (CS) is a new technique that promises to directly produce a compressed version of a signal by projecting it to a lower dimensional but information preserving domain before the sampling process. Continuing our exploration on the analog hardware front, Section 1.4 briefly describes the CapMux system (full details in Chapter 3). The key idea behind CapMux is that it uses a single shared signal processing chain in time multiplexed fashion to project the signal onto a set of pseudo-random sparse binary basis functions. We built and characterized a proof-of-concept 16-channel CapMux system and found that our circuit consumes 20μA on average while providing over 30dB SNR recovery in most instances.

Next, we turn our attention in Section 1.5 to efficient acquisition of relatively high bandwidth signals -- neural action potentials or ‘‘spikes’’ from multiple electrodes implanted deep within the brain. Neuroscientists are interested primarily in identifying which neurons fire, and when, based on a provided external stimulus. In order to conduct experiments in a natural setting neuroscientists need wireless neural recording. A key problem neuroscientists face is that because of high sampling rates coupled with the need for a large number of electrodes, the energy consumption of the radio transceiver is prohibitively high. This precludes wireless neural recording for most subjects due to the size of the battery needed. We explore the use of compressed sensing, which allows one to compress the signal in the course of sampling, as a way to reduce the data throughput requirements, and hence, the power consumed by the radio transceiver. As we show in Chapter 4, however, directly applying compressed sensing is inadequate because it does not provide sufficient rate reductions. We exploit the fact that action potential wave-

forms have similar morphology, especially when viewed in the compressed domain. We use knowledge of the similarity from previously recovered spikes to enhance the recovery of future spikes. This learning technique halves the data rate requirements, enough for experimentation on smaller animals for longer durations.

Section 1.6 (detailed in Chapter 5) discusses optimizing the sampling rate in a synchronous sensing regime by considering the interference caused by data from multiple sensors in a multi-hop network. Since body area networks are prone to congestion and interference, we propose a rate control algorithm for the wireless channel so that the most important data from the most informative sensors gets delivered for maximum inference quality. Also, by quantifying the application level goals, it is possible to compute an optimal rate allocation for a centralized detection system.

Next, we introduce a new concept of erasure coding that optimizes the sampling subsystem to account for losses in the communication medium, while jointly performing compression. This coding strategy makes use of the theoretical underpinnings of compressive sensing to show that oversampling, when done correctly, is an inexpensive mechanism to protect against lost data. We explain in Section 1.7 and then fully in Chapter 6 under what conditions this oversampling strategy works and show that in specific instances, it is a capacity achieving strategy, in that no other coding method can provide higher performance.

The final section (Section 1.8, detailed in Chapter 7) questions whether computation and signal detection in a transform domain performed locally is better than that performed on the server-side. The main issue with server-side processing is that all the data would need to transmitted to be able to perform the detection there, whereas doing the detection locally would be computationally expensive and would mandate hard fusion algorithms for inferencing. This problem is solved using techniques from compressive sensing, that shift the burden of the computation to the server-side while requiring only a fraction of the data to be sensed and transmitted. We extend this approach to a detection scenario and show that even higher gains are possible.

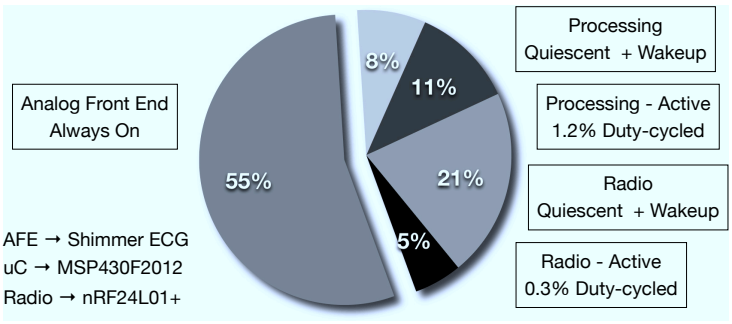

Figure 1.3: Power contribution of sub-systems in a wireless ECG system composed of best-in-class commercial modules.

1.3 Duty Cycling Analog Front Ends

We begin by considering the sampling interface of the sensing information chain for improving the lifetime of monitoring devices. Higher user acceptance of continuous long-term monitors for physiological signals demands devices that are small and unobtrusive. This, in turn, dictates the frequency of replacement or recharge of its power source. Today, continuous monitoring devices, such as the Holter monitor for ambulatory electrocardiography, recommend battery replacement every 2-4 days [OOM, Shi10a], resulting in inconvenience and monitoring discontinuities.

At a high level, a wireless medical monitoring system, such as the Holter monitor, typically consists of three subsystems -- an analog front end (AFE) for signal conditioning, a microprocessor (including A/D converter) for digital signal processing, and a radio that communicates data to an upstream aggregator. Reducing system power, according to Amdahl's Law, entails reducing the power of each subsystem proportionally.

While advances in process technologies improve the speed and power of digital circuits, their analog counterpart have not benefited as much [Mur06a]. They are constrained by electronic noise, loading effects, and accuracy requirements and there are inherent trade-offs between linearity, operating conditions, bandwidth, noise immunity and power dissipation. The analog front end typically involves low noise amplification and filtering for enhancing the dynamic range of the acquired signal and its performance is crucial

to the entire downstream signal chain. Reducing the power consumption of the AFE, therefore, must not come at the expense of deteriorating its characteristics.

Digital logic is aggressively duty cycled, exploiting the relatively large periods of inactivity between sample acquisition, data processing, and communication. Figure 1.3 illustrates the breakdown in average power consumption of a representative wireless ambulatory electrocardiography (ECG) system composed from ''best-in-class'' commercially mature subsystem modules. While the always-on active power consumption of the microprocessor and radio is much higher than the AFE (see Table 2.1), the extent of duty-cycling reduces the average power to a tiny fraction, even when accounting for sleep power and wake-up latency.

Duty Cycling the AFE Physiological signals are low bandwidth, allowing low sampling and transmission rates. For Figure 1.3, for example, these are 256 samples/sec and 1 packet/sec respectively. Furthermore, medical monitoring devices are primarily transmitters so do not expend much radio receive energy. AFE circuits, however, cannot be duty cycled effectively. The key reason being that front end circuits are meant for signal conditioning, which usually involves analog filters with large time constants. While the essential building blocks of analog design, the operational amplifiers (or their discrete transistor counterparts) stabilize relatively rapidly upon applying power, the time constants of conventional analog filters mandate a long wake-up latency to reach their designed performance characteristics. Since the time constant of the filter is intrinsically linked to the filter response, reducing the wake-up latency by reducing the time constant is unacceptable.

For diagnostic quality ECG, the spectrum extends as low as 0.05Hz [Bra00], which mandates a filter time constant of 3.18s. Figure 1.4 plots the actual wake-up latency measured from a Shimmer ECG analog front end [Shi10b]. The signal stabilizes after about 6s.

Two observations guided the circuit design of an AFE that supports duty cycling.

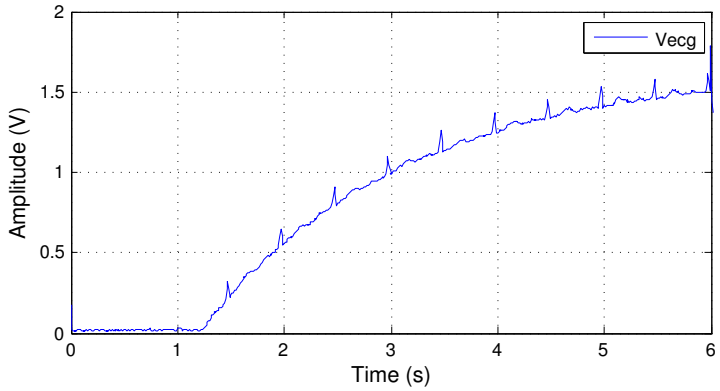

Figure 1.4: Wake up latency of the Shimmer ECG analog front end is about 6s from power up.

First, duty cycling involves power gating circuits repetitively, rather than in a one-off manner. Second, filters are constructed from capacitors (C) and inductors (L), which are ‘‘storage’’ circuit elements whose I/V characteristics shape the frequency corners of the filter. Specifically, the state of an R-C filter is stored in the electric field within the capacitor and similarly, the state of an R-L filter is stored in the magnetic field about the inductor. If filters ‘‘remember’’ what state they were in before power was removed and re-establish that state when they are turned back on, the stored charge will approximate continuous operation.

Our solution to making filters retain their state is straightforward: disconnect the elements with memory from the circuit before power is removed and insert them back in after power is re-applied. We do this using a ‘‘flying-capacitor’’ circuit topology: a pair of analog switches are inserted at the terminals of each capacitor and inductor in the filter circuit. The switches are turned on just after power is applied and are turned off just before power is removed.

By employing this architectural addition to the filter structure, it is possible to eliminate most, if not all, of the *restart* latency mandated by filter blocks, both active and passive, without significantly affecting the filter's response. Specifically, in Chapter 2 (reproduced from [CFS11]), we show that:

- The restart delay of analog filters can be reduced by retaining the state of their memory holding elements across power gating cycles.

- Filters can be made to ''remember'' their state by creating a memory element using a ''flying capacitor'' in the filter circuit. We provide an analysis on the requirements of the switches and of duty-cycle parameters. With a multi-stage commercial off-the-shelf ECG platform, we show that the restart latency can be reduced by three orders of magnitude from 6s to 5ms.

- Our modified platform can be used for energy efficient QRS detection and extraction and we illustrate how this leads to a 3× reduction in the analog front end's power consumption.

1.4 Low Power Compressed Sensing in Hardware

In this section, we introduce a new energy efficient front end design that makes use of analog domain compression. It is well known that many physiological signals follow specific patterns and models, which can be exploited to parametrize and compress sensed data. This compression step, however, is traditionally performed in the digital domain only after the data has already been sampled at the Nyquist rate. Nyquist rate sampling is rather wasteful for compressible signals since their information content may be vastly lower than that assumed by the Nyquist criterion. Furthermore, it is not always practical to perform compression at low power sensing sites and the data might need to be transported uncompressed (also wasteful!) to a high powered collection center.

Compressed sensing is a recent breakthrough in signal processing that allows one to acquire a compressible signal at much below its Nyquist rate [CRT06b]. The Shannon-Nyquist theorem provides sufficient conditions for recovering a periodically sampled signal based on its bandwidth but falls short when trying to include knowledge of other signal characteristics. Compressed sensing (CS) is a mathematical tool that can utilize *a priori*

knowledge of the sparseness or compressibility of a signal of interest within a model framework to acquire a signal at essentially its ''information rate''. The basic tenet of CS is that if a signal is known to be sparse (contains few non-zero values) or compressible (values decay quickly to zero) in a known domain, it is wasteful to sample the signal at the Nyquist rate because most of the data will be thrown away subsequent to compression.

There are three aspects to implementing CS -- knowing the domain in which the signal is sparse, low dimensional signal acquisition and the recovery process. The sparse domain is a transform space, typically a set of linear functions, that facilitates a compressed representation of the input signal. That is, when the signal is transformed from its native domain (say, time or space) to the sparse domain (say, frequency or wavelets), a few values in the sparse domain may be sufficient to describe the signal with high fidelity. This compaction property is the cornerstone of compressed sensing. As examples, audio is known to be compressible in the frequency domain and natural images are compressible in the wavelet domain.

Signal acquisition in CS involves projecting the signal to a lower dimensional domain that is ''incoherent'' to the sparse domain *before* sampling [CR07]. This incoherent projection may be viewed as an information preserving transform that, ideally, maximizes the information collected about the signal in each sample. The fact that this incoherent sampling domain is of lower dimensionality yields compression. The projection process generates a set of compressed measurements and can be accomplished at relatively low complexity, making CS an attractive alternative for energy efficient sensing. Chapter 3 describes in detail our design for a low power hardware implementation of this projection process.

The recovery process exploits the fact that there are few information bearing components in the sparse domain in order to identify them. Generally speaking, it looks for the most compact (sparsest) solution that meets the constraints set by the compressed measurements. The quality of reconstruction depends approximately on the ratio of the number of compressed measurements acquired to the number of information bearing (non-

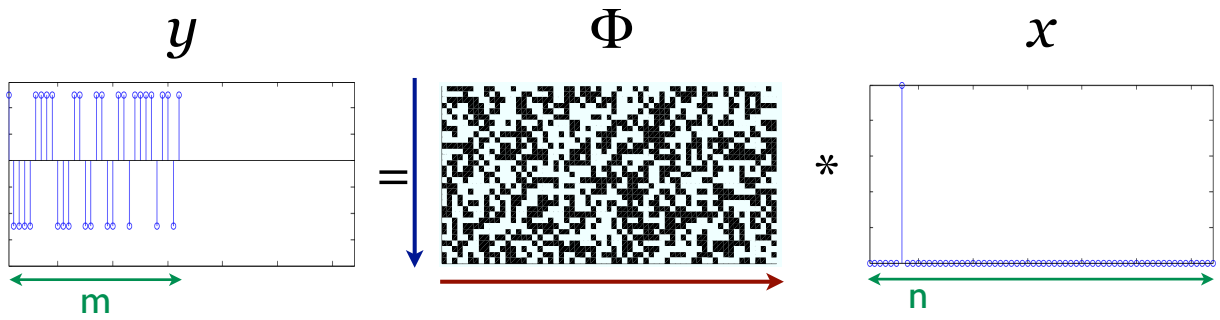

Figure 1.5: Visual representation of compressed sensing through pseudo-random projections from a $+1/-1$ (black/white) Bernoulli distribution. The measurement vector $y \in \mathbb{R}^m$ is computed by projecting the sparse input vector $x \in \mathbb{R}^n$ using a sensing matrix $\Phi \in \mathbb{R}^{m \times n}$ by $y = \Phi x$.

zero) components. The catch, however, is that the recovery process is computationally intensive. A number of recovery algorithms exist that each trade off computation for accuracy differently [CWB08, TG07, CDS98].

CS could be viewed as an asymmetric compression scheme with economical encoding but expensive recovery. This makes compressed sensing particularly well suited for low power physical sensing, where sensor devices are highly constrained, both in energy and computational resources. It is expected that the compressed domain samples would be delivered to a capable base-station or backend for signal recovery or inferencing.

Sampling Compressively Returning to signal acquisition, the key intuition behind CS's incoherent sampling strategy is that of spreading information content in the signal vector (with respect to the sparse domain) across the compressed domain samples. In some sense, one could view each compressed sample as providing an independent summary of the input signal. Researchers have identified that domains constructed from certain random distributions have high incoherence with virtually any sparsity inducing basis with high probability. This is termed the universality property of compressed sensing [CT06]. Figure 1.5 shows a visual representation of a conceptual CS acquisition. In this case, we assume that the input signal, x, shown rightmost, is sparse in the time domain with an exaggerated sparsity of just one non-zero component. The sampling matrix, Φ,

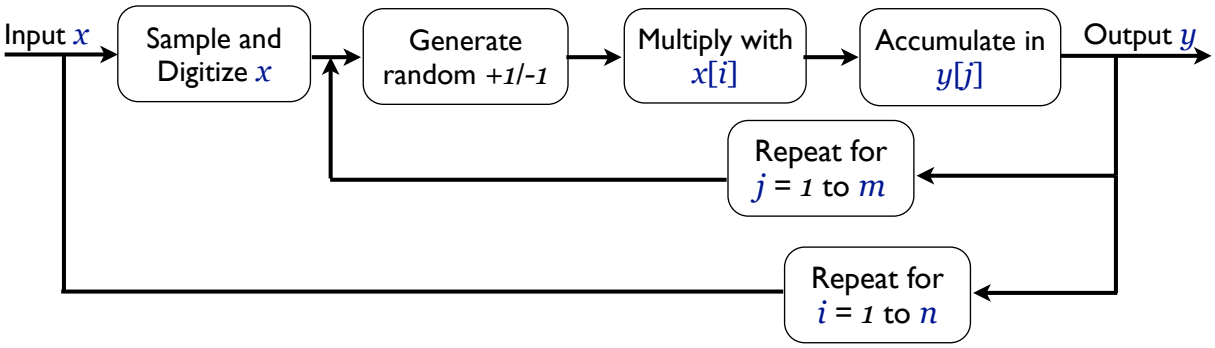

Figure 1.6: Software implementation of Compressed Sensing needs pre-digitized data and $\mathcal{O}(mn)$ operations.

is generated from a random $+1/-1$ (black/white) Bernoulli distribution, i.e., we toss an unbiased coin for each element of the matrix. The measurement vector, y, smaller in dimensionality than x, is computed through $y = \Phi x$ and constitutes the compressed domain samples we seek.

The effect of using a random kernel is that regardless of where the information bearing components lie in the sparse domain, there is a statistical chance that the sampling matrix captures it. Observe, for example in Figure 1.5, how one non-zero value in x manifests itself in each of the compressed measurements in y. It must be mentioned that while randomized projections are not the most efficient[1], the flexibility afforded by the universality property is overwhelmingly attractive.

Although CS holds promise for many applications, the need for incoherent projections prior to sampling has limited its direct impact to instances where the sampling domain is inherently incoherent to the signal of interest. For example, this holds for magnetic resonance imaging (MRI), where sampling is performed in the frequency domain while the signal being recovered is an image in the spatial domain [LDP07]. In other sensing applications such as EEG (electroencephalography) or ECG (electrocardiography) monitoring, where sampling is traditionally performed in the time domain, the incoherent projections have to be applied explicitly [KMK11]. Figure 1.6 depicts such a software approach to

[1]Ideally, the rows of the measurement matrix should be orthogonal. Randomly generated matrices only have weak orthogonality guarantees [LKR12].

CS. The inner loop performs m row-wise multiply-and-accumulate (blue downward arrow in Figure 1.5) and the outer loop executes across columns for every sample acquired (red rightward arrow in Figure 1.5). The output y is read every n samples and the accumulators are then reset.

There are three practical issues with this software based CS technique. First, it requires that the signal be explicitly sampled and digitized before projections can be computed. Since CS was developed as a solution to avoid unnecessary sampling, while viable in some instances [KMK11], this technique does not fully exploit the advantages that compressed sensing offers. Second, an order of $\mathcal{O}(mn)$ explicit mathematical operations are typically required to compute the projection. The computational cost of explicit compression for some transform domains may actually be lower ($\mathcal{O}(n\log n)$ for FFT), although software CS operations are simpler (add/subtract). Third, since samples are digitized prior to the projection, quantization error accumulates as n increases. That is, as the compression ratio (n/m) increases, so does the effect of quantization [SBB06].

In Chapter 3, we propose a novel hardware approach to compressive sampling. Our system, called CapMux, combines digital logic with a custom analog front end to realize randomized projections at very low power. CapMux takes as input a time domain signal of variable duration (i.e. variable n) and produces a fixed number of compressed domain analog measurements (i.e. fixed m) over that duration. These compressed measurements can be fed directly to an analog-to-digital converter to be digitized, transported and processed for signal recovery or inference. The key idea that makes our architecture low power is being able to amortize the quiescent current consumption costs of high performance analog components through time-multiplexing. CapMux is constructed from just one active analog signal processing chain that can be shared across an arbitrary number of channels. CapMux not only leads to a lower average power per measurement channel, but also admits low complexity scaling.

Chapter 3 (reproduced from [CMS12]) provides the following details:

- It introduces CapMux, a scalable architecture for low power compressed sensing. CS involves projecting or convolving a signal with a set of random basis vectors. In analog circuit terms, this can be achieved by multiplying the signal with each random vector independently and simultaneously and integrating (or low pass filtering) the result. Conventionally, this would require as many analog processing chains as measurement channels desired [KLW06]. CapMux, on the other hand, shares access to a single modified analog processing chain by time multiplexing its use. Our architecture hinges on the design of a multi-channel integrator that saves its state while switching time slots. The idea was inspired from a four decade old technique [BP72] used to construct higher order active filters from a single operational amplifier.

- It described the realization of a 16 channel CapMux prototype designed from commercially available components. Random vector generation and time slot synchronization for the multi-integrator is orchestrated by a micro-controller that also contains the ADC to sample the compressed measurements. We describe calibration routines for managing component variation and the handling of parasitic capacitances that become significant due to the large number of channel switches. We also outline an extension to a larger 64 channel board.

- It evaluates the prototype implementation for signals sparse in the time, frequency and wavelet domains. We characterize the board in terms of its signal recovery quality in these sparse domains and show how various tunable parameters affect its performance. The analog front end consumes less than 20μA and yields recovery in excess of 30dB SNR consistently.

1.5 Compressively Acquiring Neural Action Potentials

Moving on to the processing element of the sensing information chain, we explore compressed sensing to lower communication cost for a wireless neural recorder. Neurons communicate with each other using electrical signals called ‘‘action potentials’’ or ‘‘spikes’’.

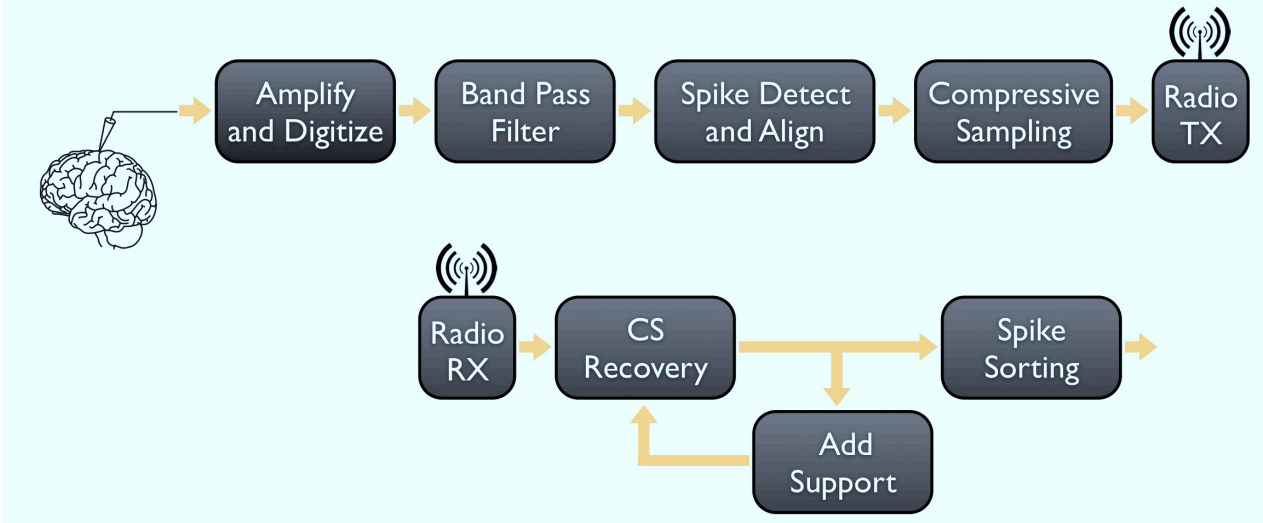

Figure 1.7: Schematic representation of our proposed compressive wireless neural recording system, with the top half *in vivo* and the bottom half *ex vivo*.

Many neuroscience experiments and clinical studies record spikes using implanted electrode arrays. In a traditional neural recording system, the acquired signals are transmitted outside the brain using wires, but this limits the freedom of movement of the subject, increases the risk of infection, and leads to motion artifacts in the recordings. Wireless neural recording systems, either implanted inside the brain or attached to the scalp, promise to circumvent these issues but are subject to stringent power constraints.

The power consumption of a wireless neural recording system that transmits raw data is typically dominated by the radio transmitter. Because the power consumed by a given radio is directly proportional to the transmission rate, the data rate must be lowered in order to reduce system power. Spike sorting, the process of assigning recorded action potentials to their source neurons, is a common analysis performed on acquired data. Spike sorting exploits the fact that action potentials from distinct neurons have unique patterns or morphologies. Based on learning and clustering the spike waveforms, action potentials from multiple neurons can be distinguished. Spike sorting is computationally and memory intensive and is therefore performed *ex vivo* at an upstream data aggregator.

As described in Section 1.4, compressive sensing (CS) is a recently developed theory that enables signal reconstruction from a small number of non-adaptively acquired sample

measurements corresponding to the information content of the signal rather than to its bandwidth [CT05]. Information content or sparsity is quantified by estimating the number of the significant coefficients when the signal is projected into a space that accentuates its principal components. Therefore, if action potentials are sparse, compressive sensing would allow us to reduce communication costs and bandwidth compared to transmitting raw action potentials acquired at the Nyquist rate. Figure 1.7 depicts a schematic diagram of our proposed compressive neural recording system. Our implanted device would perform bandpass filtering, spike detection, and alignment on-chip to extract the action potential waveforms. These spike waveform windows are then sequentially coded through a compressive sensing block and transmitted using a low-power radio.

In Chapter 4, reproduced from [CKG11], we introduce a ''learned union of supports'' for spike recovery to enhance sparsity in the reconstruction. We observed that action potentials from different neurons have subtly different sparsity patterns or supports, and that supports of spikes from the same neuron are very similar. Recovering the signal using a weighted basis pursuit as done in Section 1.8 [CWB08, LV10a, CKZ09b], where the indices of the weights are a union of the support sets of previously recovered spikes, results in higher signal quality. We will explain in Section 4.3 why this ensues. Specifically, we will demonstrate that it allows us to achieve up to 9.5 dB higher SNDR (signal to noise-plus-distortion ratio) on real neural datasets when compared to conventional basis pursuit for the same compression ratio but with receiver side modifications only. We analyze the power consumption of the wireless neural recording system and show that compressed sensing can provide high-quality reconstructions of the spike morphology at a nominal increase in power when compared to sending only features for spike sorting [KGM09]. We verify that classification accuracies of up to 90% can be obtained by sending just 15 CS measurements per spike, which corresponds to a 60$\times$ data-rate reduction when compared to raw data transmission and a 3.2$\times$ reduction compared to raw action potential transmission.

1.6 Interference Aware Sampling

In a wireless sensing environment, when nodes forward their information to a fusion node (e.g. physiological sensors sending data to a smart phone) for application level processing, there is high likelihood that neighboring nodes interfere with one another, especially when the sampling and transmission rates are high. In this scenario it is paramount that data from nodes be prioritized such that only information that improves detection performance is transmitted. This may be interpreted as saying that it is the Quality of Information (QoI) delivered to the end user that is of primary interest.

In general, measurements from different sensor nodes do not contribute equally to the QoI because of differing sensing modalities, node locations, noise levels, sensing channel conditions, fault status, and physical process dynamics. In addition, metrics of QoI are highly application dependent, such as probability of detection of an event or fidelity of reconstruction of a spatiotemporal process. Despite these considerations, traditional data collection and dissemination schemes have maintained that ''all bits are created equal'' and thus data from different source nodes are handled equitably within the network. Resultantly, network protocols have been designed with a focus on metrics such as throughput, packet delivery ratio, latency, and fair division of bandwidth, and are thus oblivious to the importance and quality of sensor data and the target application.

We argue that sampling and transmission protocols for sensing applications need to be cognizant of and use feedback from the sensor fusion algorithms to explicitly optimize for application-relevant QoI metrics during network resource allocation decisions. For traffic engineering in computer networks, researchers introduced the concept of Quality of Service (QoS) as a means of labeling and prioritizing data flows according to a set of static predefined policies to ensure that data from the most important source or flow gets preferential treatment. For physiological sensing, however, this definition of QoS is inadequate because the notion of importance is associated not with a source, but with the data itself and moreover, on the *value* of the data to the inferencing application.

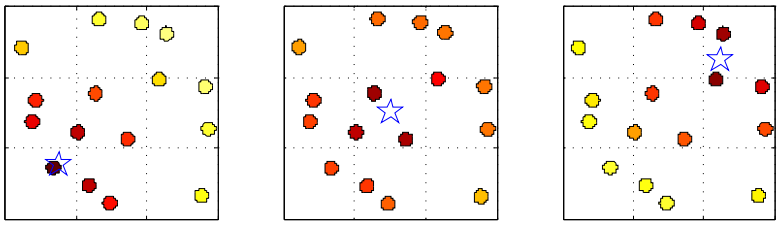

Figure 1.8: An example of foveated sensing. Sensors (dots) closer to the event (blue star) send out more data (darker shades) than ones further away.

One may interpret this as saying that the importance of a source node and its resultant share of network resources should now depend on the *quality* of data currently being produced (sensed) by the node. We first attempt to develop an intuitive appreciation of this concept.

Foveated Sensing Consider the following example in an outdoor setting: we wish to track a moving target in a field. The field is instrumented with a set of acoustic sensors [GMP] for object tracking, say. To improve tracking accuracy a heuristic could be applied: once a target is detected and is being actively tracked, *nodes closer to the target should be able to send more data than ones further away from* it. Intuitively, one would expect that nodes closer to the event have access to better quality data and funneling more data from these nodes would improve the accuracy at the fusion center. Conversely, dropping data from distant nodes under congestion would not affect the end result significantly. However, distant nodes should not be starved so that new events could be detected too. This scheme is depicted pictorially in Figure 1.8. In effect, the heuristic dynamically changes the priority of nodes' transmissions based on an estimate of the event location (one could go further to predict direction and preemptively control rates as well). A mechanism akin to this, termed foveated sensing, occurs in the human vision system to focus our attention on the most salient objects in our field of view [Fri06].

Quantifying QoI In Chapter 5, reproduced from [CZK09b], we illustrate a mechanism of quantifying and utilizing QoI for allocating network resources based on the salience

of each node. This in turn directly reflects the synchronous sampling rate allocated to each node in a distributed sensing environment. We develop this technique in the context of a centralized event detection scenario where nodes sense and transmit information in a multi-hop fashion through relay-forwarding to a fusion center. This act of wireless transmissions may result in interference, especially when nodes are transmitting near the network link capacity.

To solve this problem, we develop a mathematical formulation that involves maximizing a convex relationship between rates from each sensing node and the detection performance. This relationship is derived in detail in Section 5.3.1 using results from binary hypothesis testing. In particular, we find that the rate of each node is decided by the signal-to-noise ratio (SNR) each node contributes to the fusion algorithm at the sink. This means that a node that is either too far away from the event or is vulnerable to external noise sources will be given a lower salience score.

Then, using knowledge of the network topology, we construct a set of constraints that ensures that if the network transmissions are appropriately scheduled or they follow an efficient contention resolution protocol, that network capacity will be attained without interference. This is achieved easily by considering the hidden terminal and exposed terminal problems in wireless MAC design. Specifically, it is seen that when a particular node is transmitting, the node's next hop (which relays traffic toward the sink) cannot simultaneously transmit since it is receiving and radios are half-duplex, and none of the next hop's neighbors can transmit (or they would interfere). Apart from this set of interferers, any other node can transmit simultaneously. Using this guideline, a set of interference constraints are created.

As a final step, the convex formula is plugged into a solver along with the interference constraints to compute a set of rates that optimizes detection performance. Since network topology may not be known in advance or the sensing nodes may be mobile themselves, we outline a practical network feedback protocol that seeks to allocate the sampling and transmission rates optimally. In our evaluation of this methodology, we show that the

feedback based protocol does achieve near-optimal rates without any prior knowledge of the network topology.

We also demonstrate through analysis and simulation that allocating sampling rates in a QoI-aware manner delivers substantial application-level performance benefits, especially as networks grow because our careful rate selection shifts the bottleneck link away from the sink, allowing the ''best'' nodes to participate more effectively. A fortunate side effect of this is that it relieves nodes closer to the sink, improving mean network lifetime. In conclusion, the philosophy behind our approach is similar to recent efforts in Content Centric Networking [CH] that endow the networking stack with knowledge of the intent of the communication transaction. The difference is that our sampling strategy is not only content-aware, but is also cognizant of the *effect* the content has on the application.

1.7 Overcoming Channel Erasures Through Sampling

Having handled the case of allocating rates for multiple sensors according to their contribution to the quality of inference, we still have to resolve the issue of handling data loss. Data loss in wireless sensor systems is inevitable, either due to exogenous (such as transmission medium impediments) or endogenous (such as faulty sensors) causes. While there have been many attempts at coping with this issue, compressive sensing (CS) enables a new perspective. Since many natural signals are compressible, it is possible to employ CS, not only to compress the data to reduce the effective sampling rate, but also to improve the robustness of the system to channel erasures. This is possible because reconstruction algorithms for compressively sampled signals are not hampered by the stochastic nature of wireless link disturbances and sensor malfunctions, which has traditionally plagued attempts at proactively handling the effects of these errors.

Realistically, data losses creep into the system owing to two inevitable circumstances -- wireless link quality variations because of noise and interference and temporary sensor faults. To cope with these issues, reactive schemes like retransmissions (end-to-end or

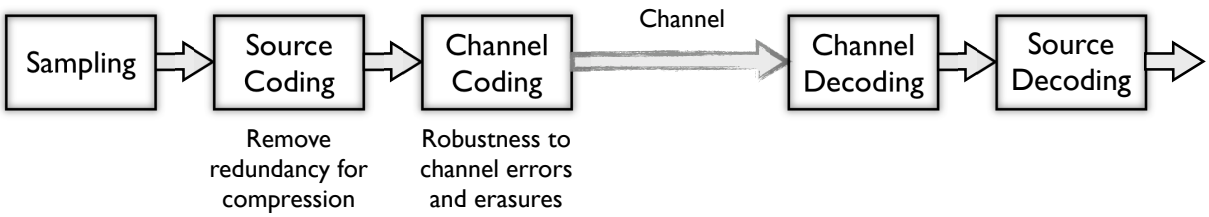

Figure 1.9: Channel coding is applied typically after source compression, if any, to protect the data against errors and erasures in the communication channel.

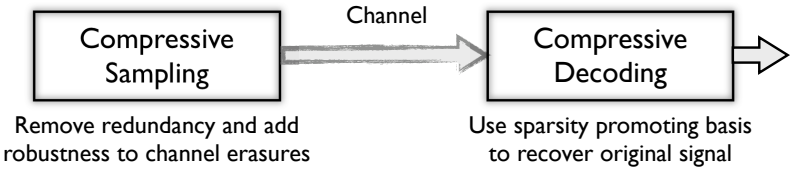

Figure 1.10: Compressive sampling can be used as a joint source-erasure coding strategy even in the presence of extreme channel losses.

hop-by-hop) have been popularly employed, but are inefficient in many situations and inappropriate in others. Retransmission may be ineffective, for example, when data is logged and communicated in bulk for later analysis [MLT08] or stored in Flash [OL08]. Proactive schemes such as error correcting codes have also been used, though in a limited sense, due to their high computational complexity, both at the encoder as well as the decoder.

In [CZK09a], we introduced the idea of using compressive censing (CS) [CT06] as a low-encoding complexity, proactive sensing approach for robustness to even extreme data losses. Utilizing the fact that CS strategies make inherent use of randomness (recall random projections from Section 1.4) within the sensing process, we surmise that data lost through the stochastic nature of an erasure channel is indistinguishable from an *a priori* lower sensing rate at the fusion center. We verify this conjecture empirically and show that it is sufficient to proactively increase the sampling rate in order to maintain reconstruction accuracy.

The traditional view of applying source coding followed by channel coding is depicted in Figure 1.9, where sampling is succeeded by the compression routine, lossless or lossy

in nature, to remove redundancy in the data. This step is performed at the application layer and utilizes prior knowledge of signal characteristics to determine the most compact representation for the signal of interest. After compression, the data is handed over to the communication block, where just before transmission, typically at the physical layer of the communication stack, the data is encoded to introduce a controlled amount of redundancy. If transmitted symbols are received in error or not at all through the channel, the decoder may be able to recover the original data using this extra information. The channel decoder uses a forward error correction mechanism and an error detection mechanism, like the CRC (cyclic-redundancy-check), to ensure that it decoded the data from the sensing node correctly. Beyond this, the data is passed on to the source decoding routine which reverses the compression process to recreate the originally sensed signal.

On the other hand, if one were to employ compressive sensing for joint source and channel coding, the sampling stage would itself subsume all the coding blocks. The CS sampling block uses one of a variety of different sampling techniques that ensure that sufficient unique information is captured within the sampling process. We propose that the CS sampling block should be designed now, not just to include prior knowledge of signal characteristics in terms of its sparsity in a specific domain, but should also consider channel characteristics and tune the sampling process to improve the robustness to channel impairments.

The Restricted Isometry Property To see why CS can be used as an effective erasure coding technique, we briefly introduce the concept of the restricted isometry property. First, we label some terms that we will require. The problem we seek to address is optimizing the acquisition of an n-length vector $f \in \mathbb{R}^n$ at a sensor node such that it can be recovered accurately at a base station one or more wireless hops away. An assumption for applying CS is that the original signal vector f is representable in some domain Ψ as a sparse or compressed vector, $x \in \mathbb{R}^n$, which has few non-zero values. We can represent f equivalently then as $f = \Psi x$, where $\Psi \in \mathbb{C}^{n \times n}$ represents an orthonormal

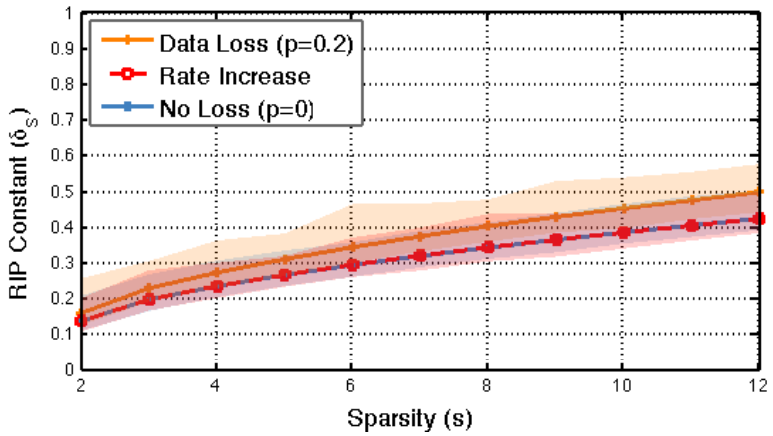

Figure 1.11: Effect of data loss on RIP constant with average loss probability p in a memoryless Bernoulli channel. Also shown is the improvement in RIP constant by increasing rate to $k/(1-p)$. Shading indicates the min-max across 1000 Monte-Carlo runs.

basis such as the inverse Fourier basis ($\psi_{\omega,j} = \frac{1}{\sqrt{n}}\exp(\mathrm{i}2\pi\omega j/n)$). The signal f is then acquired in its natural domain by projecting it through a sensing matrix $\Phi \in \mathbb{R}^{k\times n}$ to generate k measurements $z = \Phi f = \Phi\Psi x = Ax$.

Intuitively, the restricted isometry property [Can08] for the CS matrix A (defined in Chapter 6) quantifies how well the sampling process has captured the signal through the associated RIP constant δ_s (defined in Equation 1) . A good sampling matrix Φ + compression matrix Ψ combination would have a RIP constant close to zero. By increasing the number of measurements k, δ_s can be made arbitrarily small.

It has been shown in [CT06] that the RIP is satisfied when the matrix A is constructed randomly using an *i.i.d.* Gaussian r.v. such that $A_{ij} = \mathcal{N}(0, \frac{1}{n})$ or an equiprobable $\pm\frac{1}{\sqrt{n}}$ Bernoulli r.v. It was also shown in [CT06] that A can be constructed by randomly selecting the rows of a Fourier basis matrix, such as Ψ. This implies then, that Φ is essentially a $k \times n$ random sampling matrix constructed by selecting k rows independently and uniformly from an $n \times n$ identity matrix I_n.

Handling Data Losses We can model the erasures introduced by the transmission channel with an average loss probability, $p \in [0, 1]$. We assume an independent Bernoulli

process so that the probability of any packet (sample) being dropped is equal to p. Now, since $k \cdot p$ packets would have been dropped through the channel, it is bound to affect the CS reconstruction performance. This can be quantified by computing the RIP constant of $A' = CA = C\Phi\Psi$, where C is an $k \cdot (1-p) \times k$ matrix (constructed from I_m) that enumerates which samples got delivered.

As a measure of protecting against this performance loss, we increase the sensing rate such that we collect $\bar{k}$ samples from the sensor node instead of k, where $\bar{k} = k/(1-p)$. We construct an equivalent $\bar{\Phi}$ matrix of size $\bar{k} \times n$ from these samples and the CS sensing matrix $\bar{A} = \bar{\Phi}\Psi$. If now these data samples are communicated through a lossy channel, the resultant sensing matrix will be $\bar{A}' = \bar{C}\bar{A} = \bar{C}\bar{\Phi}\Psi$. We can then expect that, if $\delta_s(\bar{A}') \leq \delta_s(A)$, the performance of CS reconstruction after losses will be equivalent to that of the original reconstruction.

We compute $\delta_s(A)$ and $\delta_s(\bar{A}')$ using Definition 1 for 1000 randomly generated 256×1024 Fourier random sampling matrices as shown in Figure 1.11. The solid blue curve labeled ''No Loss'' indicates $\delta_s(A)$. The shading illustrates the min-max values over all Φ. With loss probabilities $p = 0.2$, we see an increase in RIP constant $\delta_s(A')$ for the data loss cases. Also the variation around the mean is larger for $\delta_s(A')$.

It can be shown that the probability distributions of time-stamps extracted from Φ and $\bar{\Phi}' = \bar{C}\bar{\Phi}$ are identical when $\bar{C}$ comes from an independent Bernoulli channel. This means that losses due to the channel are indistinguishable from an *a priori* reduced random sampling rate at the sensor node. This in turn means that, if the channel is not congested, increasing the sensing rate by a factor of $p/(1-p)$ will restore the delivery rate to k. The effect of this increase is substantiated in Figure 1.11 and establishes $\delta_s(\bar{A}') \approx \delta_s(A)$ for the Bernoulli channel.

In summary, we have explored the application of Compressive Sensing to handling data loss from erasure channels by viewing it as a low encoding cost, proactive, error correction scheme. We employed the RIP to illustrate that for a memoryless channel even extreme stochasticity in losses can be handled cheaply and effectively. In Chapter

6 (reproduced from [CCZ10]), we go into further detail and evaluate this coding strategy with a more realistic bursty channel as well as other sampling matrices.

1.8 Energy Efficient Compressive Detection

Finally, we focus on the inferencing stage of the sensing chain and show how compression, when applied in the course of sampling, can reduce the data rate required by the inference engine without adding a significant energy burden to the sampling stage. Since the primary use for physiological sensing has been in continuous long term monitoring, extreme energy constraints come to fore since these platforms are typically battery operated [CCC08, DGA05, DBH08, BISa]. Achieving high system lifetime, therefore, requires a concerted effort in reducing the sensor sampling, processing and radio communication costs while maintaining application level objectives. One way communication cost can be reduced is through compressing the acquired signal before transmission by exploiting any redundancy within it.

The decision on whether to compress or transmit (without compression) depends heavily on the compression algorithm used and the relative energy cost of compression versus transmission [HCB00]. As a general rule, it is better to compress when the node is further away from the sink (in terms of hops) than for nodes close by [NTG]. This is because compressing before multi-hop transport saves energy at all the relay nodes in the network. However, making this decision is difficult especially in a mobile environment since the network topology information must again be available to all nodes in the system.

An alternative to the traditional form of source compression, which is typically performed after the signal is completely acquired, is compressive sensing [CRT06a, CDS98] (CS), introduced in Section 1.4. CS suggests that if the signal is sparse or compressible, the sampling process can itself be designed so as to acquire only essential information. CS enables signal acquisition with average sampling rates far below the Nyquist requirement and eliminates the explicit compression step altogether. This not only saves energy in the

sampling subsystem through reduced sampling, the *processing subsystem* through reduced complexity (no explicit compression), and the *communication subsystem* through reduced transmission, but also enables the capture of substantially more complex signals where it would not be possible otherwise. For example, in applications interested in high-frequency signals, low power embedded sensing devices just can not keep up at Nyquist sampling rates [AGN].

Compressive sensing involves taking sample measurements in an ''incoherent'' domain through a linear transformation [BDD08]. This step may be viewed computationally equivalent to compression if this transformation sparsifies the signal. However, the key insight underpinning CS mechanisms is that, though the incoherent domain does not sparsify the signal directly, it describes the signal sufficiently uniquely for perfect recovery to succeed from a fraction of measurements. The computational advantage of doing this comes from the fact that some incoherent transformations can be done implicitly and cheaply at the source. To achieve this, however, the designer needs to (a) fabricate a domain that is incoherent with the sparsifying one and (b) transform the signal to it through sampling. Researchers have shown [BDD08], quite remarkably, that taking appropriate random projections of the signal before sampling satisfies both these requirements adequately for a large class of compressible signals.

The issue with applying CS in embedded systems is that while acquisition is cheap, reconstruction algorithms are computationally severe. Interestingly, it is this asymmetric architecture that makes CS an excellent choice for low-power physiological sensing. This is because medical sensing usually includes a back-end data collection and fusion center (where the condition is ultimately inferred) that is endowed with a considerable amount of computing and storage ability. This means that, if the sensing nodes are able to take random projections of the sampled signal and communicate them to the fusion center, it is possible to reconstruct the signal with high probability using a fraction of what the Nyquist rate would have required.

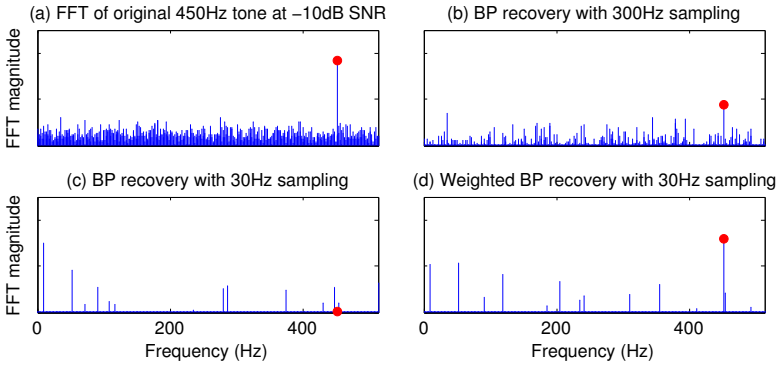

Figure 1.12: Frequency (FFT) coefficients for the CS reconstruction of a 450Hz tone at -10dB SNR with different sampling rates and recovery strategies.

Compressive Detection While many CS mechanisms have focused on signal reconstruction, some researchers [HN07, DDW07, DBH08] have found that the number of samples needed to reliably detect features in the signal, even in a noisy and interference prone environment, can be considerably lower if full CS recovery is not required. This is done by utilizing knowledge of *where* the event may be present in the sparse domain. For example, if the event of interest is a signature of known support, recovery algorithms would look for the signal at those support indices with higher priority.

A technique often used for full reconstruction using compressive sensing is called Basis Pursuit (BP), which poses the search for the sparse solution as a linear programming problem. There are two drawbacks to using BP directly for detection, however. First, though BP can complete in polynomial time, the computational requirement is very high [TG07]. And second, since BP attempts to reconstruct the signal completely, the number of measurements and hence energy required for comparable detection performance may be excessive too. We tailor BP's linear programming problem to include prior knowledge of the event signature. This is done by biasing components of the solution through a weighting matrix (details in Section 7.2, reproduced from [CKZ09b]) that prioritizes the search to prefer solutions with the known support indices. The effect of this weighting procedure is that the biased components 'stand out' because they are artificially enhanced against background noise.

Our proposed Weighted Basis Pursuit (WBP) is visually depicted in Figure 1.12 for the detection of a sinusoidal tone at 450 Hz in the presence of white noise. The reconstruction is performed in the Fourier (frequency) domain from randomly collected samples at different rates. When no weighting is applied, the average sampling rate needs to be as high as 300Hz to detect the tone -- the red dot in Figure 1.12b is just above the noise floor. While this is below the Nyquist rate of 900Hz, the gains are not impressive. If the sampling rate is lowered to 30Hz, no detection is possible (1.12c). However, if weighting is applied, the frequency tones immediately stand out (1.12d), implying a near 30× benefit over the Nyquist rate. A detailed evaluation of both simulated and experimental performance for different sampling rates in various noisy environments is deferred until Section 7.4.

Implementing Compressive Detection Perhaps the most important aspect of implementing CS is the random linear transformation for sampling. Note that this transformation must not only be incoherent with the domain in which the signal is sparse, but it must also be substantially cheaper to implement than explicit compression. Much research has been undertaken in the CS community to search for suitable pseudo-random transforms but most require a form of additional front-end hardware before the ADC similar to CapMux (Chapter 3) or some software oriented techniques that assume Nyquist sampling once more. A key contribution we have made is a demonstration of compressed detection mechanisms on commercial sensor nodes without additional hardware. To achieve this, we use a uniform random sampling procedure that is known to be incoherent with any orthogonal basis [RV06], such as the Fourier basis. However, this random sampling is inherently non-causal and may also violate ADC hold times. In Section 7.3, we show how both these limitations can be overcome effectively and inexpensively.

In our evaluation of these ideas, we show through simulations and an implementation on sensor nodes that WBP is not only feasible at rates 30× below the Nyquist requirement but that it delivers comparable detection performance with up to 10× increased

energy efficiency. We show that CS based detection is robust not just to Gaussian noise sources but also to non-stationary high- power narrowband interference. WBP always outperforms conventional BP, especially in high SNR regimes. Our empirical study also shows, however, that the computational complexity of good random number generation is non-trivial for these low-power embedded devices and that efficient algorithms for implementing CS are necessary.

CHAPTER 2

Duty Cycling Analog Front End Circuits

2.1 Introduction

We begin our exploration of energy efficiency mechanisms for physiological sensing with the sensor interface. Affordable, wearable, embedded electronic medical sensor systems that enable continuous long term monitoring of physiological signals could revolutionize health care. Collecting data in this manner produces long-run, high-quality datasets, the analysis of which has already demonstrated potential in preventive medicine, stress inferencing [PRH11], and self-care [Dep05]. Realizing this vision requires improvements in both capability and popularity. User acceptance is more readily achieved when these devices are small and unobtrusive. This, in turn, dictates the device's size and the frequency of replacement or recharge of its power source. Today, continuous monitoring devices, such as the Holter monitor for ambulatory electrocardiography, recommend battery replacement every 2-4 days [OOM, Shi10a], resulting in inconvenience and monitoring discontinuities.

The Analog Front End as Energy Bottleneck At a high level, a wireless medical monitoring system, such as the Holter monitor, typically consists of three subsystems -- an analog front end (AFE) for signal conditioning, a microprocessor (including A/D converter) for digital signal processing, and a radio that communicates data to an up-

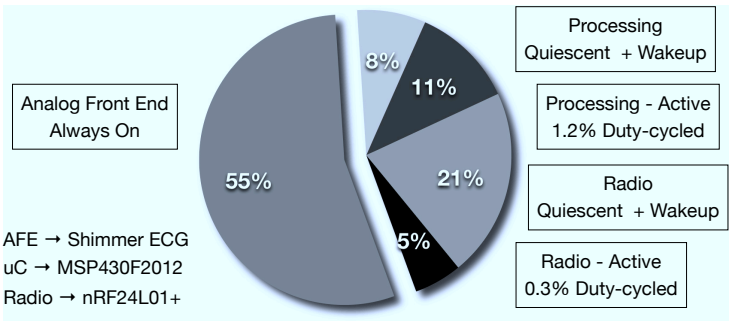

Figure 2.1: Power contribution of sub-systems in a wireless ECG system composed of best-in-class commercial modules. The microprocessor and radio can be aggressively duty cycled, but not the analog front end.

stream aggregator. Reducing system power, according to Amdahl's Law, entails reducing the power of each subsystem proportionally. With constant advances in digital process technologies [Tex11], and more efficient radio architectures appearing on the horizon [Dec11, Qua11], the average power consumption of the latter two components has been steadily dropping.

Analog circuits, on the other hand, have not benefited as dramatically from technology scaling [Mur06a]. They are constrained by electronic noise, loading effects, and accuracy requirements and there are inherent trade-offs between linearity, operating conditions, bandwidth, noise immunity and power dissipation. The analog front end typically involves low noise amplification and filtering for enhancing the dynamic range of the acquired signal and its performance is crucial to the entire downstream signal chain. Reducing the power consumption of the AFE, therefore, must not come at the expense of deteriorating its characteristics.

Advances in process technologies improve both the speed and dynamic power consumption of digital circuits simultaneously -- as transistor sizes shrink, digital circuits do more in less active time with less peak power. It is no surprise then that it is common practice to aggressively duty cycle digital logic exploiting the relatively large periods of inactivity between sample acquisition, data processing, and communication. With adequate buffering, the same strategy is also applicable to radio transmission by power gating

the transceiver between bursts of packets. This is particularly beneficial for applications such as medical monitoring that can tolerate some signal latency.

Figure 2.1 illustrates the breakdown in average power consumption of a representative wireless ambulatory electrocardiography (ECG) system composed, in a vendor-agnostic manner, from ''best-in-class'' commercially mature subsystem modules. While the always-on active power consumption of the microprocessor and radio is much higher than the AFE (see Table 2.1), the extent of duty-cycling reduces the average power to a tiny fraction, even when accounting for sleep power and wake-up latency. The resultant imbalance in power consumption among the subsystems is clear and motivated this effort.

The Need for Duty Cycling The AFE is the highest consumer not just in our COTS based system, but in other reported recording systems that have taken the ASIC route as well [HKC09] (>60%). We argue that, upon scrutiny, many heavily optimized medical monitoring systems will show the same behavior. While this may seem counter-intuitive at first, the prime reason it holds for ECG (and other medical monitoring systems) is that physiological signals are low bandwidth, allowing low sampling and transmission rates. For Figure 2.1, for example, these are 256 samples/sec and 1 packet/sec respectively. Furthermore, medical monitoring devices are primarily transmitters so do not expend much radio receive energy. Coupled with the fact that microprocessors and radios available today have small sleep currents (<1μA) and rapid wake-up (<1ms) while providing high performance (<1nJ/instr. and <50nJ/bit respectively) in their active state, duty-cycling has skewed the proportion of power the analog front end consumes. Left unchecked, this trend is likely to continue. It should be mentioned that the AFE does not include the analog-to-digital conversion, which is usually part of the microprocessor energy budget and is typically around 20-50nJ/conversion [Mur06b].

While processor and radio duty-cycling works exceedingly well, prevalent AFE circuits used in medical devices today cannot be duty cycled effectively. The key reason being that front end circuits are meant for signal conditioning, which usually involves analog

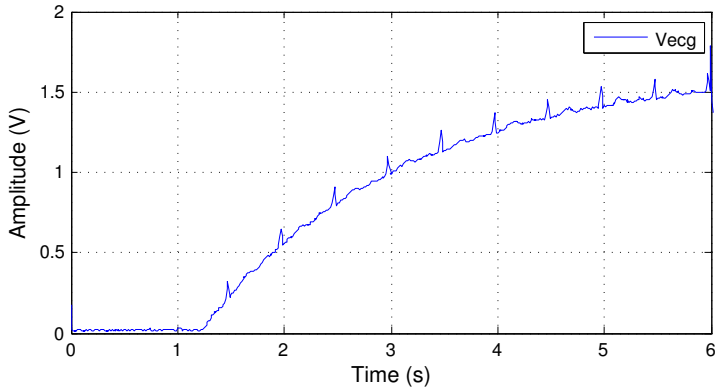

Figure 2.2: Wake up latency of the Shimmer ECG analog front end is about 6s from power up to get to the bias voltage of 1.5V.

filters with large time constants. While the essential building blocks of analog design, the operational amplifiers (or their discrete transistor counterparts) stabilize relatively rapidly upon applying power, the time constants of conventional analog filters mandate a long wake-up latency to reach their designed performance characteristics. Since the time constant of the filter is intrinsically linked to the filter response, reducing the wake-up latency by reducing the time constant is unacceptable.

For diagnostic quality ECG, the spectrum extends as low as f_{HP}=0.05Hz [Bra00], which mandates a filter time constant of $\tau_{HP} = \frac{1}{2\pi f_{HP}}$=3.18s. Figure 2.2 plots the actual wake-up latency measured from a Shimmer ECG analog front end [Shi10b]. The board was connected to a Fluke PS240 patient physiology simulator and the tiny spikes are the R-peaks in the ECG signal. The voltage stabilizes after about 6000ms (about $2 \times \tau_{HP}$) to the bias voltage of the circuit at 1.5V. Consequently, the AFE is left always-on even when there are opportunities for duty cycling.

Duty Cycling the AFE Two observations guided the circuit design of an AFE that supports duty cycling. First, duty cycling involves power gating circuits repetitively, rather than in a one-off manner. This implies that effective duty cycling is not limited by the initial wake-up delay of the circuit when power is first applied, but rather by the restart delay when power is *re-applied* following a short interruption. Second, filters are

constructed from capacitors (C) and inductors (L). Both are ‘‘storage’’ circuit elements whose I/V characteristics are time-dependent resulting in the integration or differentiation that shapes the frequency corners of the filter. Specifically, the state of an R-C filter is stored in the electric field within the capacitor and similarly, the state of an R-L filter is stored in the magnetic field about the inductor. If filters ‘‘remember’’ what state they were in before power was removed and re-establish that state when they are turned back on, the stored charge will approximate continuous operation.

Our solution to making filters retain their state is straightforward: disconnect the elements with memory from the circuit before power is removed and insert them back in after power is re-applied. We do this using a ‘‘flying-capacitor’’ circuit topology: a pair of analog switches are inserted in series at the terminals of each capacitor and inductor in the filter circuit. The switches are turned on just after power is applied and are turned off just before power is removed. Although this idea sounds similar to the well known switched capacitor filter, there are fundamental topological and principle differences, which we explain in Section 2.5.

By employing this architectural addition to the filter structure, it is possible to eliminate most, if not all, of the *restart* latency mandated by filter blocks, both active and passive, without significantly affecting the filter's response. While this is a simple modification that can be applied to any filter circuit (even existing boards, as we show in Section 2.4), switches aren't perfect and one must account for the effects of droop and charge injection that are a result of this addition. This chapter makes the following contributions:

- We introduce the idea that the restart delay of analog filters can be reduced (or even eliminated) by retaining the state of their memory holding elements across power gating cycles. We describe how this is both necessary and sufficient for effective duty-cycling of analog front end circuits that use active and passive filters for signal conditioning before A/D conversion.

- We show how filters can be made to ‘‘remember’’ their state by creating a memory element using a ‘‘flying capacitor’’ in the filter circuit. We provide an analysis of the requirements of the switches and of duty-cycle parameters. We evaluate the performance of our solution first on a single pole active filter and later show that it scales to a commercial off-the-shelf ECG platform that has multiple stages in the analog front end. With this platform, we show that the restart latency can be reduced by three orders of magnitude from 6s to 5ms.

- We utilize our modified platform for energy efficient QRS detection and extraction and illustrate how this leads to a 3× reduction in the analog front end's power consumption. Together with other elements, this extends the monitor's lifetime to 2 years on AA batteries.

2.2 Filters that Remember

Active filters are commonly used in the analog front end of sensing systems. They are required for signal conditioning and for ensuring a high dynamic range for the waveform of interest. While it is possible to design front ends without filters and process the signal entirely in the digital domain, the consequential demand on the resolution, noise performance and sampling rate of the A/D converter makes this impractical for most applications. This is especially true for medical monitoring, where portability and low power are critical and where analog designs have evolved over decades to meet these needs [Web98].

2.2.1 Duty Cycling a High Pass Filter

A common task in the analog front end is that of eliminating DC offset. The DC offset in biophysical signals may be due to the placement of the sensors or due to stationary calibration errors and is typically much larger than the signal of interest. It is essential,

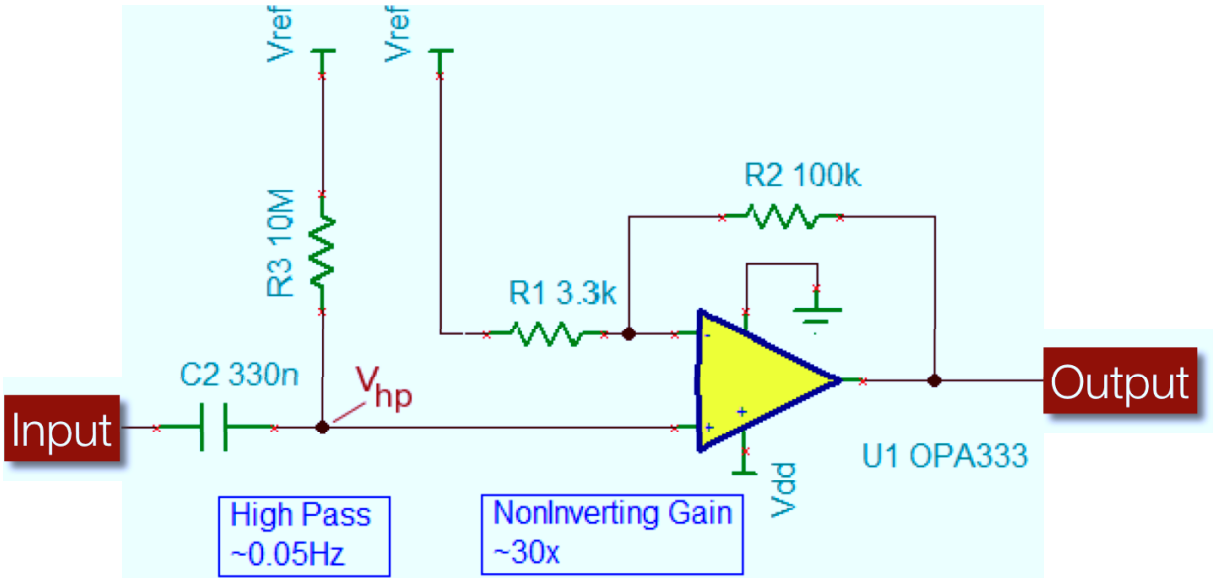

Figure 2.3: Schematic of single pole 0.05 Hz high pass filter with 30x gain using TI OPA333.

then, to filter the DC component out before any amplification to avoid saturation in downstream stages. An additional source of DC offset is bias voltage. Since monitoring devices are battery powered, there is typically no negative supply available in the circuit. This means that the inherently bipolar signal needs to be re-biased to some positive reference voltage, about which it can swing. The reference voltage is selected to maximize dynamic range and is usually mid-scale between the full voltage level and ground. Note that although a negative supply could be generated internally to avoid re-biasing, this is not done for its low efficiency and noise injection. We cover these aspects of using bipolar supplies in Section 2.5.

We introduce our solution using a simplified example. Figure 2.3 shows part of the analog front end from a diagnostic electrocardiography (ECG) monitor from Shimmer Research [Shi10b]. The schematic describes a single pole high pass filter that is used for DC offset removal and a gain stage that provides a $\sim 30\times$ boost to the weak signal picked up by the electrodes. The low frequency corner of the filter is selected depending on the application, and in this case, is set at 0.05Hz to allow accurate ST segments to be recorded [Bra00].

We begin by analyzing the behavior of the system when power is initially applied. The active elements in the circuit are the gain stage and an op-amp buffer (not shown) that generates V_{ref} and we will assume that when power is turned on, these active elements turn on almost instantaneously. While this is not strictly true, the time required for the op-amps to stabilize upon power up ($<100\mu$s) is much smaller than the time constants we will describe next. For our analysis, we also need the state of the input/output terminals of the op-amps when they are turned off. Unfortunately, device manufacturers do not specify this and from observations from the devices we use -- TI OPA333 and TI TLV2454 -- we infer that in power down mode, the op-amps have their inputs at high impedance and their output grounded.

Capacitor C_2 is connected to the input ECG signal V_i, to V_{ref} through R_3 and to the non-inverting input of the op-amp U_1. Let the right side of C_2 be labeled as V_{hp}, the high pass node. It is this node that is most significant for our study. Since the input to op-amp U_1 is in high impedance (assuming infinite impedance for simplicity), V_{hp} is affected by V_{ref} and V_i only. When power is turned on, C_2 charges through R_3 to reach potential V_{ref} and to V_i through the output impedance R_i of the input terminal. Using KVL and Laplace transforms, the voltage at C_2 can be found by:

$$\begin{aligned} V_{hp}(t) &= \frac{R_3}{R_3+R_i}\left(V_{hp}(0)-V_{ref}+V_i\right)e^{-\frac{t}{(R_3+R_i)C_2}} \\ &\quad +V_{ref} \end{aligned} \tag{2.1}$$

where, $V_{hp}(0)$ is the initial voltage on the capacitor. Now, V_i will appear on the output of the op-amp only after V_{hp} reaches a specific point. This is because the output stage provides a gain of 30$\times$ and with a voltage swing of 3V, the non-inverting input and inverting input must be within 100mV of each other to avoid saturation. Since the inverting input of the op-amp is already at V_{ref} when the circuit is powered, the output signal will only appear once:

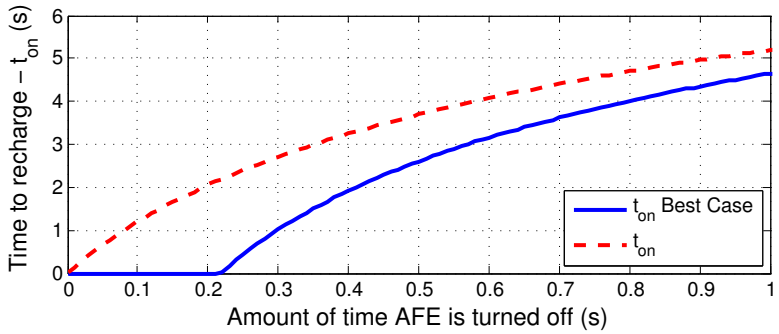

Figure 2.4: Non-linearity in re-charging time of V_{hp} until output comes out of saturation after temporary power shutdown. Best case: V_{hp} charges fully to V_{ref}.

$$V_{hp}(t) > V_{ref} - \frac{V_{dd}}{\frac{R_2}{R_1} + 1} \tag{2.2}$$

From Equations 2.1 and 2.2, one can derive the time it takes to reach this point to obtain:

$$\begin{aligned} \tau_{hp} &= (R_3 + R_i)C_2 \log\left(\frac{V_{ref} - V_i - V_{hp}(0)}{F}\right) \\ &\quad -(R_3 + R_i)C_2 \log\left(\frac{V_{dd}}{G}\right) \end{aligned} \tag{2.3}$$

where $F = \frac{R_i}{R_3} + 1$ and $G = \frac{R_2}{R_1} + 1$. This relationship reiterates the logarithmic dependence of the charging time on the initial voltage of the capacitor $V_{hp}(0)$ and highlights the need for V_{hp} to be close to V_{ref} when the circuit starts up. Based on Equation 2.3, we can analyze circuit behavior when it is duty cycled. When power is removed, V_{ref} will fall to zero since it is the output of the V_{ref} op-amp. Therefore, C2 will immediately start discharging from its value toward ground. The voltage V_{hp}, t_{off} seconds later is:

$$V_{hp}(t_{off}) = \frac{V_{ref} + V_i}{F} e^{-\frac{t_{off}}{(R_3 + R_i)C_2}} \tag{2.4}$$

assuming that the capacitor was allowed to charge to V_{ref} before duty cycling. When power is re-applied, the amount of time required for V_{hp} to recharge to the threshold in

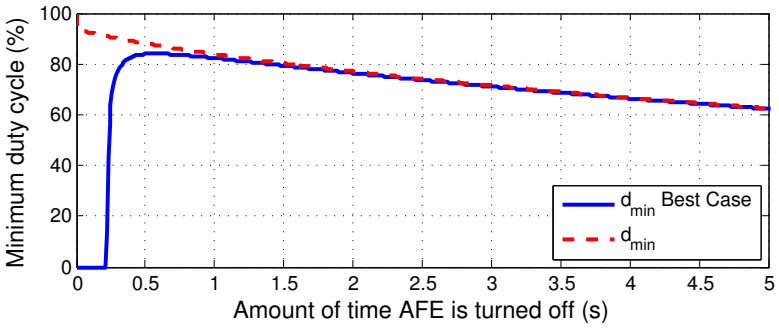

Figure 2.5: Minimum allowable duty cycle required for output to follow V_i.

Equation 2.2 can be computed by inserting Equation 2.4 as $V_{hp}(0)$ in Equation 2.3. The relationship between minimum re-charging time and turn off time, after simplifying for a zero mean V_i and $Ri \ll R_3$ is given by:

$$t_{on} > R_3 C_2 \left[log\left(1 - e^{-\frac{t_{off}}{R_3 C_2}}\right) + log\left(V_{ref}\right) - log\left(\frac{V_{dd}}{G}\right)\right] \tag{2.5}$$

and is valid for $t_{off} > -R_3 C_2 log\left(1 - \frac{V_{dd}}{G V_{ref}}\right)$ since this is the time it takes for the output signal to saturate to ground. A pictorial view of Equation 2.5 is shown in Figure 2.4, which shows the non-linearity in the recharge time (note the asymmetric axes). The minimum allowable duty cycle, d_{min}, can be computed from Equation 2.5 using:

$$d_{min} > \frac{t_{on}}{t_{on} + t_{off}} \tag{2.6}$$

and is shown in Figure 2.5 for various values of power down time. It should be emphasized that even this $> 60\%$ duty cycle is a best case situation for two reasons. First, when V_{hp} does not charge fully to V_{ref}, the dynamic range of the signal that can be amplified is reduced. And second, we assumed that V_{hp} did charge to V_{ref} in Equation 2.4 and if this were not true, the minimum on time and hence allowable duty cycle would be nominally higher.

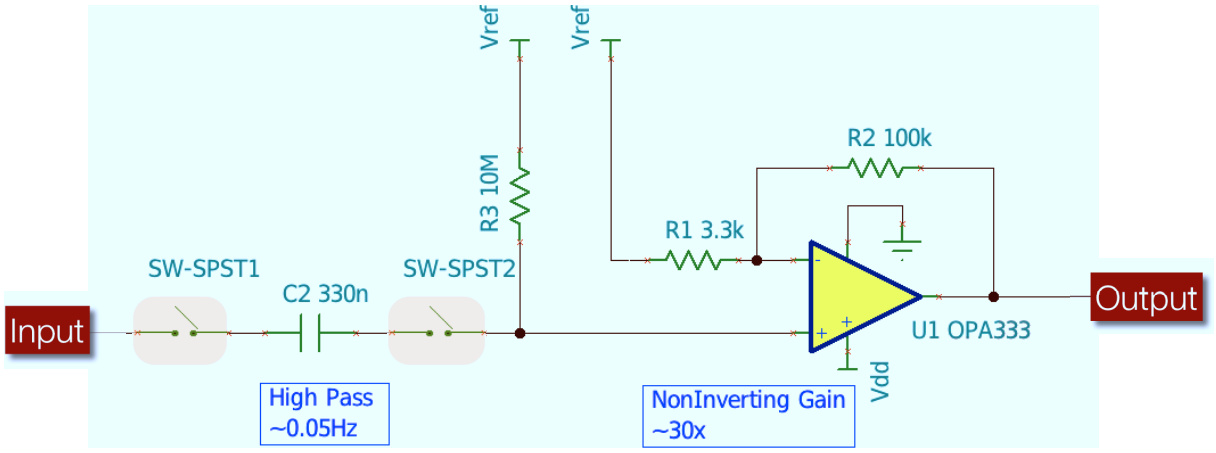

Figure 2.6: High pass filter with "flying capacitor" switches to save state.

2.2.2 Saving Filter State

From the above analysis, it is seen that in order to effectively duty cycle the analog front end, the voltage at node V_{hp} should remain close to $V_{ref} - V_i$ when the circuit is turned off. Intuitively, one may imagine that the state of the filter is held in the voltage at node V_{hp}, that is, within the charge on the capacitor. This would imply that if we were to disallow the capacitor to discharge when power is turned off, we could relax the wake-up latency requirement in Equation 2.5. This is fortunately easy to do.

Our solution to saving the filter's state is to add a pair of analog switches in series with the capacitor. The switches, as shown in Figure 2.6, are turned on and off simultaneously by the same power management controller that handles power gating for duty cycling. A strict timing condition needs to be imposed, however. The switches need to be opened a short time, Δ_{off}, *before* the power is gated away and should be closed some time, Δ_{on}, *after* power is restored. Δ_{off} is selected based on the switching time of the switch and Δ_{on} is configured based on the settling time of the rest of the analog circuit.

Through this "flying capacitor" technique, the filter capacitor is isolated from the active elements that are powered down and is prevented from discharging. When power is re-applied, the filter can continue operating almost instantaneously, oblivious to the power interruption. Note that the switches stay powered while the rest of the circuit is

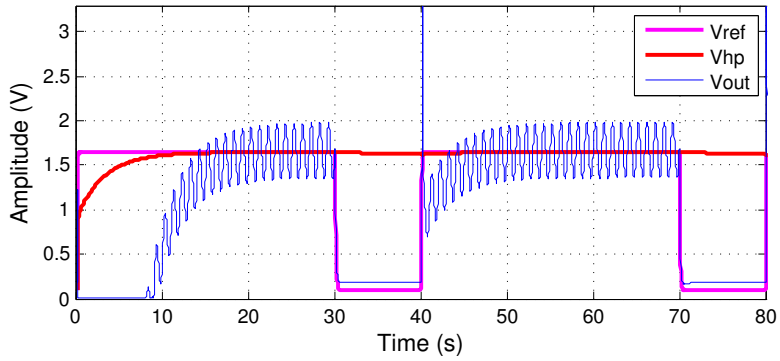

Figure 2.7: Spice simulation of high pass switcher filter showing slow initial wake-up and quick wake-up in the second iteration.

switched off and the switches must be selected such that they have a flat response for the frequency band of interest.

Figure 2.7 shows results from a Spice simulation of the high pass filter circuit with a 1 Hz sinusoidal test input and a 30s-10s duty cycle. The results clearly indicate the sluggishness of the circuit the first time power is turned on and the rapidity with which the output follows the input even after the circuit is turned off for 10s. The plot also shows how the voltage at node V_{hp} takes almost 10s to reach V_{ref} initially and how it maintains its voltage even when the circuit is turned off. The slight delay in the second power cycle is due to leakage through the switch. Since the gain of the system is very high, even a small amount of leakage from the capacitor (33mV over 10s in this case) is visible. This and a number of other issues arise in this switcher based "filters that remember" (FTR) that each cause the wake-up latency to increase. The following sections discuss the effect of each aspect.

2.2.3 Effect of Switch Leakage

Since switches are not perfect isolators when they are turned off, the charge on the capacitor will leak during power down. We can compute the effect of this leakage on the wake-up latency of the filter if we know the effective resistance, R_{off}, of the switch during power gating. The capacitor will discharge like the non-switched scenario in Equation

2.4, except that it will do so through the switch resistance:

$$V_{hp}(t_{off}) = V_{ref} e^{-\frac{t_{off}}{R_{off} C_2}} \tag{2.7}$$

The recharge time can be found in the same way as Equation 2.5 to get:

$$t_{lkg} = R_3 C_2 \left[log\left(1 - e^{-\frac{t_{off}}{R_{off} C_2}}\right) + log\left(V_{ref}\right) - log\left(\frac{V_{dd}}{G}\right) \right] \tag{2.8}$$

It should be noted that the effect of leakage is minuscule compared to the value in Equation 2.5 since discharge occurs through R_{off} which may be as high as $10^{10}\Omega$ for pass transistors. For example, at t_{off}=1s, worst case t_{lkg} amounts to 15ms for this R_{off}.

2.2.4 Effect of Charge Injection

Low power analog switches can be realized conveniently using MOS transistors. A significant issue with using MOS based switches, however, is charge injection when the switch is turned off [WVR87]. Charge injection alters the voltage across the capacitor and is especially troublesome for sample-and-hold-circuits and switched capacitor filters. Charge injection occurs in MOSFETs because of carriers that are trapped in the channel under the gate when the switch is suddenly turned off. The carriers couple through gate-to-diffusion overlap capacitances and affect the charge on the filter capacitor that is connected to its source or drain.

The effect of charge injection in our FTR architecture is slightly diminished compared to sample-and-hold circuits and switched capacitor filters. This is because we have symmetric switches on either terminal of our filter capacitor and charge injection will affect both sides simultaneously. If the switches are well matched, the amount of charge delivered to each terminal will be equal and cancel out. Nevertheless, we can derive a relationship for the worst case bound for increase in wake-up latency caused by charge injection.

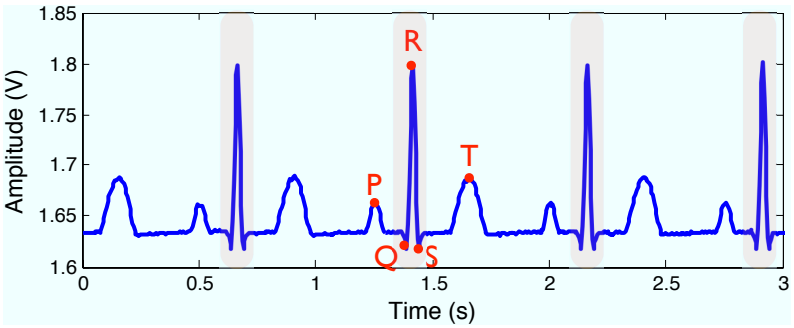

Figure 2.8: A segment of an ECG waveform showing the P-QRS-T sections with short QRS complexes marked.

Assume that based on the charge injected by each switch, ΔQ_{inj} Coulombs of charge enter the filter capacitor C_2. The change in voltage due to this extra charge is $\Delta V_{inj} = \Delta Q_{inj}/C_2$. When the switch turns on, therefore, the voltage on the capacitor will be $V_{hp} = V_{ref} \pm \Delta V_{inj}$ assuming the capacitor was at V_{ref} when the switch was turned off. Similar to the analysis above for leakage from Equation 2.5, the amount of time needed to recharge the capacitor to its nominal voltage can be found using:

$$t_{inj} = R_3 C_2 \left[log\left(\frac{\Delta Q_{inj}}{C_2 F}\right) - log\left(\frac{V_{dd}}{G}\right)\right] \tag{2.9}$$

Typical switches have a charge injection of about 5-50pC [Tex], which translates to $150\mu V$ change in the capacitor voltage for a 330nF capacitor and 5ms of additional latency.

2.2.5 Effect of Switch Series Resistance

In their on-state, switches may have a series resistance between 1-10Ω and this affects the frequency response of a filter. Low series resistance is desirable, but there is one caveat. Low series resistance is achieved by sizing the MOS transistors to be relatively large, but large transistors have particularly pronounced charge injection due to the increased number of carriers in the wider channel. Switch selection must therefore be made carefully to balance the effect of series resistance on the frequency response and the effect of charge injection on wake-up latency. When using the FTR architecture, one could include the

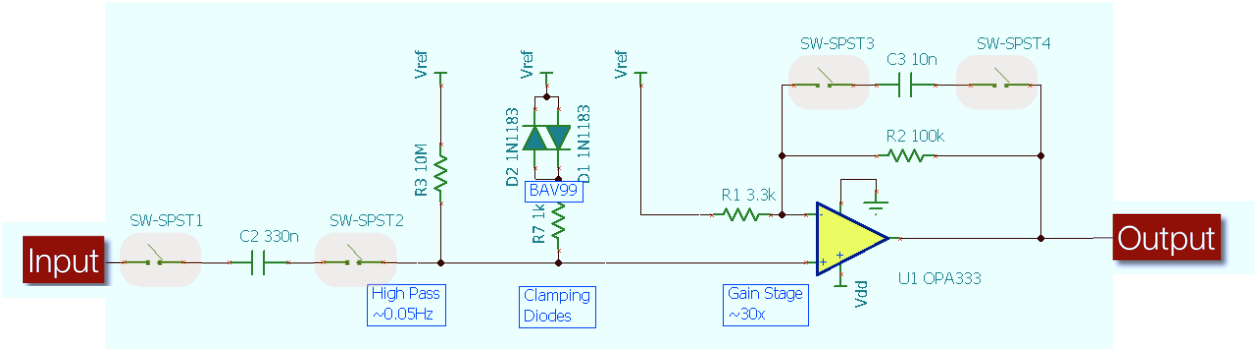

Figure 2.9: Schematic of final stage in Shimmer ECG front end. A pair of analog switches disconnects each capacitor to save its state before powering off.

switches' resistances within the effective series resistance parameter of the filter capacitor to model this effect on frequency response.

2.3 Using a Duty Cycled AFE for ECG

A duty cycled AFE can be employed whenever circuit characteristics permit. For example, in Section 2.1, we suggested that one could use a duty cycle-able AFE by turning it off between sample acquisitions. This strategy works well when Δ_{on}, the settling time of the analog circuit is much lower than the sampling period, t_{per}, and when $P_{off \rightarrow on}$, the power consumption of the circuit during wake up meets:

$$P_{off \rightarrow on} < \frac{(P_{on} - P_{off})(t_{per} - t_{acq} - \Delta_{off})}{\Delta_{on}} + P_{off} \tag{2.10}$$

where, P_{on} and P_{off} are the power consumption in the on and off states respectively and t_{acq} is the sample acquisition time. This relationship essentially compares the average power consumption with and without duty cycling accounting for the wake up latency and power. Unfortunately, this relationship does not hold for all circuits due to various reasons -- see Section 2.4.4 for an example.

Some applications, however, only require capturing short segments of the entire waveform to achieve diagnostically continuous monitoring. This means that the wake-up latency and power consumed during that event could be amortized over multiple sample

acquisitions at the cost of keeping the AFE on longer. One example for electrocardiography recording was developed by Hua, et. al [HLR], where they observed that accurate heart rate information is sufficient for many purposes. The beat rate is computed by detecting the R-peak of the QRS complex as shown in Figure 2.8 and inverting the R-to-R interval between consecutive peaks. Their power reduction scheme consisted of turning off the AFE between peaks and predicting a future time to re-power the AFE (to acquire a burst of samples) just in time to sense and detect another R-peak. In addition, they use a sophisticated modeling technique to estimate R-times retrospectively in case of a misprediction.

Inspired by their approach, we evaluate a QRS extraction algorithm that provides the signal waveform window corresponding to each QRS complex (in addition to the heart rate). The QRS complex is significant for diagnosing many conditions, such as multiple of forms of arrhythmia [KSS98, SHE03]. The QRS complex represents the depolarization of the ventricles and from Figure 2.8, we observe that it is a fairly conspicuous but quick event -- lasting 80-120ms when the heart is functioning properly. Note that this application is particularly well suited to AFE duty cycling due to the quasi-periodic nature of heart beats. Applications that involve tracking asynchronous, rare and potentially life-threatening phenomena such as the onset of ventricular fibrillation (cardiac arrest) should avoid burst sampling to ensure no critical data is missed.

Analyzing 48 recordings from PhysioNet's MIT-BIH Arrhythmia [GAG00] database, we estimated that an oracle scheme that provided an accurate prediction of the occurrence of each QRS complex would lead to an ideal sensor duty cycle of 11.5±3.5% (variation across the traces). That is, by turning on the AFE at the right moment and keeping it powered 11% of the time, each QRS complex could be correctly detected, extracted and communicated. The R-peak can be detected from the QRS complex and the heart rate can also be computed accordingly. The 11% estimate does not yet include the effect of wake-up latency and is meant to set a lower bound for the achievable duty cycle with ideal hardware and perfect knowledge of QRS instants.

Our duty cycling strategy operates by estimating a model of heart rate from previous beats and predicting the next R-time to within a factor of the current R-R interval. QRS instants follow the heart beat rate of an individual and it is well known that human heart rate is limited to within 30-220bpm. The upper bound is especially important because it implies that once a QRS complex is detected, another one will not occur for another 270ms, and thus the entire system, including the AFE can be turned off for that duration. Second, while heart rate is known to fluctuate in both a long term and short term fashion, large short term variations are quite limited. This implies that a conservative measure of tolerance with respect to the QRS prediction may be sufficient for capturing the signals of interest most of the time.

Let the estimated R-R interval in iteration k be termed $\hat{r}_k$. If the k-th R-peak is detected at time p_k, a first order approximation of the R-R interval can be found using $\hat{r}_k = p_k - p_{k-1}$. Although a more sophisticated estimation algorithm, such as [HLR], may result in a more accurate estimate of the heart rate, we use first order approximation to curtail processing burden and hence power.

Based on the estimate $\hat{r}_k$, we predict that the next R-peak will occur, at most, $\omega\hat{r}_k$ seconds later. That is, $\tilde{p}_{k+1} = p_k + \omega\hat{r}_k$. Where, ω ($\omega > 0$) is the tolerance parameter that balances the trade-off between duty cycle and missed beats. A very low ω would wake the sensor up too early while a very high ω would wake it up too late, potentially missing beats. The instant that the sensor is actually powered up needs to account for the length of the QRS complex and the non-zero wake-up latency, Δ_{on}, of the AFE:

$$t_{k+1} = p_k + \omega\hat{r}_k - \frac{\tau_{QRS}}{2} - \Delta_{on} \tag{2.11}$$

This relationship assumes that the QRS complex is symmetric about the R-peak, which is reasonable in most cases. If the QRS complex is determined to be asymmetric due to an abnormal condition, a lower ω could be used. Furthermore, our algorithm implements a feedback mechanism to set ω. The sensor stays powered on until an R-peak is found by

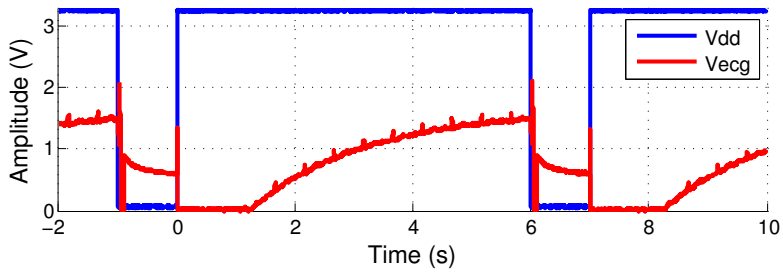

Figure 2.10: Wake-up latency of Shimmer ECG before any modifications, shown with a 6s on time and 1s off time.

a QRS detection algorithm executing on the sampled data. Since it is known that heart beats must occur within a maximum time period (even in bradycardia), if a QRS complex does not occur within a reasonable factor of its predicted instant, the sensor may have missed the beat. When the sensor detects this condition, it reverts to a conservative ω to ensure that future beats are not missed and guesses the time of the missed beat as being just before the sensor was turned on. An evaluation of this algorithm is presented in the next section.

2.4 Evaluation Results

We evaluate our solution on a commercially available ambulatory ECG monitoring analog front end from Shimmer Research [Shi10b]. Figure 2.9 shows the schematic of the final stage of the front end, before the signal is fed to the A/D converter. There are three sections in the design: high pass filter, signal clamping diodes and low pass filter with gain. The high pass filter section was discussed in Section 2.2.

The signal clamper limits the signal to within one diode drop of the bias voltage to prevent damage to the final op-amp stage in cases of defibrillation and lead switching [Web98]. It also provides a preferred path for capacitor C2 to charge. This reduces the wake-up delay considerably since half the charging voltage is reached before the diode cuts off.

Figure 2.10 is a signal snapshot measured from the AFE with an on-off duty cycle

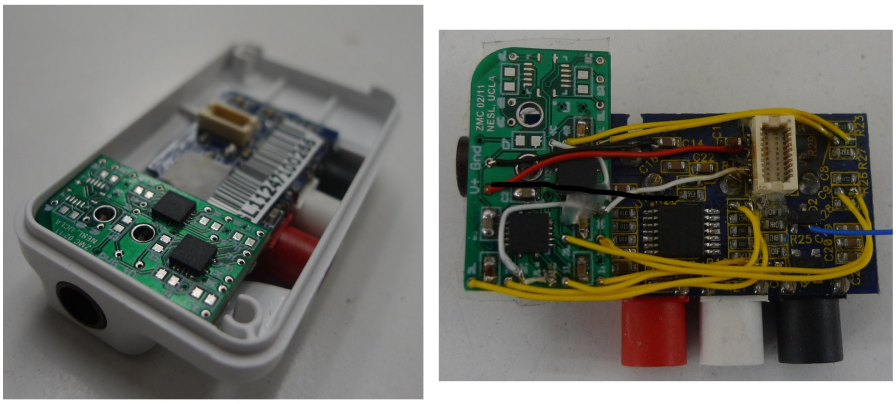

Figure 2.11: The ShimmerFTR circuit is designed to fit inside the Shimmer ECG enclosure for seamless integration. Wiring to filter capacitors (yellow) to vacant pads on board. Wiring for power (red/black) and switch control (white) made to unused pins on Shimmer internal expansion connector.

of 6s-1s respectively. Compared to the high pass filter in Section 2.2, this circuit has a reduced wake-up delay of about 6s before it reaches the bias voltage of 1.65V, while the circuit in Figure 2.3 starts in 18s. This difference is attributed primarily to the accelerated wake-up due to the signal clamping diodes.

2.4.1 ShimmerFTR: Modifying the Shimmer AFE

Figure 2.9 also marks the analog switches that were added to the schematic to enable it to be duty cycled. We used the TI TS3A4751 analog switches for their low 0.9Ω series resistance and relatively low charge injection of 3.2pC for a 1nF load. With our load, we measured the charge injection mismatch to be closer to 30pC. Since the circuit has two capacitors, C2 for high pass and C3 for low pass, and the front end provides 2 ECG channels (II and III), we used eight switches in all.

Figure 2.11 (left) shows the (partially populated) board we designed to fit inside the Shimmer ECG enclosure. While we used a larger 16mm^2 switch, designers could opt for a 4mm^2 package to reduce area overhead. We de-populated the filter capacitors from the Shimmer board and placed them on our board instead. We then wired the Shimmer board

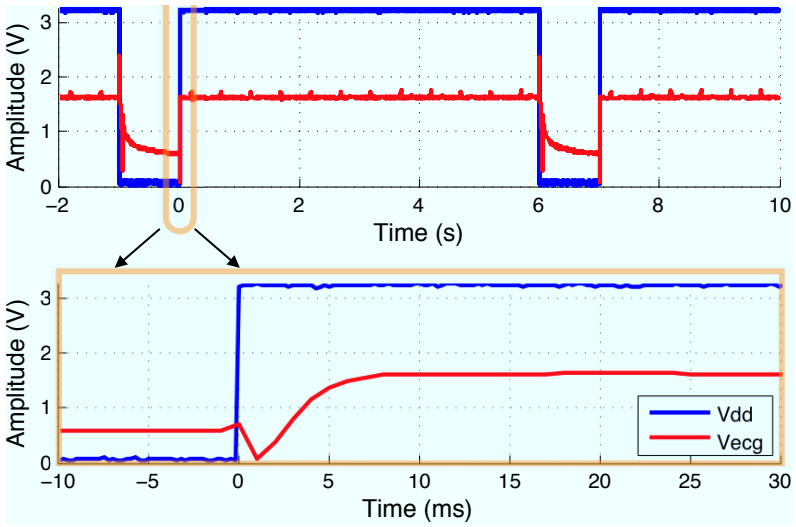

Figure 2.12: Wake-up latency of Shimmer ECG using switches on either side of high pass filter capacitor C2 in Figure 2.9.

to line the switches in series (yellow cables) with the rest of the filter circuit (Figure 2.11 right). The switches are powered and controlled independently from the rest of the AFE through two unused pins on the Shimmer Internal Expansion Connector that interfaces to the MSP430 micro-controller that the AFE is connected to (red/white cables).

2.4.2 ShimmerFTR Wake-up Delay

Figure 2.12 shows the measured result of using just the capacitor C2 switches (i.e. SPST1 and SPST2). In this way, only the high pass filter (time constant=3.18s) was affected. The low pass filter is kept un-switched. The top part of the figure shows the result with the same duty cycle of 6s-1s as above, clearly indicating the almost instantaneous turn on time of the circuit. The tiny spikes are the R-peaks to be detected. Incidentally, the waveform shown in Figure 2.8 was the ECG signal (amplitude adjusted) recorded in this case. The bottom plot is a temporally zoomed version spanning 40ms near the start of the switching transient (note the change in axis). The bottom plot illustrates that much of the reduction in wake-up delay is already achieved by switching just the high pass filter capacitor and that the signal is stable at V_{ref} in about 10ms.

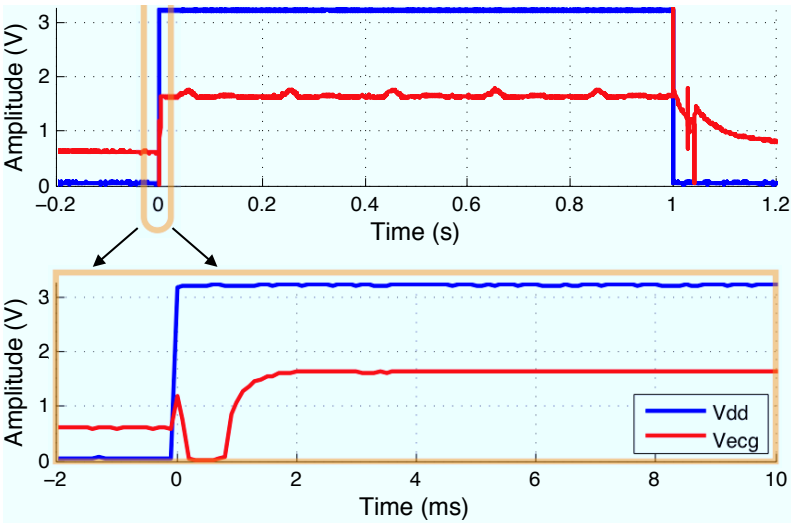

Figure 2.13: Wake-up latency of Shimmer ECG using switches on either side of C2 as well as low pass filter capacitor C3 in Figure 2.9.

When the low pass filter capacitor is also switched (time constant=1ms), the wake-up delay is reduced even further to about 2ms as shown in bottom half of Figure 2.13. From additional experimentation, we found 2ms to be a practical lower bound for this circuit, which results from the start-up delay of the op-amps, the charging time of the reference buffer capacitance and the effect of charge injection.

A 2ms wake up latency implies that the system can sustain switching rates up to 500Hz at a 100% duty cycle. Although 500Hz is deemed sufficient for diagnostic quality ECG, the real benefit of a switched AFE architecture is obtained at low duty cycles. We next evaluate our burst sampling strategy for QRS extraction to achieve this goal.

2.4.3 QRS Detection Evaluation

To evaluate our QRS detection scheme, we use recordings from PhysioNet's MIT-BIH [GAG00] database. Arguably, this represents a worst case condition since the dataset contains a high incidence of arrhythmia (mean variation of -14bpm to +19bpm from median heart rate), which challenges the predictive power of our first order beat rate model (Equation 2.11). Also, for our evaluation, we use a conservative estimate of 5ms

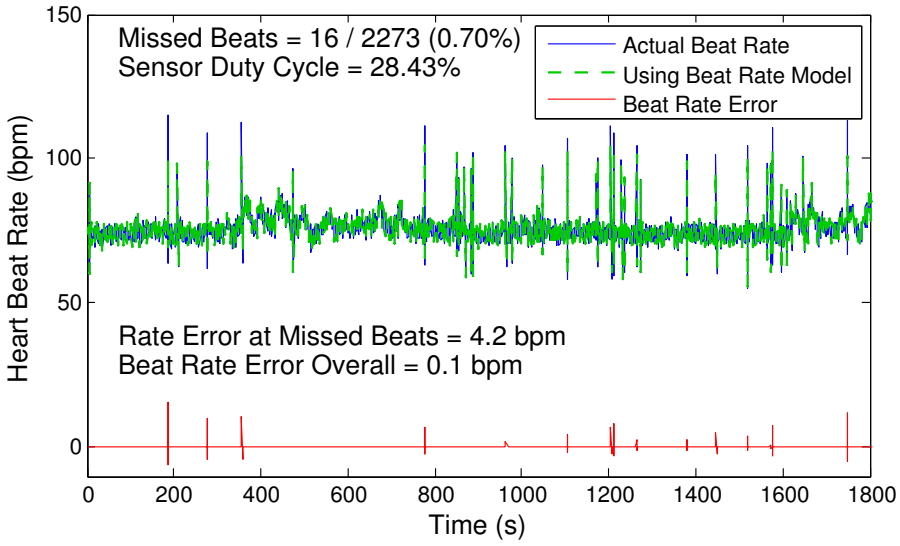

Figure 2.14: Estimating the beat rate using a duty cycled analog front end.

for our wake up latency to account for variability in recharge overhead from charge injection and leakage.

Figure 2.14 shows the actual beat rate of a 30min segment of ECG. Overlaid is the result from the QRS prediction algorithm we described in Section 2.3 with a fixed $\omega = 0.8$. For this dataset, with the sensor wake-up delay set at 5ms and the QRS window size set at 100ms, we see that less than 1% of the beats were missed. With a 28% sensor duty cycle, this meant that the QRS complexes for these 16 beats were lost, but the heart rate estimate at these beats was only 4.2bpm off. Overall, the heart rate estimate for the entire 30min segment differed by less than 1 bpm. Note how even though the heart rate signal has multiple spurious beats (many more than 16), much of the arrhythmia was captured correctly.

Figure 2.15 depicts the trade-off between the number of missed beats and the achievable sensor duty cycle based on varying the parameter ω. Error bars indicate the first and third quartiles across the 48 datasets used. It is interesting that there is an optimal value of ω for which the sensor duty cycle is minimum (close to the 20%). The reason that the sensor duty cycle increases beyond that point is because for $\omega > 0.98$, the sensor starts missing beats very often and needs to stay powered on until it detects the next QRS

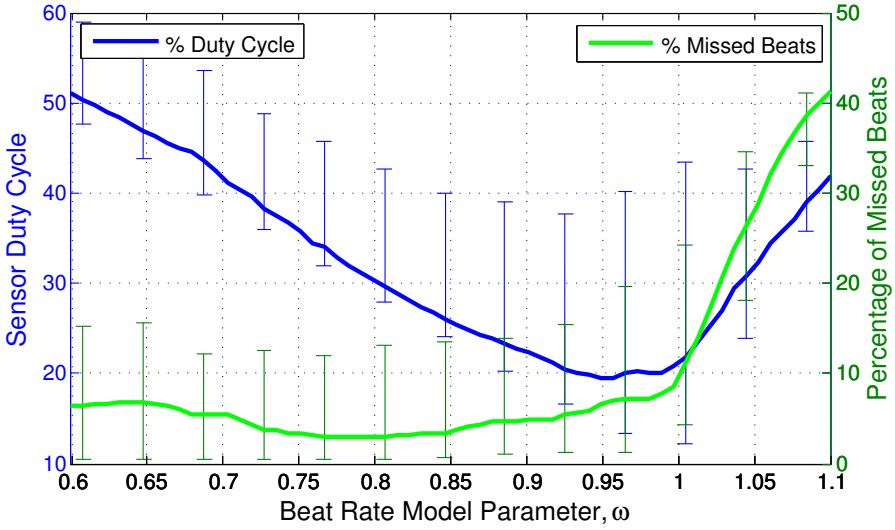

Figure 2.15: Trade offs between duty cycle and number of missed beats for varying model parameter.

Table 2.1: Power consumption of ECG subsystems

Subsystem Power (mW)	Always On	Duty Cycled	
		μC + Radio	+ AFE
Shimmer AFE	0.535	0.535	**0.174**
μC - Quiescent	0	0.003	0.003
μC - Active	12	0.0353	0.209
Radio - Quiescent	0	0.025	0.0231
Radio - Active	18	0.081	0.0304
Total	30.535	0.6793	0.4395
Lifetime (days)	12	552	853

complex. This is evidenced by the fact that there is a sharp increase in the percentage of missed beats at that point. It must be added that for a dataset from the Long Term ST database [GAG00], akin to a more realistic setting, the algorithm achieved an 8% duty cycle with less than 0.2% missed beats.

2.4.4 ShimmerFTR Energy

Finally, we can evaluate the overall system power when using ShimmerFTR. Since the circuit is turned off completely in sleep mode, the active elements require settling time when the circuit is turned back on. This usually involves setting up appropriate internal bias currents and this increases the current consumption during the switching transient. The power consumption is almost double that of its quiescent value of 535μW for a short period of 5ms each time the board is powered on. Along with the 2μA consumed by the switches, this results in approximately 8μA of overhead current assuming an 80bpm average heart rate.

Table 2.1 lists the average power consumption of the subsystems for a duty cycled ECG monitor. The first column is the base power of the systems assuming they are on all the time. The second column results when aggressive duty cycling is applied to the processor and radio, showing how the AFE, which had a minuscule contribution earlier suddenly becomes influential. Values for the radio (nRF24L01+) using its power-down state are measured for a 256Hz sampled ECG signal at 10-bits per sample buffered for 96 bytes

before a burst is transmitted. A more extensive study on the power characterization of this radio for ECG sensing can be found in [Cha11]. Microprocessor values were measured and verified with datasheets and [Roo10], which reports ADC power at 17.1nW/Hz inclusive of voltage reference, combined with instruction level simulation for the MSP430F2012. The values labeled as the quiescent also include contribution of wake up power.

The third column adds our duty cycling methodology for QRS extraction to the AFE. Accounting for the switching overhead and a 28% AFE duty cycle, the measured power consumption for the AFE was reduced by 3×. This resulted in a median of 3% missed beats, even with extreme arrhythmia. The overall system power decreased by 35% and the lifetime of the monitor is about 2 yrs on 3000mAh L91 AA batteries.

Notice that the reduction in AFE power is offset by a substantial increase in processor power due to additional QRS extraction and beat rate modeling (52K instr/sec extra -- unoptimized code). The radio power consumption has reduced because only short segments corresponding to each QRS complex are now transmitted (64 samples/beat at 80bpm). The slight reduction in radio quiescent power stems from the lower data rate, which allows longer buffering (800ms versus 300ms without FTR) and fewer expensive wake ups.

2.5 Related Work and Discussion

In the context of ambulatory monitoring, there is no prior work with a specific focus on mitigating AFE wake-up delay. Priors have either ignored the importance of wake-up delay in analog filters or incorrectly determined it. A broad category of work has recognized that the AFE imposes start-up constraints and duty-cycling restrictions. The works have sought out algorithmic solutions using caching, sample bursting, blind periods, or other techniques to justify longer active times.

Representative of these is Hua, et. al. [HLR], which was mentioned in Section 2.3. Their scheme consists of predicting when the next QRS complex would occur, turning

on the analog front end and ADC just-in-time to capture it and turning them both off after the QRS complex is detected. In their simulation environment, it was assumed that the duty cycle they use, 10-25% of the inter-beat period, is adequate for initializing the circuit after each power gating cycle. This corresponds to only 500ms in the best case (30bpm heart rate) and as little as 15ms when the heart is in distress (severe tachycardia) -- inadequate to stabilize an AFE with sub-Hz filters.

Wake-up latency is a significant issue in duty cycling digital logic and radio transceivers as well. Some researchers reduce wake-up latency by leaving the sleep transistors between the supply rails and the digital logic in weak inversion rather that in the off mode [AND06]. By doing this, the transistors do leak some current even when they are off, but are able to transition to active mode fairly rapidly. This technique could be applied to our switcher architecture too, not to save state but to solve a separate issue -- peak power during wake-up. As noted in Section 2.4.4, turning off the op-amps requires them to re-establish internal bias voltages causing transient current spikes. A solution would be to use op-amps with a shutdown feature. These op-amps maintain their internal state at the expense of slight leakage ($<1\mu$A) to prevent re-biasing at every wake up. Using leakage currents to save the state of memory elements directly is not practical because it would affect the charge stored in them, result in an altered filter response and likely cause distortion at the output.

Perhaps the most striking in similarity to the FTR architecture is the switched capacitor block. Switched capacitor (SC) blocks are widely used in analog designs as replacements for resistors. They present a more versatile and more compact alternative for integrated circuits than poly-silicon resistance. Apart from the topological difference between FTR and SC blocks (Figure 2.16), there is a fundamental difference in operating principle. SC blocks emulate a resistance by controlled *charge transfer* at a specific rate, whereas FTR targets *charge retention* to save filter state while duty cycling. For SC blocks, the switches are alternately turned on and off (opposite polarities) and the rate coupled with the capacitance determines the average resistance. In FTR, the capacitance

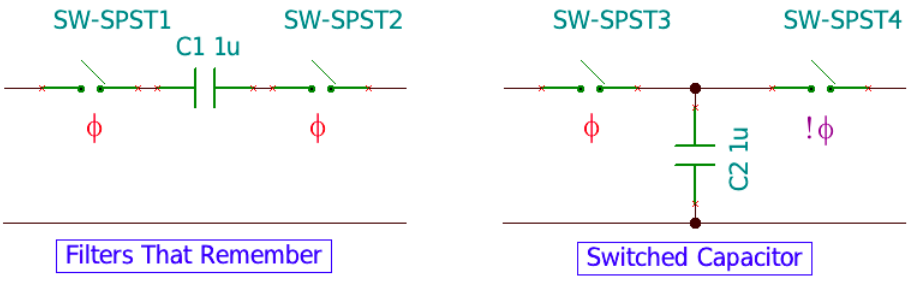

Figure 2.16: Filters that remember are different topologically and in working principle from switched capacitor blocks.

is designed for the required filter performance and the switching rate depends on the duty cycling period.

Alternative Approaches Since wake up latency primarily stems from charging the capacitor to the bias voltage to handle bipolar signals, it might be tempting to reduce latency by simply equipping the circuit with bipolar supplies. While promising, this is not the most desirable route to take for a few reasons. First, to create a mid-potential source losses must ensue. In the case of series sources with center tapping the cost and size of the energy source is doubled (ceteris paribus). If an inverting switch-mode power supply is used, the regulator must maintain operation continuously resulting in substantial quiescent losses and potential noise injection into the signals we are trying to capture. Second, interfacing with microprocessors (all of which are unipolar) presents additional challenges requiring additional hardware to perform level translation to ensure voltage compatibility. Third, using a bipolar supply does not solve the critical issue of filter state loss. When the circuit is powered up, the filter capacitor will start at ground regardless of where it was when power was removed.

In a similar vein, the use of a switch to short out the large current-limiting resistor of the filter element is also ill-advised. In so doing, the actual state of the filter is lost. The level on the capacitor when the circuit is turned off is critical and may deviate substantially from the reference voltage. This approach was evaluated experimentally

and rejected during the course of this investigation.

A better approach to reducing power is tightly matching the requirements of the AFE with those of the application. For example, commercial fitness heart rate monitors (e.g. Polar [LV01]) can achieve years of lifetime without duty cycling. Since this class of systems only seeks to determine heart rate, their approach is based on a filtered signal crossing a threshold (a 1-bit quantization). This results in a very limited requirement for linearity due to a lack of harmonic complexity. The analog front end can be designed with extremely low power op-amps, where the large voltage offsets (as big as the ECG waveform itself), low gain bandwidth product and additional noise of the amplifiers are inconsequential. Laukkanen, et. al. evaluate the accuracy of their devices to be within ± 6 beats/min, $\sim$95% of the time, with respect to a Holter monitor.

Even if better digitization was applied, the textile electrode-body interface in these devices give a ''lump'' (vs. a true) QRS complex preventing the recovery of a detailed waveform. While average heart rate measurement benefits from this biophysical low-pass filter, it is unacceptable for long term medical monitoring since important events such as arrhythmia, rapid change in heart rate (as from fear or other anticipation of exercise), and indicators for other acquired pathology are being missed [LV01].

Limitations A key limitation of the FTR approach, like other burst sampling techniques, is that the system would miss asynchronous events and anomalies. FTR is better suited at acquiring (quasi-)periodic signals and if Inequality 2.10 is satisfied, turning off the AFE between sample acquisitions. Unfortunately, Inequality 2.10 is not satisfied for ShimmerFTR with a 256Hz sampling rate due to charge injection effects. That said, a 28% duty cycle (active for 210ms at 80bpm) is sufficient to capture most (but not all) anomalies. FTR is also unsuitable when the switch series resistance would affect the filter response unacceptably or when the required duty cycle is too high -- when acquiring long S-T segments or P-wave segments, for example.

2.6 Conclusion

This chapter described how the addition of two SPST switches per filter (active or passive) can eliminate most, if not all, of the *restart* latency mandated by filter time constant, without significantly affecting its frequency response. The technique is not exclusively relevant to the ambulatory ECG monitoring circuits that we analyzed. The analog front end is beginning to influence the power budget for other biophysical sensors, like EEG [HKC09] and accelerometers as well. We argue that whenever an application requires only a part of the signal waveform -- to compute strides with accelerometers or spikes in EEG -- duty cycling the analog front end will lead to substantial benefit. As an example, consider the ADXL335, a popular low-cost low-power 3-axis accelerometer. The device is rated at 350μA in active mode and provides analog outputs that must be filtered for noise reduction and anti-aliasing. An application such as stride analysis, which requires information in the 2Hz - 5Hz band [BBS08], may opt to sample the signal at 10Hz, a sample every 100ms. However, this requires a filter capacitance of 1μF resulting in a practical wake up delay of 160ms (see note 7 in Table 1 in [Ana10]) precluding the use of duty cycling. With filters that remember, even with a conservative estimate of 10ms for start up, the accelerometer could be running at a tenth of its power.

As others have noted, there is an additional processing burden associated with this form of duty cycling to predict the right instant to wake up, but we believe that our switch based filter architecture has shifted the scales in favor of AFE duty cycling. It should be emphasized that duty cycling the analog front end not only reduces its power but also leads to indirect benefits in downstream processing steps, since it lowers sampling, processing and communication burden proportionally.

CHAPTER 3

CapMux: A Scalable Analog Front End for Low Power Compressed Sensing

3.1 Introduction

Many problems in physiological sensing can be tackled more effectively when we have at our disposal long-run high-quality sensor data. However, constraints on batteries and form factor of wireless sensor nodes combined with the desire to operate them for extended periods necessitate that signal acquisition itself be energy efficient and that data be available in a form amenable to efficient transport and analytics.

Many physical phenomena follow specific patterns and models, which can be exploited to parametrize and compress the sensed data. This compression step, however, is traditionally performed in the digital domain only after the data has already been sampled at the Nyquist rate. Nyquist rate sampling is rather wasteful for compressible signals since their information content may be vastly lower than that assumed by the Nyquist criterion. Furthermore, it is not always practical to perform compression at low power sensing sites and the data might need to be transported uncompressed (also wasteful!) to a high powered collection center.

Compressed sensing is a recent breakthrough in signal processing that allows one to acquire a compressible signal at much below its Nyquist rate [CRT06b]. While the

Shannon-Nyquist theorem provides sufficient conditions for recovering a periodically sampled signal based on its bandwidth, it falls short when trying to include knowledge of other signal characteristics. Compressed sensing (CS) is a mathematical tool that can utilize *a priori* knowledge of the sparseness or compressibility of a signal of interest within a model framework to acquire a signal at essentially its ''information rate''. The basic tenet of CS is that if a signal is known to be sparse (contains few non-zero values) or compressible (values decay quickly to zero) in a known domain, it is wasteful to sample the signal at the Nyquist rate because most of the data will be thrown away subsequent to compression.

There are three aspects to implementing CS -- knowing the domain in which the signal is sparse, low dimensional signal acquisition and the recovery process. The sparse domain is a transform space, typically a set of linear functions, that facilitates a compressed representation of the input signal. That is, when the signal is transformed from its native domain (say, time or space) to the sparse domain (say, frequency or wavelets), a few values in the sparse domain may be sufficient to describe the signal with high fidelity. This compaction property is the cornerstone of compressed sensing. As examples, audio is known to be compressible in the frequency domain and natural images are compressible in the wavelet domain.

Signal acquisition in CS involves projecting the signal to a lower dimensional domain that is ''incoherent'' to the sparse domain *before* sampling [CR07]. This incoherent projection may be viewed as an information preserving transform that, ideally, maximizes the information collected about the signal in each sample. The fact that this incoherent sampling domain is of lower dimensionality yields compression. The projection process generates a set of compressed measurements and can be accomplished at relatively low complexity, making CS an attractive alternative for energy efficient sensing. This chapter describes in detail our design for a low power hardware implementation of this projection process.

The recovery process exploits the fact that there are few information bearing compo-

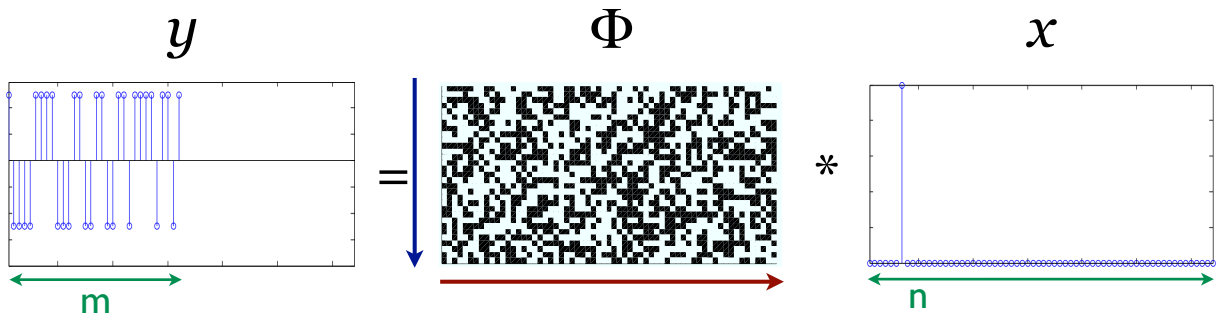

Figure 3.1: Visual representation of compressed sensing through pseudo-random projections from a $+1/-1$ (black/white) Bernoulli distribution. The measurement vector $y \in \mathbb{R}^m$ is computed by projecting the sparse input vector $x \in \mathbb{R}^n$ using a sensing matrix $\Phi \in \mathbb{R}^{m \times n}$ by $y = \Phi x$.

nents in the sparse domain in order to identify them. Generally speaking, it looks for the most compact (sparsest) solution that meets the constraints set by the compressed measurements. The quality of reconstruction depends approximately on the ratio of the number of compressed measurements acquired to the number of information bearing (non-zero) components. The catch, however, is that the recovery process is computationally intensive. A number of recovery algorithms exist that each trade off computation for accuracy differently [CWB08, TG07, CDS98].

CS could be viewed as an asymmetric compression scheme with economical encoding but expensive recovery. This makes compressed sensing particularly well suited for low power physical sensing, where sensor devices are highly constrained, both in energy and computational resources. It is expected that the compressed domain samples would be delivered to a capable base-station or backend for signal recovery or inferencing.

Sampling Compressively

Returning to signal acquisition, the key intuition behind CS's incoherent sampling strategy is that of spreading information content in the signal vector (with respect to the sparse domain) across the compressed domain samples. In some sense, one could view each compressed sample as providing an independent summary of the input signal. Researchers have identified that domains constructed from certain random distributions have high

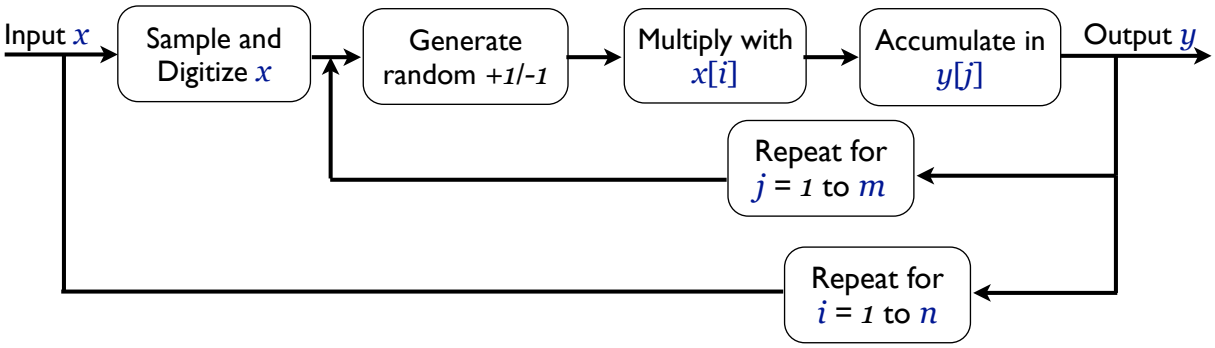

Figure 3.2: Software implementation of Compressed Sensing needs pre-digitized data and $\mathcal{O}(mn)$ operations.

incoherence with virtually any sparsity inducing basis with high probability. This is termed the universality property of compressed sensing [CT06]. Figure 3.1 shows a visual representation of a conceptual CS acquisition. In this case, we assume that the input signal, x, shown rightmost, is sparse in the time domain with an exaggerated sparsity of just one non-zero component. The sampling matrix, Φ, is generated from a random $+1/-1$ (black/white) Bernoulli distribution, i.e., we toss an unbiased coin for each element of the matrix. The measurement vector, y, smaller in dimensionality than x, is computed through $y = \Phi x$ and constitutes the compressed domain samples we seek.

The effect of using a random kernel is that regardless of where the information bearing components lie in the sparse domain, there is a statistical chance that the sampling matrix captures it. Observe, for example in Figure 3.1, how one non-zero value in x manifests itself in each of the compressed measurements in y. It must be mentioned that while randomized projections are not the most efficient[1], the flexibility afforded by the universality property is overwhelmingly attractive.

Although CS holds promise for many applications, the need for incoherent projections prior to sampling has limited its direct impact to instances where the sampling domain is inherently incoherent to the signal of interest. For example, this holds for magnetic resonance imaging (MRI), where sampling is performed in the frequency domain while

[1]Ideally, the rows of the measurement matrix should be orthogonal. Randomly generated matrices only have weak orthogonality guarantees [LKR12].

the signal being recovered is an image in the spatial domain [LDP07]. In other sensing applications such as EEG (electroencephalograph) or ECG (electrocardiograph) monitoring, where sampling is traditionally performed in the time domain, the incoherent projections have to be applied explicitly [KMK11]. Figure 3.2 depicts a software approach to CS. The inner loop performs m row-wise multiply-and-accumulate (blue downward arrow in Figure 3.1) and the outer loop executes across columns for every sample acquired (red rightward arrow in Figure 3.1). The output y is read every n samples and the accumulators are then reset.

There are three practical issues with this software based CS technique. First, it requires that the signal be explicitly sampled and digitized before projections can be computed. Since CS was developed as a solution to avoid unnecessary sampling, while viable in some instances [KMK11], this technique does not fully exploit the advantages that compressed sensing offers. Second, an order of $\mathcal{O}(mn)$ explicit mathematical operations are typically required to compute the projection. The computational cost of explicit compression for some transform domains may actually be lower ($\mathcal{O}(n\log n)$ for FFT), although software CS operations are simpler (add/subtract). Third, since samples are digitized prior to the projection, quantization error accumulates as n increases. That is, as the compression ratio (n/m) increases, so does the effect of quantization [SBB06].

In this chpter, we propose a novel hardware approach to compressive sampling. Our system, called CapMux, combines digital logic with a custom analog front end to realize randomized projections at very low power. CapMux takes as input a time domain signal of variable duration (i.e. variable n) and produces a fixed number of compressed domain analog measurements (i.e. fixed m) over that duration. These compressed measurements can be fed directly to an analog-to-digital converter to be digitized, transported and processed for signal recovery or inference. The key idea that makes our architecture low power is being able to amortize the quiescent current consumption costs of high performance analog components through time-multiplexing. CapMux is constructed from just one active analog signal processing chain that can be shared across an arbitrary

number of channels. CapMux not only leads to a lower average power per measurement channel, but also admits low complexity scaling.

The chapter (reproduced from [CMS12]) makes the following contributions:

- We introduce CapMux, a scalable architecture for low power compressed sensing. CS involves projecting or convolving a signal with a set of random basis vectors. In analog circuit terms, this can be achieved by multiplying the signal with each random vector independently and simultaneously and integrating (or low pass filtering) the result. Conventionally, this would require as many analog processing chains as measurement channels desired [KLW06]. CapMux, on the other hand, shares access to a single modified analog processing chain by time multiplexing its use. Our architecture hinges on the design of a multi-channel integrator that saves its state while switching time slots. The idea was inspired from a four decade old technique [BP72] used to construct higher order active filters from a single operational amplifier.

- We have realized a 16 channel CapMux prototype designed from commercially available components. Random vector generation and time slot synchronization for the multi-integrator is orchestrated by a micro-controller that also contains the ADC to sample the compressed measurements. We describe calibration routines for managing component variation and the handling of parasitic capacitances that become significant due to the large number of channel switches. We also outline an extension to a larger 64 channel board.

- We evaluate our prototype implementation for signals sparse in the time, frequency and wavelet domains. We characterize the board in terms of its signal recovery quality in these sparse domains and show how various tunable parameters affect its performance. The analog front end consumes less than 20μA and yields recovery in excess of 30dB SNR consistently.

3.2 Compressed Sensing and Related Work

We first briefly describe the usual compressed sensing procedure: Assume that the signal of interest $x \in \mathbb{R}^n$ and a set of measurements $y \in \mathbb{R}^m,\ m \ll n$ are available to us, such that $y = \Phi x$, where $\Phi \in \mathbb{R}^{m \times n}$ is a sensing matrix. Then, under the condition that x is sufficiently sparse, the solution to the following combinatorial optimization problem recovers the signal exactly:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 0} \quad \text{s.t.} \quad y = \Phi \tilde{x} \tag{3.1}$$

where $\|x\|_{\ell 0} \triangleq |\{i : x_i \neq 0\}|$, the number of non-zero elements in x. Finding a solution to Equation 3.1 is NP-hard in general. Instead, a convex relaxation is proposed [CW06]:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 1} \quad \text{s.t.} \quad y = \Phi \tilde{x} \tag{3.2}$$

where $\|x\|_{\ell 1} \triangleq \sum_{i=1}^{n} |x_i|$. It is shown in [CRT06a] that under the sparsity condition and when Φ satisfies the so-called restricted isometry property (RIP), the reconstruction $\hat{x}$ is exact with overwhelming probability [CW06]. Practically, this means that if the signal is sparse in the sensing domain, then taking m measurements through a suitable linear transformation Φ will be sufficient to reconstruct the signal. If the signal is not sparse in the sensing domain, but in another known domain, the reconstruction must be performed in two steps. Assume a separate invertible linear transformation Ψ under which the signal is rendered sparse, $z = \Psi x$ where $\Psi \in \mathbb{C}^{n \times n}$. For example, if the signal was a single frequency tone, then its time domain representation z is not sparse, but with Ψ as the Inverse Fourier transform, x is sparse. The equivalent reconstruction procedure is then:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 1} \quad \text{s.t.} \quad y = \Phi \Psi \tilde{x} \tag{3.3}$$

An additional requirement for reconstruction to succeed is that Ψ be incoherent [DH01]

with the projection matrix Φ or equivalently that the combination $\Phi\Psi$ satisfies the RIP. This holds with high probability, for example, when the elements of Φ are independent realizations of a Gaussian random variable, $\mathcal{N}(0, \frac{1}{n})$ or of an equiprobable $\pm\frac{1}{\sqrt{n}}$ Bernoulli random variable, *regardless of* Ψ *(w.h.p.)* [CW06] .

Implementing randomized projections $y = \Phi x$ has been the subject of much research and in some domains, such as MRI, happens naturally through non-uniform sampling [LDP07]. In other applications, where the sensing domain is temporal or spatial, projections have to be performed explicitly. Figure 3.2 depicts the software approach taken by numerous prototype implementations [BG09, KMK11, CCS10] when sampling rates and power permit. As explained in Sec. 3.1, software CS does not provide ADC sampling rate reduction and its consequent savings in power.

Direct hardware implementations do exist as well. Some of the earliest demonstrations of CS were developed by Duarte, et. al. [DDT08] for compressive imaging using a single pixel. The randomized projections were accomplished using a digital micro-mirror array that was programmed with the random vectors. While it has limited use in natural photography, due mainly to aperture time requirements, certain non-visible light applications are quite promising [CMB08].

For time domain signals, Kirolos and Laska, et. al. [KLW06, LKD07, TLD10] introduced the concept of analog-to-information conversion that constitutes randomized demodulation (RD). This system entails multiplying the input signal with a random bit stream of $\pm$1s at a rate higher than the Nyquist rate of the input signal followed by a low pass filter of known impulse response. The compressed measurements are acquired by sampling the output of the low pass filter at regular intervals. While an elegant architecture, RD suffers from an important drawback. As Yu, et. al. [YHS08] point out, because the measurements are obtained by sampling the output of the analog filter sequentially, they are no longer independent due to the convolution in the filter.

Yu suggests a parallel structure that applies RD to the signal simultaneously across a number of branches that each use a different random bit stream for the multiplication.

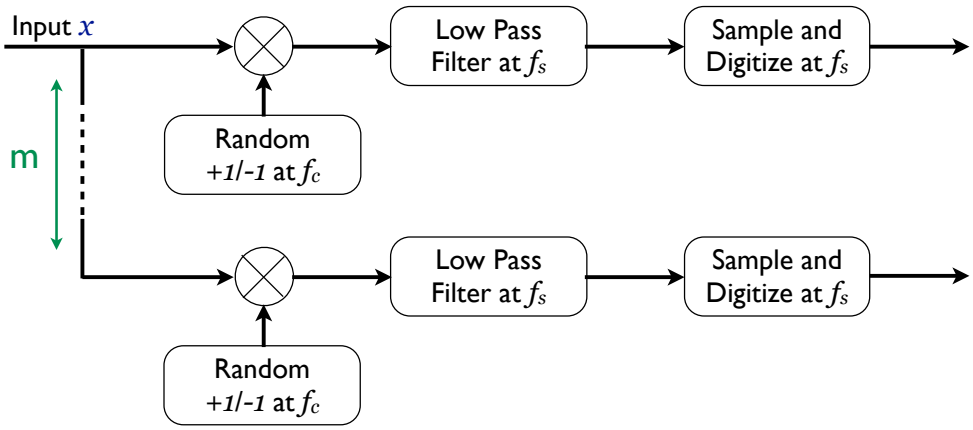

Figure 3.3: Hardware implementation of Compressed Sensing using parallel random demodulation needs m signal processing chains for m independent measurements.

The advantage is that each branch now produces independent compressed measurements but at the cost of immense hardware complexity. Mishali, et. al. [MED09, ME10] follow a similar architecture in their modulated wideband converter, except that they reduce the required number of independent measurement channels by assuming that the signal is structured in such a way that it only occupies some discrete bands in the frequency spectrum. A simplified representation of the parallel RD implementation is shown in Figure 3.3. Here, f_c is the ''chipping'' rate for the random demodulation and f_s is the sub-Nyquist sampling rate per channel. The discrete-time equivalent of the sampling matrix specified over a window of time T is of dimensionality $mf_sT \times f_cT$ with each row corresponding to one bit stream.

An interesting recent paper by Slavinsky, et. al. [SLD11] describes a novel architecture called CSMux that captures simultaneously multiple signals that are jointly sparse. The signals are multiplied each with an independent random bit stream and then summed before sampling at the Nyquist rate of one of the signals. In some ways, this is equivalent to the modulated wideband converter, except that CSMux requires that the signals be split, through band pass filtering, into individual frequency bands prior to processing. In all the above approaches, the requirement of multiple analog processing branches is evident even if fairly restrictive assumptions are made about the sparsity of the signal

and the domain it is sparse in -- Mishali's prototype consists of 4 independent branches. Scaling these systems to a large number of measurement channels while maintaining CS's universality property becomes impractical.

Furthermore, these architectures (with the exception of CSMux) require active analog components in each branch that each consume quiescent current. The multiplier can be handled passively [SLD11] but the low pass filters are constructed from high performance, low noise opamps. As the number of channels is scaled to increase the number of independent measurements, the quiescent current increases proportionally. For low power sensing applications, aggregate quiescent current can easily exceed the active power of the ADC, negating the benefits accrued through compressed sensing. The following section describes our CapMux architecture, which uses a single analog processing chain and scales to an arbitrary number of independent channels (of reduced bandwidth) with minimal hardware complexity.

3.3 The CapMux Compressed Sampler

The CapMux architecture is illustrated in Figure 3.4. CapMux uses time-multiplexing to amortize the quiescent current costs of one high performance integration opamp across the m measurement channels. Observe that the signal is multiplied with one random bitstream generated at mf_c instead of the m bitstreams at f_c in the parallel RD architecture shown in Figure 3.3. The ADC sampling the output of CapMux generates measurements at the rate of mf_s, which is equal to the combined outputs of the branches in parallel RD.

The CapMux system hinges on the operation of a multi-channel integrator. For simplicity, assume for now that the multi-channel integrator consists of a bank of m independent integrator blocks, only one of which is active at a time. The role of each hypothetical integrator is equivalent to that of the low pass filter in RD, in that it sums the values output by the random multiplier. The channel synchronization block selects one integra-

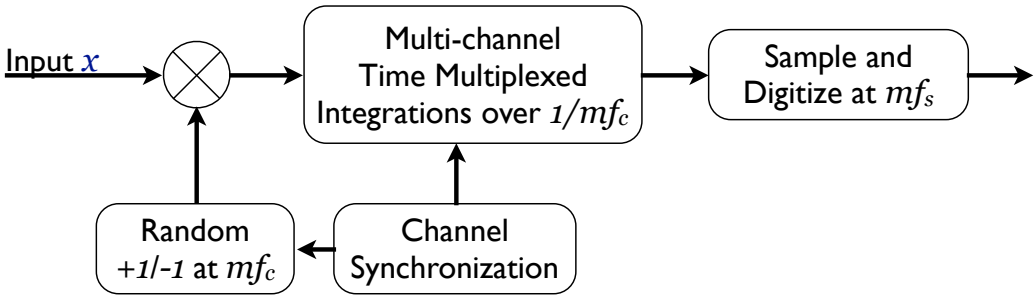

Figure 3.4: Hardware implementation of Compressed Sensing in CapMux using a single time multiplexed signal processing chain.

tor from this virtual bank in round robin fashion at the same rate as random bitstream generation. That is, each integration only lasts for one random multiplication, for a time slot τ ($= 1/mf_c$ in this case) long and each integrator channel only sees $1/m$-th of the entire bitstream.

One integration round over the m channels is accomplished in $m\tau = 1/f_c$ seconds and if the input signal does not change over this time period (this can be explicitly ensured or simply assumed for low bandwidth signals), the set of m integrations approximates compressed sensing projections over one column of Φ as shown by the blue downward arrow in Figure 3.1 or one inner loop iteration in the software CS approach shown in Figure 3.2. If this process is continued for $T = n/f_c$ seconds, the values accumulated in each hypothetical integrator constitute the m compressed measurements corresponding to $y = \Phi x$. After every T seconds, the integrator channels can be sampled by a low rate ADC and reset to start a new projection block. Mathematically, each integration operation can be written as:

$$y_{ji} = \int_{\frac{i}{f_c}+\frac{j}{mf_c}}^{\frac{i}{f_c}+\frac{j}{mf_c}+\tau} \Phi_{ji} x(t) dt \tag{3.4}$$

for row $j \in 1...m$ and column $i \in 1...n$ of sensing matrix Φ. For a constant input signal within the integration slot, this is simply $y_{ji} = \Phi_{ji} x_i \tau$ and across n integration rounds

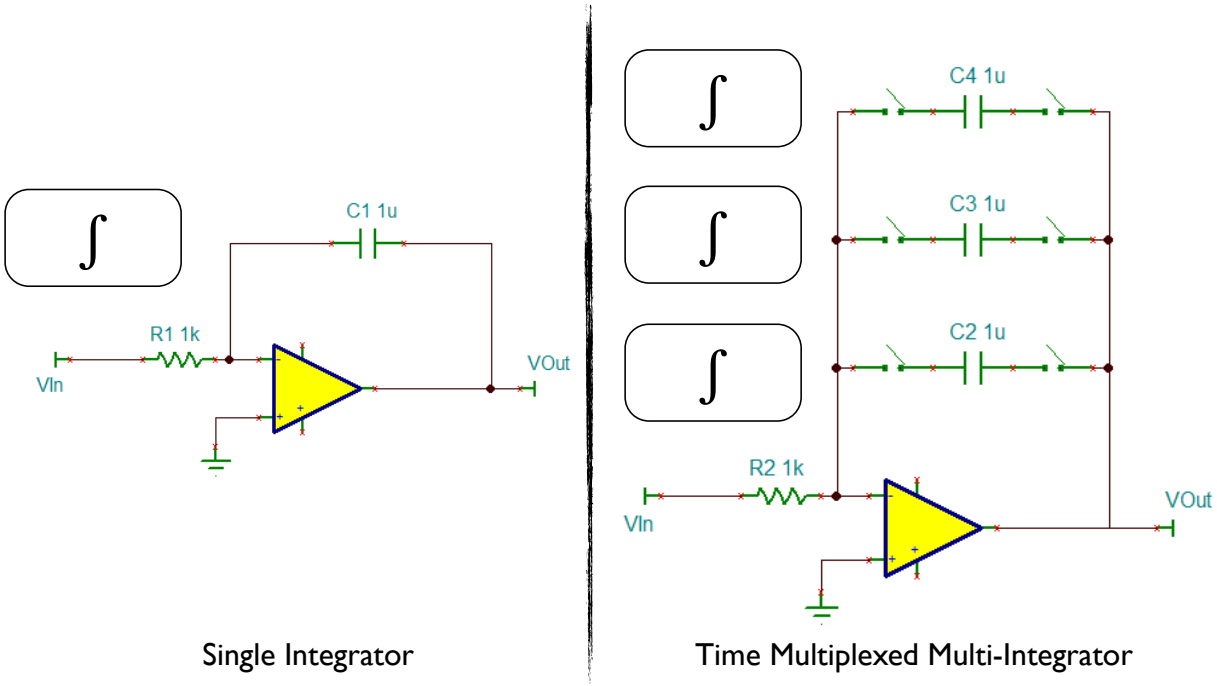

Figure 3.5: Implementing the time-multiplexed integration.

output of channel j would be:

$$y_j = \sum_{i=1}^{n} \Phi_{ji} x_i \tau = \tau \Phi_j x \tag{3.5}$$

the j-th row of the CS projection matrix with scaling τ.

3.3.1 The Multi-Integrator

The integrator is specially constructed to emulate integration of m channels independently. The design of the multi-integrator is shown in Figure 3.5. An active integrator is essentially an opamp in the voltage-to-current conversion mode that provides a charging current proportional to the input voltage to a capacitor in the negative feedback path, resulting in signal integration. The output of the integrator is given by $v_j = v_j(0) - \tau/(RC) \cdot x_i$ for channel j and time slot i of the input signal with an initial value $v_j(0) = Q_j(0)/C$, where $Q_j(0)$ is the charge stored in the capacitor of capacitance C before integration.

Bruton and Pederson [BP72] show how one opamp can be time-multiplexed to implement higher order filters. The multi-integrator uses this idea to share an opamp across m channels of integration. Since the state of the integrator is stored in the charge on the capacitor, a multi-channel integrator can be formed with a parallel bank of capacitors only one of which is switched in at any time. The channel capacitor switched in is the one actively integrating and by synchronizing capacitor switching to the random multiplication, compressed projections can be achieved. Note that this system is not only low power, because it expends little quiescent current, but is also low energy, because despite the time-multiplexing operation, the duration over which the signal is processed to produce m compressed measurements is the same as that of the parallel RD design. In contrast to parallel RD, scaling this design to a desired number of measurement channels only entails adding to the switched capacitor bank and modifying the random bitstream generator accordingly. As switches and capacitors occupy little circuit area and power, CapMux can be scaled arbitrarily (but with reduced bandwidth, as described next).

3.3.2 Practical Considerations

The key parameter that determines the performance of CapMux is the chipping rate, f_c. The chipping rate must be higher than the Nyquist rate of the input signal but the maximum achievable chipping rate in CapMux depends on the total integration time for one round (one column of the sensing matrix). With Bernoulli sampling matrices, the chipping rate is given by $f_c = 1/m\tau$. Channel count m and per channel integration time τ should, therefore, be kept as small as possible for high f_c. In the next section, we show how effective channel count can be reduced, but integration time is harder to reduce because it is governed by the speed (slew rate) of the opamp, the leakage characteristics of the switches and capacitors, and the overall noise of the circuit. A low τ would also mean a reduced integrated charge and a lower signal-to-noise ratio (SNR), unless the capacitors are small too. But, one must ensure that total switch parasitic capacitance ($\sim$2-5pF per switch) should be a tiny fraction of integration capacitance. At a given τ, the maximum

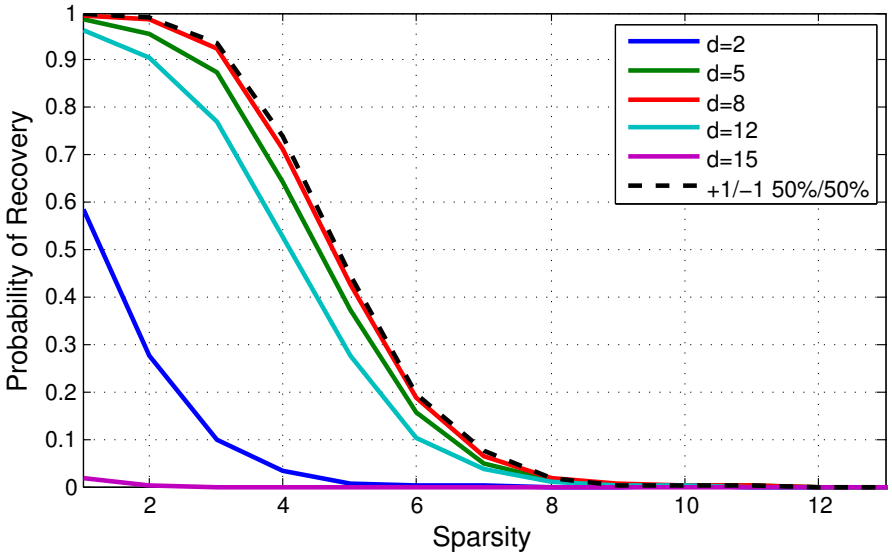

Figure 3.6: Empirical recovery performance with 16×64 sparse binary sampling matrices of varying density compared to traditional Bernoulli sampling matrices.

allowable signal bandwidth is given by $1/(2m\tau)$ for Bernoulli sensing matrices.

3.3.3 Sparse Binary Matrices

It has recently been observed that sparse binary matrices [LKR12] also admit recovery guarantees similar to that of Bernoulli matrices while maintaining the universality property. Two properties make sparse binary matrices (SBM) especially attractive. First, SBMs consist of 0/1s instead of ± 1s. Second, the number of 1s in each column is a small fixed number called the degree or density of the matrix with $d \ll m$, especially when m is large. The first property implies that an implementation no longer requires signal inversion (multiplication is realized by switching a pass through or inverted signal into the integrator [SLD11]) and the second property implies that only as many capacitors need to be switched in each integration round as the density. Since the density can be much lower than the number of channels, the total integration time is now $d\tau$ instead of $m\tau$ and the maximum allowable signal bandwidth is now given by $1/(2d\tau)$. Figure 3.6 shows the empirical recovery performance with SBM matrices of signals sparse in the time domain (the worst case scenario for SBM). For 16×64 matrices, a compression ratio of 4:1,

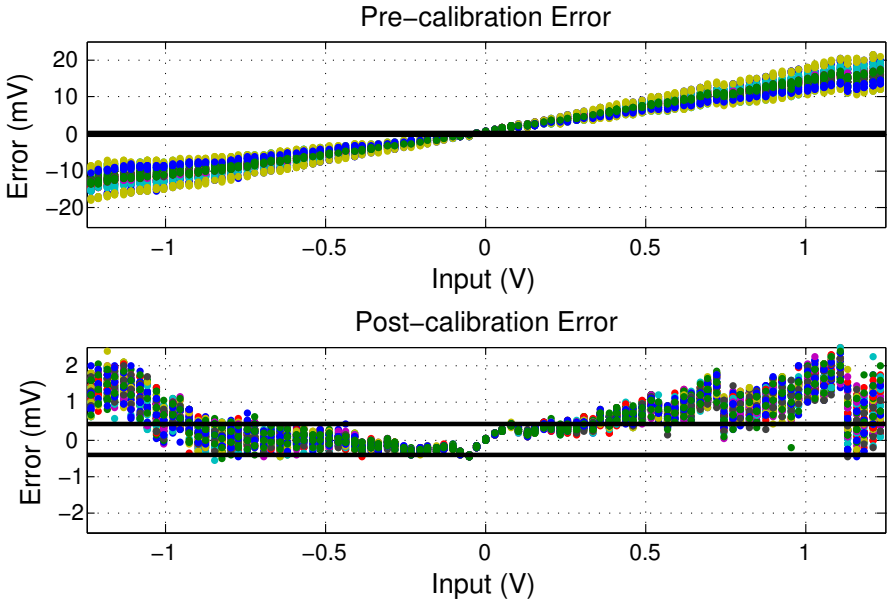

Figure 3.7: Error in mV between measured and expected values before and after calibration. Different colors indicate values from different channels. Solid black lines indicate inherent error due to 12-bit ADC quantization.

$d = 8$ performs as well as Bernoulli matrices, which implies an instantaneous doubling of chipping rate and allowable signal bandwidth. It can also be observed that density above half the number of channels ($d > 8$) deteriorates recovery performance. This is to be expected as the extra non-zeros in the matrix do not provide information diversity and rather deteriorates it. Sparse binary matrices have the drawback that since it only consists of positive values, the integration always grows over time (unless the input signal is bipolar), reducing the potential window size (the number of columns of the equivalent matrix).

3.3.4 Calibrating For Component Variability

Variability and non-ideal behavior of components causes the output of CapMux to stray from expectations (see top plot in Figure 3.7). While error can be reduced to a large extent by using high-cost high-precision components, it can not be completely eliminated. Furthermore, better performance in op-amps usually comes at the cost of higher quiescent

current resulting in an increased energy footprint.

Two factors affect CapMux performance significantly that require to be accounted for through calibration routines: capacitor variations and integrator offset. Integrator offset is caused by opamp input offset voltage that causes small charging currents even when the input is zero. Capacitor variations alter the transfer characteristics of the channels, manifesting as unequal integrator gains and offsets for each. If we express the output of the integrator as a function of these component variations and the input amplitude, x, we have

$$V_{out} = V_{os-m} + \beta_m G x \tag{3.6}$$

where V_{os-m} is the channel-dependent offset voltage at the output of the integrator, G is the nominal channel gain, and β_m is the per-channel gain scalar. By feeding a set of known input amplitudes and measuring the output, we apply linear regression per channel to yield the individual β_m scalars and V_{os-m} offsets. The effect that this calibration has on the output of the integrator is illustrated by comparing the error before and after (Figure 3.7). Note the change in y-axis between the top and bottom plots.

If we take samples across time during the integration of one channel across 64 samples, we see the benefit of having calibrated the system to non-idealities. Figure 3.8 shows the ideal and measured values of the integrator and the corresponding error for the compression of a 3-sparse signal using a sparse binary matrix of density $d = 8$. Despite noise, leakage, and component variation, the integrated signal remains within 5% accuracy of the ideal across all 64 samples.

3.3.5 Handling Switch Parasitics

In order to share access to a single integration path, care must be taken to adequately isolate each channel in time so as to minimize cross-talk. Cross talk occurs because of switch parasitic capacitances C_p that add up when the number of channels is high. C_p appears in parallel with the integrator channel capacitance C and gets charged by the

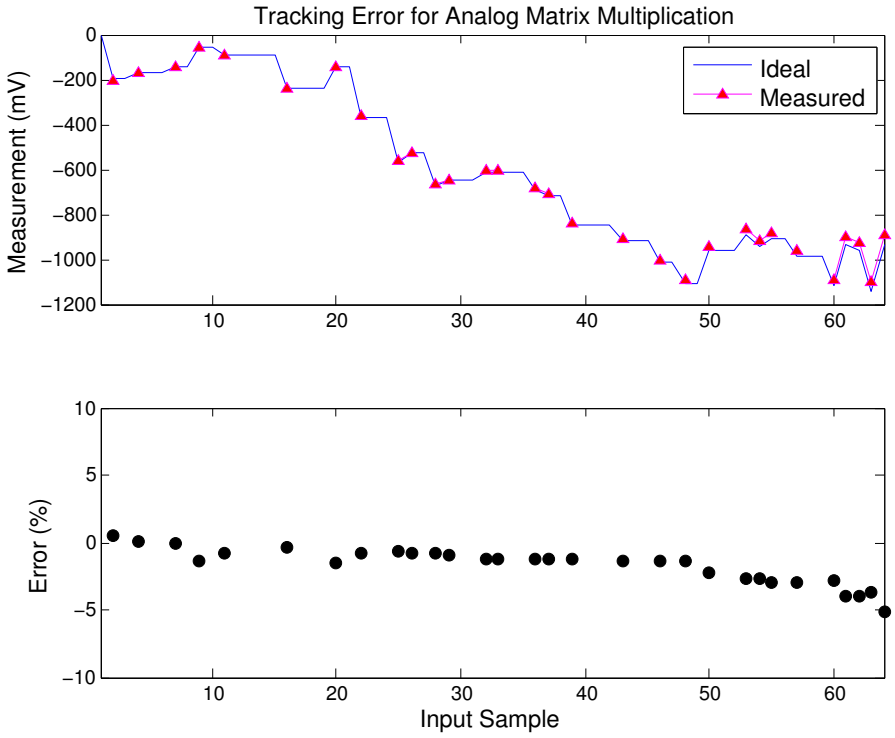

Figure 3.8: An example of per-column error for analog matrix multiplication after calibration, with density $d = 8$ and sparsity $s = 3$. Samples where the corresponding element of the sensing matrix is 0 are omitted.

same current. When the channel is switched to a different capacitor in the bank, the charge on C_p gets re-distributed to the new channel capacitance that gets switched in. To reduce the effect of cross-talk, the ratio C_p/C must be made very small (by increasing C), but this results in a high integration time τ, which reduces the bandwidth performance of CapMux.

Instead, subsequent to each integration, after switching out one channel and before switching in the next one, we first discharge the accumulated charge on the switch parasitic capacitance. This is done by setting the opamp input to zero and closing the negative feedback path. This condition is maintained for $\tau_p = V_{max}/SR$, where V_{max} is the maximum voltage that could be integrated and is equal to the voltage rail of the circuit and SR is the nominal slew rate of the amplifier in V/s. The effect of this change is a slight reduction in the chipping rate to $f_c = 1/(m(\tau + \tau_p))$.

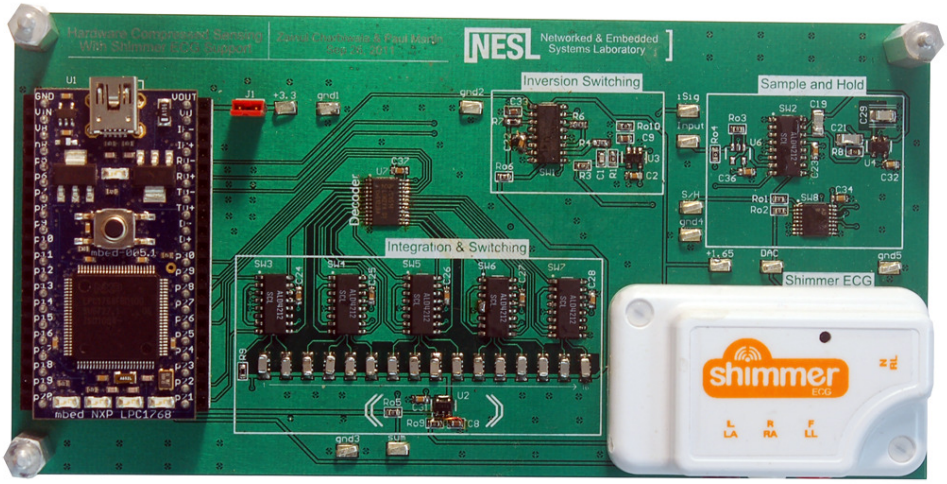

Figure 3.9: The 16-channel board with a ECG front end as application example.

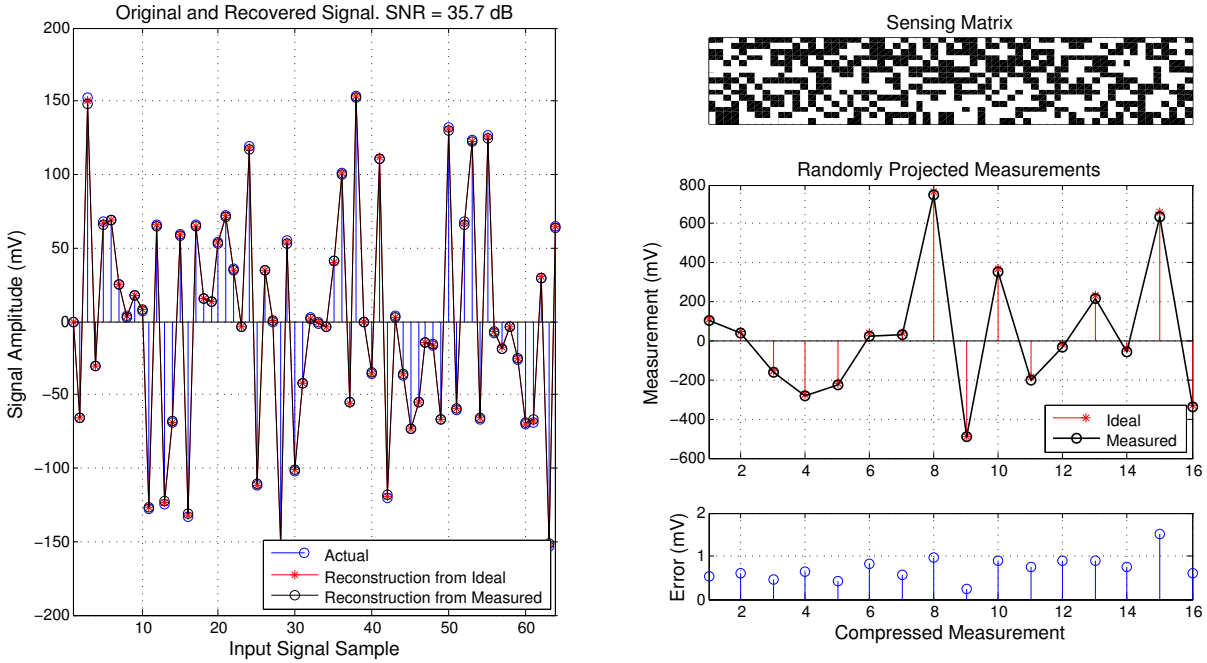

Figure 3.10: An example of compressed sensing and recovery of a 3-sparse signal in the frequency domain, with $d = 8$ using the CapMux hardware. The left plot shows the actual, ideally recovered, and experimentally recovered signal. The right side shows (from top to bottom) the sensing matrix (black = 1), the ideal and measured values after matrix projection, and the errors in mV of these projected values. The recovered signal has 35.7 dB SNR.

3.4 Evaluation

All results in this section have been measured from a 16 channel CapMux prototype as shown in Figure 3.9. Digital logic for random number generation and switch synchronization is executed by a microcontroller that also provides synthesized test signals. Figure 3.10 shows the result of acquiring and recovering a 3-sparse signal in the frequency domain. Overlaid are the actual, ideally-recovered, and experimentally-recovered signals, showing 35.7 dB of measured SNR, while the ideal recovery (with perfect analog components) would have yielded 41 dB. The right side of Figure 3.10 illustrates the sensing matrix used, the 16 measurements acquired and their error. Figure 3.10 showcases the power of CS -- a seemingly complex signal can be compressively sampled, and recovered with high accuracy given only the sparse domain and a random sensing matrix.

3.4.1 Universality

In order to verify that the CapMux system generalizes to all domains as per the theory that underlies it, we tested across time, frequency, and discrete wavelet domains, input signals of varying sparsity s, and sampling matrices of varying density d. For each s and d pair, we evaluate both the experimental SNR of the recovered signal from the CapMux hardware and the theoretical SNR under ideal conditions, given the timing and transfer characteristics of the system determined in calibration. The resulting SNRs and error bars bounding the first and third quartiles for all three domains are shown in Figs. 3.11, 3.12, and 3.13, respectively.

As shown earlier in simulation results in Figure 3.6, the probability of recovery has a maximum value when d is equal to half the number of channels, $d = 8$ in this case. Similarly, for $d < 8$ we observe an increase in the SNR of the recovered signal with an increase in the sensing matrix density for both the time and the wavelet domain. In the frequency domain, however, this does not hold. This is due to the nature of the bases involved -- the sampling domain (time) and the recovery domain (frequency) are perfectly incoherent. In other words, a sparse binary matrix of density greater than 1 adds little information beyond the first entry. Additionally, for frequency domain sparsity, the loss

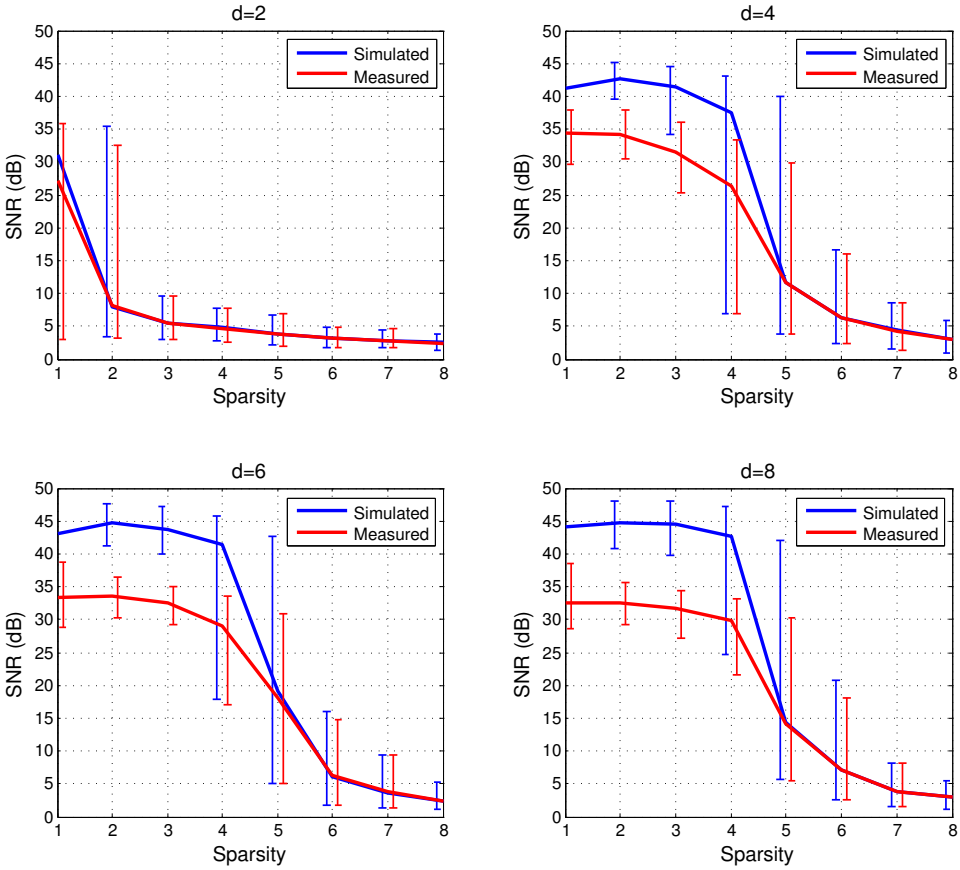

Figure 3.11: Simulated and measured SNR performance (median) for signals sparse in the time domain. Error bars indicate first and third quartiles. Increasing density of sensing matrix d improves the SNR for a given sparsity for both simulated and actual measurements.

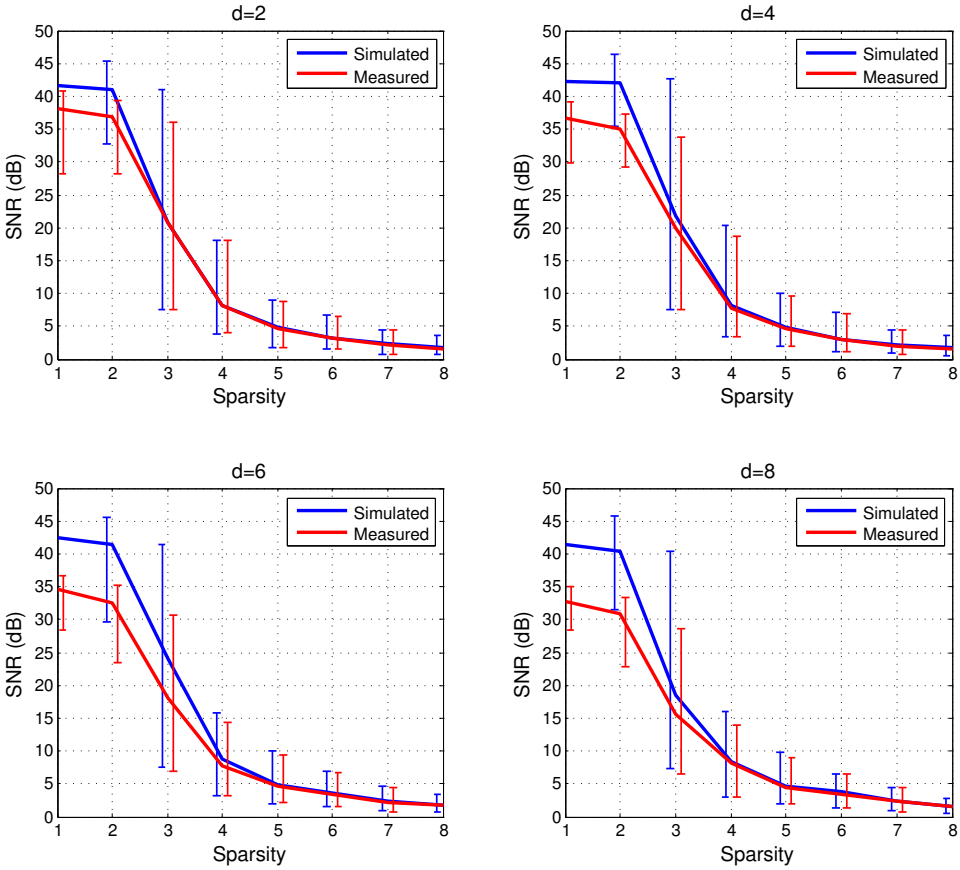

Figure 3.12: Simulated and measured SNR performance (median) for signals sparse in the frequency domain. Error bars indicate first and third quartiles. Increasing density of sensing matrix d provides no substantial benefit in simulated results and degrades the SNR somewhat in actual measurements.

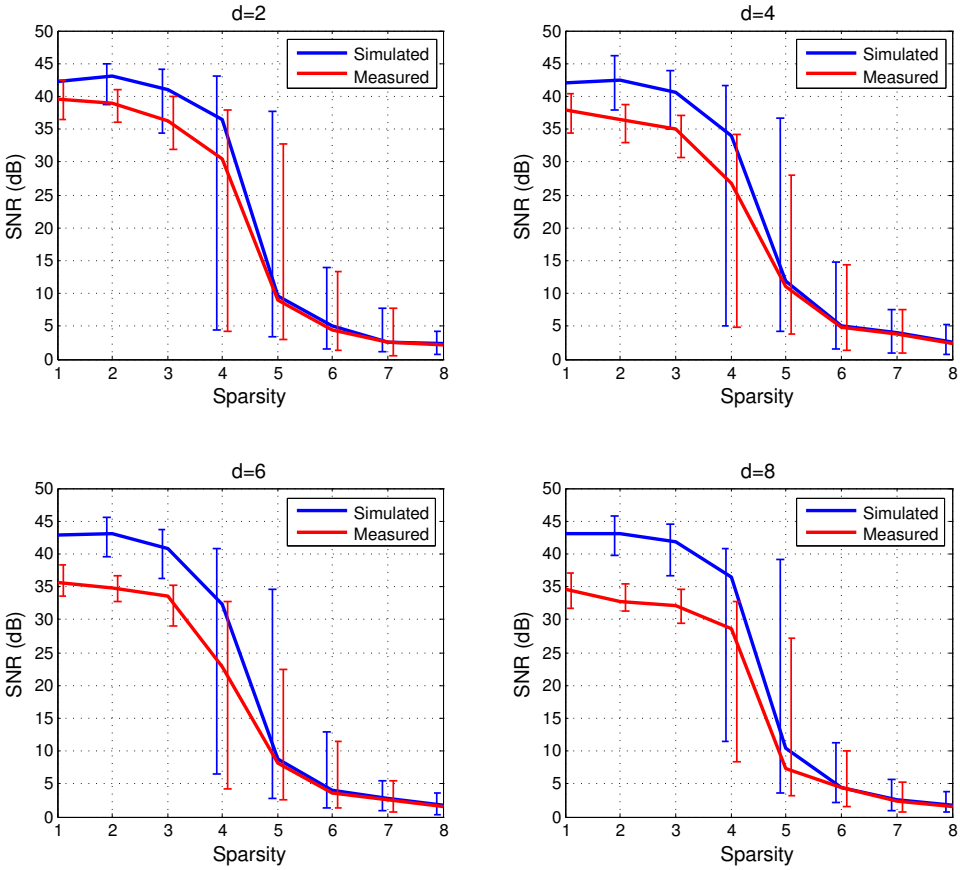

Figure 3.13: Simulated and measured SNR performance (median) for signals sparse in the wavelet domain (Daubechies-4). Error bars indicate first and third quartiles. Increasing density of sensing matrix d improves the SNR for a given sparsity for both simulated and actual measurements.

in SNR performance is low compared to simulated because the actual integrated values are high in amplitude. We find that that the SNR of the recovery is consistently high when the signal being integrated is also large.

In general, the SNR remains high ($>$ 20 dB) for signals of sparsity $s < 4$, barring sensing matrices of very low densities in the time domain, where the sampling and recovery domains are the same. In the time and wavelet domains, recovered signals have on average above 30 dB SNR for $s < 4$. Recovering signals where $s > 4$ with high fidelity would require greater than 16 channels of hardware. As we will see in the following section, the CapMux architecture can sample signals with much higher s values by scaling the number of channels with little or no reduction in signal bandwidth.

3.4.2 Energy Consumption

Energy consumed while performing compressed sensing in software can be divided into two regimes: sampling and digitizing the analog signal with an ADC and projecting the signal onto a lower dimensional basis via row-wise matrix multiplication. CS in hardware as with CapMux adds one additional regime -- that of the analog circuitry required to do the equivalent matrix multiplication in hardware -- and reduces the current consumption in the other two. The current demands of the hardware are quite low, and 85% of the total hardware current (20 μA) can be attributed to the amplifier used for the integration itself (17 μA), thanks to time-multiplexing. There is only a marginal increase in current due to transient loading of decoupling and parasitic capacitance when switching the capacitors for hardware compression. Figure 3.14 shows the CapMux hardware current for three compression episodes.

Though hardware compression does not require energy for the explicit multiplication and summation of arrays as required in software CS, it does require software control over the hardware switches. If a low power micro-controller is used, this requires waking the processor out of low power mode. This is also the case when performing software CS,

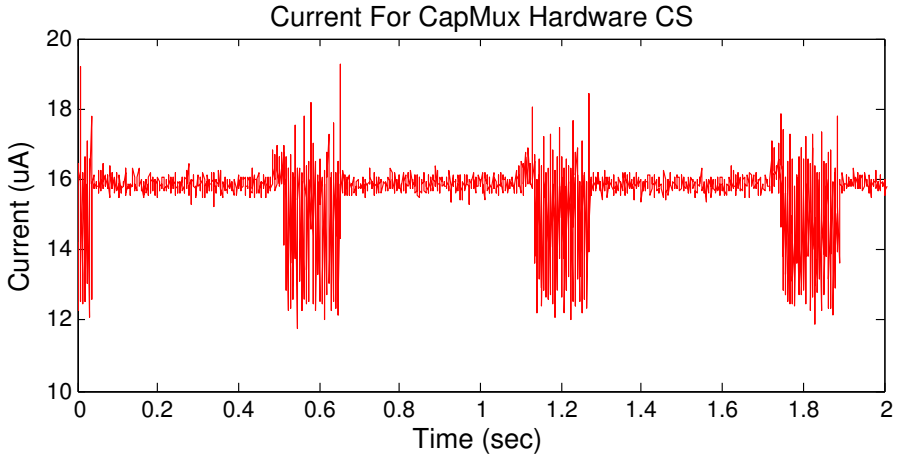

Figure 3.14: Measured current from the CapMux Hardware, with three episodes of compression. The idle current hovers around 16 μA while compression adds marginal current for transient loads.

though in that case the micro-controller is awake for a slightly longer period due to the burden of multiplying and summing the Φ matrix with x. Figure 3.15 shows the current required to wake up an MSP430G2231 micro-controller for a brief period to switch the appropriate hardware lines for CapMux. During the idle time between switching, the current is dominated by the quiescent current of the MSP430 -- around 1.2 μA -- while during switching the processor exits low power mode and consumes above 350 μA for a brief duration. Keeping this time short is paramount, because for each Δt decrease there is significant winnings over software CS where the luxury of forgoing the row-wise Φx projection does not exist.

As Figure 3.16 shows, the average current consumed during software-controlled switching is dependent on the density of Φ. Naturally, there is a trade-off between the energy consumed and the accuracy of recovery. This trade-off exists in software CS as well, because a higher SBM density requires more operations per ADC sample.

Finally, CS in hardware has a distinct advantage over CS in software in terms of energy used for analog-to-digital conversion. For the 16 channel CapMux, only 16 values need be converted to reconstruct a signal that would have taken 64 conversions for an equivalent software implementation. After a certain sampling frequency (around 10 Hz), the energy

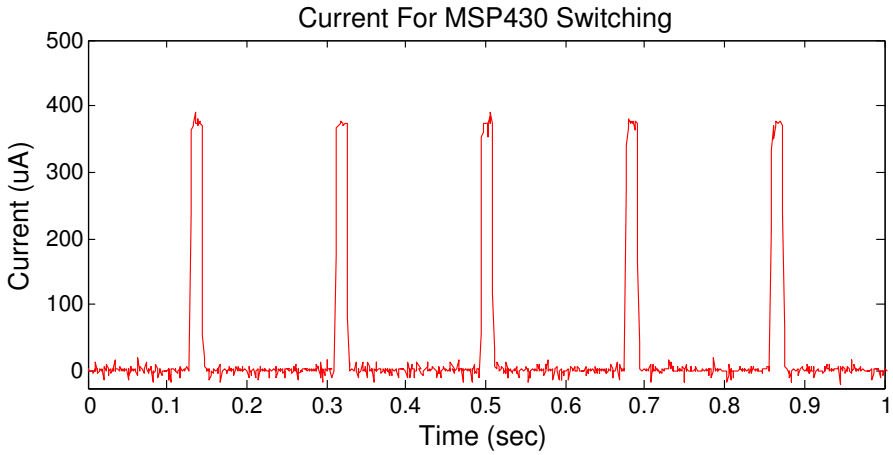

Figure 3.15: Measured current from an MSP430 emulating hardware switching. Idle current is around 1 μA while wake-up current is around 350 μA for a 1 MHz clock.

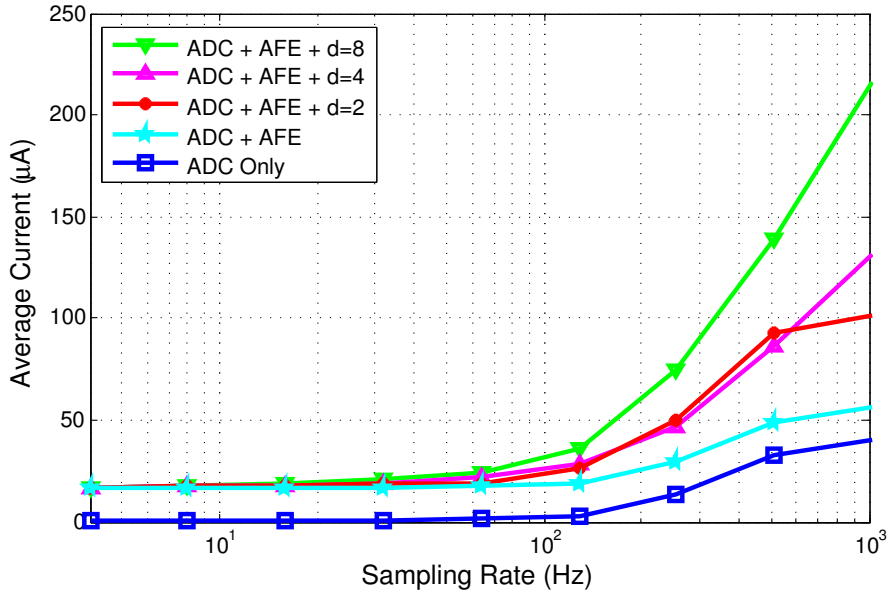

Figure 3.16: Average current consumption as a function of sampling frequency for ADC only, ADC with the CapMux analog front end (AFE), and ADC with both AFE and software control of switches for densities $d = 2$, 4, and 8.

per ADC conversion for the MSP430 becomes linear. Thus, 64 ADC conversions will take almost four times the amount of energy required to take 16 samples.

As the scale of the system increases, the power savings in terms of ADC becomes even more distinct. A 64 channel CapMux CS system will, for example, require only 64 ADC

conversions while the equivalent 64×256 software CS system will require 256.

3.5 Discussion and Future Work

Scaling and Sparsity

The results thus far have given credence to the validity of this architecture as a scalable hardware implementation of CS. The most natural extension of this work is to implement a system with more channels, allowing for the compression and reconstruction of higher dimensional ($s \gg 1$) signals with little to no increase in power consumption. As described earlier, adding more independent channels involves adding more switched capacitors to the multi-integrator and adapting the synchronization routines accordingly.

Similar to the empirical results of using a 16×64 SBM shown in Figure 3.6, Figure 3.17 shows results from a Monte Carlo simulation over 64×256 SBMs with varying densities. Note that densities of $d > 3$ result in a high probability of recovery, whereas that of the 16×64 matrix was at least $d = 5$. Additionally, the 64 channel results show a recovery probability above 90% for signals of up to sparsity $s = 12$ while the 16 channel version has 90% recovery for $s = 2$. This implies that sparse binary matrices of low density but higher channel count may be sufficient to acquire signals of longer durations and higher information content.

Bandwidth

The number of channels and density of the sampling matrix cannot be increased for free -- there is a trade-off for each in terms of the sampling rate and thus overall bandwidth of the system. Increasing the number of channels reduces the compression ratio of the system unless the duration T over which projections are being performed increases proportionally. Fortunately, SBMs of low density may be sufficient at high channel counts so the admissible signal bandwidth of $1/(2d\tau)$ remains approximately constant despite the

time multiplexed nature of CapMux. Since sparsity and signal duration may be linked (a longer signal may have more information content), one must be cognizant that a higher channel count is only beneficial when sparsity does not increase faster than the signal length.

The bandwidth of the system can be further increased by decreasing the integration time required for each element. This is achieved by reducing the capacitance per channel, at the cost of reducing SNR due to leakage effects and potentially requiring a higher-power integrating opamp to handle faster slew rates.

Further Considerations

- The results presented assume a DC signal as would be seen at the output of a sample & hold (S/H) circuit, though no such circuit was used. Preliminary results show that a S/H circuit can be implemented at the additional cost of 2 μA, but the effect that this would have on the bandwidth and compressibility of the input signal is not completely understood.

- In order to further reduce the power consumption of hardware CS, dedicated logic units such as an LFSR synthesized in a low power FPGA can be used to control the multiplexing of the integration circuit.

3.6 Conclusion

We have demonstrated a high fidelity, scalable architecture for compressed sensing in hardware. This architecture, CapMux, achieves high SNR for reconstructed sparse signals across arbitrary domains. Reconstructing the compressed signals comes at little expense in terms of analog hardware, consuming a mere 20 μA. What's more, scaling the CapMux architecture to a larger number of channels allows for accurate reconstruction of signals of higher dimensionality with little to no increase in the quiescent current of the analog

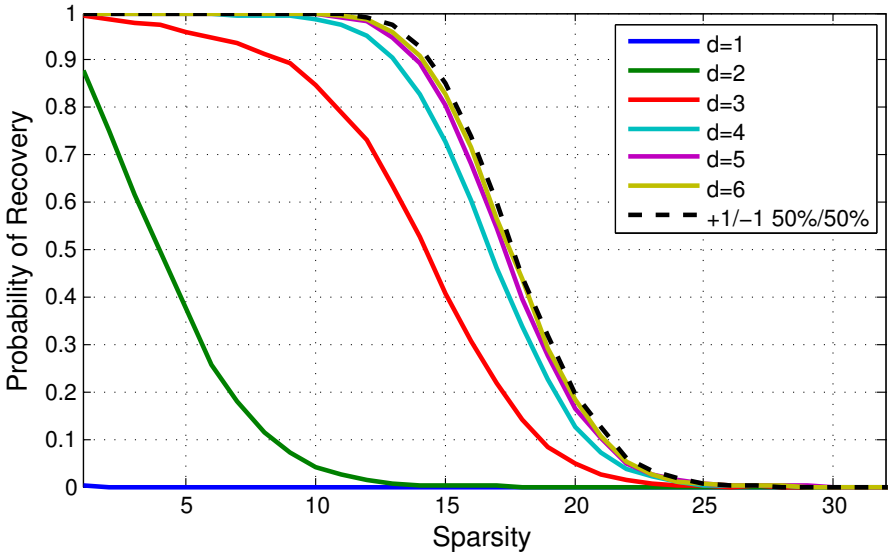

Figure 3.17: Empirical recovery performance with 64×256 sparse binary sampling matrices of low density compared to Bernoulli sampling matrices.

circuitry, due to time-multiplexed access of a single integration subsystem.

We have evaluated the power required to control the switches of the CapMux front end using a low power microcontroller such as an MSP430, showing how power scales with the density of the sensing matrix, Φ, and sampling rate.

Towards demonstrating the efficacy of the system as a scalable solution, we presented empirical results for a scaled version of the current CapMux circuit up to 64 channels. The 64 channel simulations suggest an almost six-fold increase in the ability to recover sparse signals without decreasing the sampling rate of the compressor.

CHAPTER 4

Neural Spike Compression with a Learned Union of Supports

4.1 Introduction

In this chapter, we move on the processing block of the sensing chain in the context of a wireless neural recorder. Neurons communicate with each other using electrical signals called ‘‘action potentials’’ or ‘‘spikes’’. Analyzing the action potentials generated by individual neurons may further our understanding of the functioning of the brain. Thus, many neuroscientific experiments and clinical studies record signals from neurons in the brain using implanted electrode arrays. In a traditional neural recording system, the acquired signals are transmitted outside the brain using wires, but this limits the freedom of movement of the subject, increases the risk of infection, and leads to motion artifacts in the recordings. It also precludes experiments that require recording in a socially enriched environment, in which subjects must be able to move unhindered. Wireless neural recording systems, either implanted inside the brain or attached to the scalp, promise to circumvent these issues but are subject to stringent power constraints. The power density of a neural recording ASIC needs to be significantly lower than $800\mu\text{W}/\text{mm}^2$, which is the power density known to damage brain cells [SHS98]. Besides, implanted ASICs are typically powered by inductive coupling [LLK10], limiting the amount of power that they

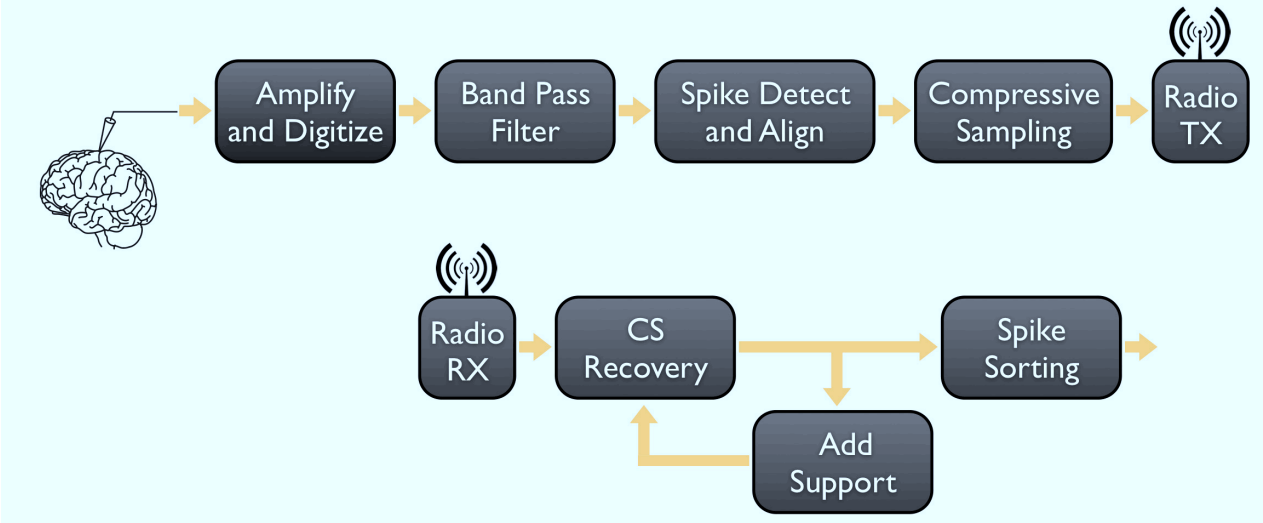

Figure 4.1: Schematic representation of our proposed compressive wireless neural recording system, with the top half *in vivo* and the bottom half *ex vivo*.

can consume.

The power consumption of a wireless neural recording system that transmits raw data is typically dominated by the radio transmitter. Because the power consumed by a given radio is directly proportional to the transmission rate, the data rate must be lowered in order to reduce the system power. Reducing the data rate also allows for simpler radio architectures, which will further reduce the power consumption.

Spike sorting, the process of assigning recorded action potentials to their source neurons, is a common analysis performed on acquired data that, when performed on-chip, can also be used to reduce the data rate [KGM09]. Transmitting the output of a spike-sorting chip---the spike features or IDs---would potentially reduce the data rate by two or three orders of magnitude compared to transmitting raw data, but it would mean that the full action potential waveforms would not be available for any other analysis. This may not be acceptable in studies that require the spike morphologies of the recorded action potentials in addition to the spike classification results. For instance, the individual spike waveforms can be used to characterize the type of neuron and possibly to distinguish between principal (primarily excitatory) and non-principal (largely inhibitory) cells for epilepsy studies. Other studies require spike widths (with varying definitions of widths),

ratios of ascending to descending slopes of the spikes [JRP07], etc. Neuroscientists often revisit previously recorded data to test new hypotheses, and thus may require different features to be extracted from the recorded spikes. Therefore, there is a need to reduce the transmitted data rate while allowing access to the recorded neural action potentials.

Compressive sensing (CS) is a recently developed theory that enables signal reconstruction from a small number of non-adaptively acquired sample measurements corresponding to the information content of the signal rather than to its bandwidth [CT05]. Information content or sparsity is quantified by estimating the number of the significant coefficients when the signal is projected into a space that accentuates its principal components. Therefore, if action potentials are sparse, compressive sensing would allow us to reduce communication costs and bandwidth compared to transmitting raw action potentials acquired at the Nyquist rate. Figure 4.1 depicts a schematic diagram of our proposed compressive neural recording system. Our implanted device would perform bandpass filtering, spike detection, and alignment on-chip to extract the action potential waveforms. Since spikes are ephemeral and rare, downstream resources are exercised only when a spike occurs. These spike waveform windows are then sequentially coded through a compressive sensing block and transmitted using a low-power radio.

Our spike recovery procedure uses a ''learned union of supports'' to enhance sparsity in the reconstruction. We observed that action potentials from different neurons have subtly different sparsity patterns or supports, and that supports of spikes from the same neuron are very similar. This is the basis for spike sorting---spikes have slightly different shapes depending on their source neuron. Recovering the signal using a weighted basis pursuit [CWB08, LV10a], where the indices of the weights are a union of the support sets of previously recovered spikes, results in higher signal quality. We will explain in Section 4.3 why this ensues. Specifically, we demonstrate that it allows us to achieve up to 17 dB higher SNDR (signal to noise-plus-distortion ratio) on real neural datasets when compared to the conventional basis pursuit for the same compression ratio. We analyze the power consumption of the wireless neural recording system and show that compressed sensing

can provide high-quality reconstructions of the spike morphology at a nominal increase in power when compared to sending only features for spike sorting [KGM09]. We verify that classification accuracies of up to 90% can be obtained by sending just 15 CS measurements per spike, which corresponds to a 60× data-rate reduction when compared to raw data transmission and a 3.2× reduction compared to raw action potential transmission.

4.2 Related Work

Although there have been several studies on compression techniques for EEG signals, there has been little investigation into compression techniques for the high-frequency action potentials. Craicun et al. reduced output data rates using vector quantization of the received action potentials [CCG11], while Kamboh et al. performed the discrete wavelet transform (DWT) on action potentials followed by coefficient thresholding to obtain data-rate reduction [KOM09]. Narasimhan et al. developed a method involving calculating the DWT for each spike and mapping each spike to a vocabulary of DWT signatures; data-rate reduction was thus achieved by transmitting only the signature ID of each spike [NTC07].

CS has been shown to be a promising technique for data-rate reduction of EEG signals [ACR09, CCS10, Avi07]. The use of CS for the compression of action potentials, however, has not yet been explored. Note that in [AO09], Aghagolzadeh and Oweiss did use CS to recover spike firing rates and demonstrated that CS is a viable alternative for reducing radio communication costs and latency. However, their method cannot be used for the action potential compression and reconstruction that is required for the reasons mentioned in Sec. 4.1. In this work, we will show that action potentials, when correctly extracted and aligned from the broadband neural signal, are sparse in the wavelet domain. We will then use CS with a learned union of supports reconstruction technique to achieve reconstruction of action potentials with high accuracy.

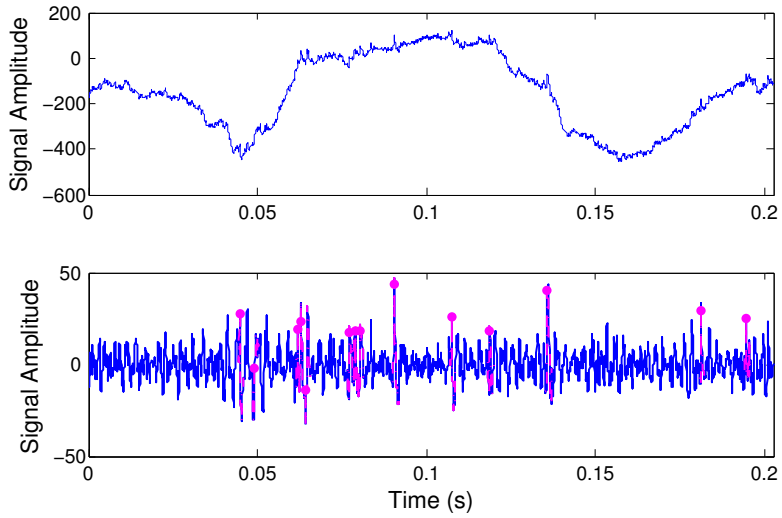

Figure 4.2: Top: A sample segment of unfiltered extracellular recording from a human subject. Bottom: Bandpass filtered (300 Hz--3 kHz) signal with markers indicating the detected spikes.

4.3 Spike Recovery Using a Learned Union of Supports

We begin by reviewing the conventional compressed sensing approach. Let the signal corresponding to an aligned spike waveform be $x \in \mathbb{R}^n$, where n is the length of the window within which the spike is completely contained. We apply ''post-sampling'' compressive sensing to this signal (referred to as software CS in Chapter 3) window to generate a set of m measurements for each spike $y = \Phi x$, where $m < n$ and $\Phi \in \mathbb{R}^{m \times n}$ is a sampling matrix that performs an arbitrary linear projection. The objective of our wireless neural recording system is to recover the stream of spikes from these compressed measurements transmitted over a low-power radio. For this recovery, we require the spike waveforms to be sparse or compressible in some domain. While many alternatives exist to sparsify action potentials ([NB04, CCG09, Owe06]), we use the discrete wavelet transform (DWT) with Daubechies filters for our analysis as it provides the best tradeoff in terms of computational requirements, compressibility and generalizability.

We assume that the spike is compressible in the DWT domain such that $z = \Psi x$ has few significant coefficients, where $\Psi \in \mathbb{R}^{n \times n}$ corresponds to the DWT operation.

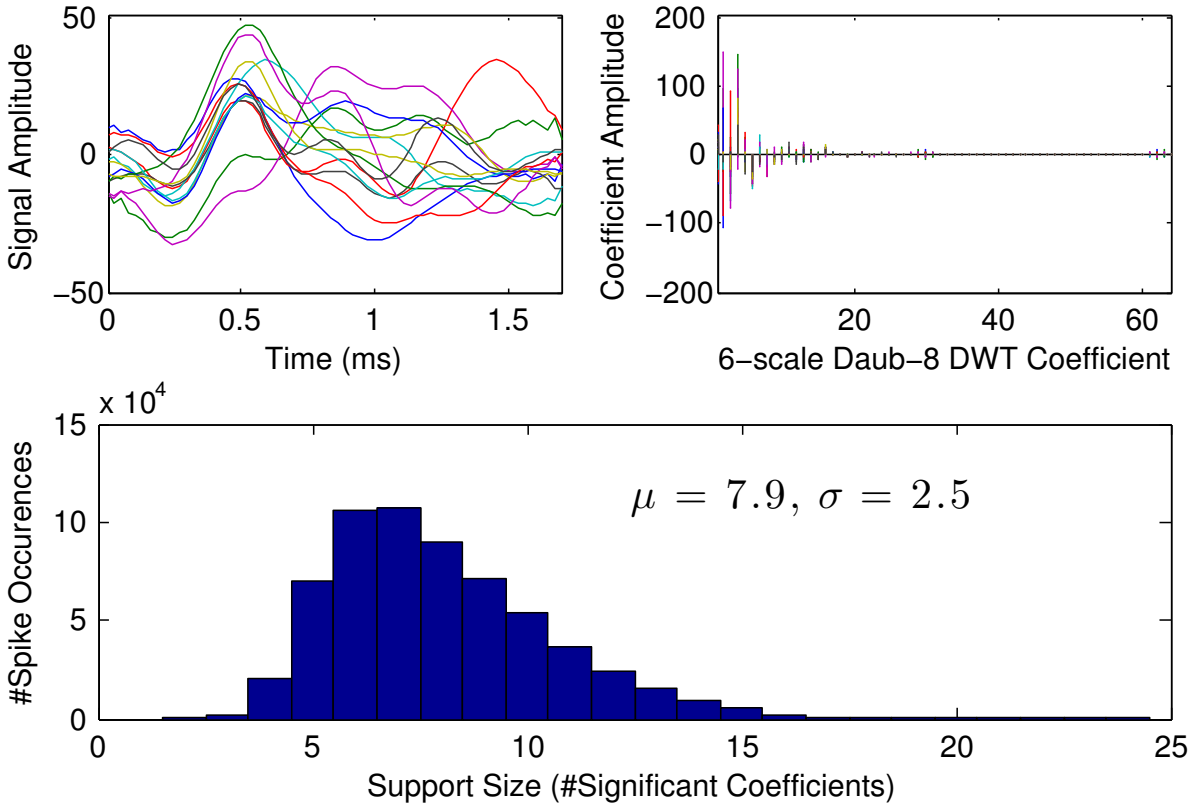

Figure 4.3: Aligned spikes and their DWT coefficients from signal segment shown in Figure 4.2 (Top). Histogram of the number of significant coefficients in the DWT domain of over 600,000 spikes extracted from human neural recordings (Bottom).

Significant coefficients are those that account for most of the signal energy. In particular, we locate the smallest set of DWT values that retains 99% of the ℓ_2-norm of the spike. We term this set of DWT coefficients as the spike's ''support''. That is, we find the smallest set T such that:

$$\text{supp}(z) = T \quad \|z_T\|_2 \geq C \|z\|_2 \tag{4.1}$$

where, z_T is an approximation of z with only the terms in the set T and $C = 0.99$.

Before proceeding, let us confirm that our sparsity assumption holds. Figure 4.2 shows a sample segment of unfiltered neural data from one of nine 40-μm diameter electrodes positioned in the hippocampal formation of a human epilepsy patient. As per Figure 4.1,

we use non-linear energy operator (NEO) (as in [KGM09, HYL09]) to extract and align spikes in a fixed window of 48 samples (Figure 4.2, bottom). The top half of Figure 4.3 shows the aligned spikes extracted from the segment in Figure 4.2 and their computed DWT coefficients, showing the large number of near-zero values. A macroscopic view of sparsity is provided in the bottom half of Figure 4.3, which plots the histogram of the size of the support computed over 600 000 spikes extracted from the entire dataset. About 8 coefficients describe the action potential adequately, a compression of 6:1 from the raw 48 samples acquired per spike.

We can now formulate our recovery procedure as the basis pursuit de-noising (BPDN) [CT05] problem:

$$\hat{z} = \underset{\tilde{z}}{\operatorname{argmin}} \frac{1}{2} \left\| y - \Phi\Psi^{-1}\tilde{z} \right\|_2^2 + \lambda \left\| \tilde{z} \right\|_1 \tag{4.2}$$

where, λ sets the significance of the sparsity with respect to the first noise tolerance term. It has been shown that when the ensemble matrix $\Phi\Psi^{-1}$ satisfies a condition known as the restricted isometry property (RIP) [CT05], the error in the solution to the above problem will be stable and bounded with overwhelming probability.

In [LV10a], Lu and Vaswani introduced a new approach to BPDN called Modified-CS when additional knowledge is available. Specifically, they show that if the support of the spike waveform (or a part thereof) was known a priori, the error in the solution to Equation (4.2) admits a lower bound. Their modified BPDN approach is given by:

$$\hat{z} = \underset{\tilde{z}}{\operatorname{argmin}} \frac{1}{2} \left\| y - \Phi\Psi^{-1}\tilde{z} \right\|_2^2 + \lambda \left\| \tilde{z}_{T^c} \right\|_1 \tag{4.3}$$

where, T^c is the complement of the known support so $\tilde{z}_{T^c}$ denotes the elements in $\tilde{z}$ that are not included within T. The bound on the ℓ_2 norm of the solution error depends on λ and T, but also Δ -- the part of the support that is unknown and Δ_e -- the part of known support that is incorrect. If the true support of the signal can be denoted by N, the relationship between these sets of supports is $N = T \cup \Delta \backslash \Delta_e$. Lu [LV10a] demonstrated

that as the size of Δ reduces, the solution error decreases dramatically, especially at high compression ratios (i.e. $n/m \gg 1$). An intuitive way of looking at modified BPDN is that it searches for a solution that sparsifies the non-significant coefficients of the signal since these would have lower energy than all the coefficients considered together.

The question of how one would estimate the support T is addressed by considering that real world compression problems typically need to recover a stream of signals (in non-overlapping windows of samples) and that the signal model evolves very slowly over time. In [LV10a], they exploit this fact by approximating the support of the k-th window with the support of the $(k-1)$-th window. For our system, this would equate to using the support from the previously recovered spike for the current one being reconstructed. This works well for MRI images as shown in [LV10b], but fails for neural action potentials. We conjecture that the reason for this stems from the fact that an implanted electrode usually picks up activity from multiple neurons in its vicinity. Since the morphology of the action potentials depends on the source neuron and neurons may fire in any order, this approximation is inappropriate.

Ideally, we would have liked to learn the support of each unique morphology discovered at an electrode and switch supports to the one being recovered. This would ensure that even if there are multiple models of the signal being recovered, the correct model would be used during reconstruction. While learning the different supports over time is quite feasible, knowing which spike support to use is impossible without computationally expensive encoder involvement. Instead, we propose performing a set union over the learned supports and furnishing Equation (4.3) with this set as T. The learning is continuous as the support of any newly recovered spikes is added to the union.

For completeness, we outline the procedure as follows: The decoder is initialized with an empty union set, $T^{(0)} = \emptyset$. When measurements for the first spike are received, the decoder uses Equation (4.3) to recover it. With an empty T, this is equivalent to using conventional BPDN Equation (4.2). After recovery, the support from this first reconstructed spike becomes the updated union set, $T^{(1)} = \text{supp}(\hat{z}^{(1)})$. When measurements

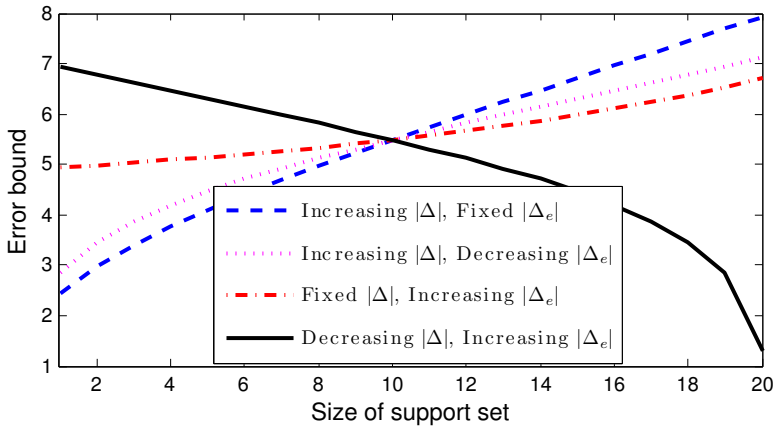

Figure 4.4: Trend lines of the error bound that trade off the size of the unknown support, Δ with the size of the superfluous support, Δ_e. The curves illustrate the sensitivity of the error bound on $|\Delta|$ and the relative insensitivity to $|\Delta_e|$.

for the second spike are received, the decoder performs modified BPDN using Equation (4.3) with the newly formed support set. Afterward, the support for this second reconstructed spike is computed and added to the union set, $T^{(2)} = T^{(1)} \cup \text{supp}(\hat{z}^{(2)})$. This process is repeated for subsequent spike measurements.

It may not be immediately clear why this learned union of supports technique is better suited to neural action potential recovery. The careful reader may even find it detrimental to the reconstruction process. On one hand, adding the supports of all spikes we encounter to the union set might expand it quickly to include all possible coefficients, effectively nullifying the contribution of the second term in Equation (4.3). And on the other hand, if we consider that modified BPDN is a weighted basis pursuit with the weights or penalty for the components in T being neutralized, any measurement noise present at those components will get amplified in the result.

We provide two justifications for our union of supports proposal, one derived analytically and one empirically from our datasets. For the first, we excerpt the bound on modified BPDN reconstruction error from [LV10a]:

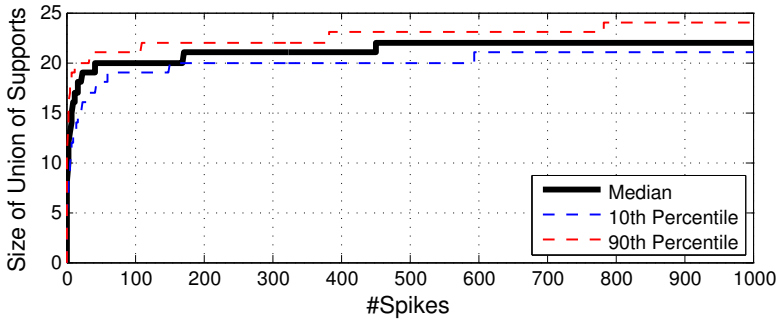

Figure 4.5: Median progression of the size of the learned union of supports over each set of 1000 spikes.

$$\begin{aligned} \|z - \hat{z}\|_2 \quad \leq \quad & \lambda\sqrt{|\Delta|}\sqrt{\frac{\theta^2_{|T|,|\Delta|}}{\left(1-\delta_{|T|}\right)^2}+1}\frac{1}{1-\delta_{|\Delta|}-\frac{\theta^2_{|T|,|\Delta|}}{1-\delta_{|T|}}} \\ & + \frac{\|\sigma\|_2}{\sqrt{1-\delta_{|N\cup\Delta_e|}}} \end{aligned} \tag{4.4}$$

where $|\cdot|$ refers to set cardinality, δ_s is the s-restricted isometry constant constant [CT05] for the matrix ensemble $\Phi\Psi^{-1}$ and $\theta_{s,s'}$ is the s, s'-restricted orthogonality constant [CT05] for the same. The second term relates to the signal noise, σ. To show that the union of supports is superior from an error bound point of view, we will need to prove that the value in Equation (4.4) is lower using our technique.

Assume that the union set at the k-th spike is T^u but that the true support of the spike is N. Let the support set from the preceding spike be denoted as T^p. We further assume that a spike with a similar morphology was previously recovered such that $|N \backslash T^u| \ll |N|$. Recalling that $N = T \cup \Delta \backslash \Delta_e$, we have $|\Delta^u| \ll |N|$ (because post-learning, the size of unknown support is fairly low) and $|\Delta^u| \leq |\Delta^p|$ (because the union set includes support of the previous spike plus all spikes in history, the unknown set for union support is guaranteed to be less than or equal to the unknown set for the preceding spike), since the spike morphologies of two consecutive spikes may be different. However, the cost of having an all-inclusive union is that elements of other spike morphologies learned over

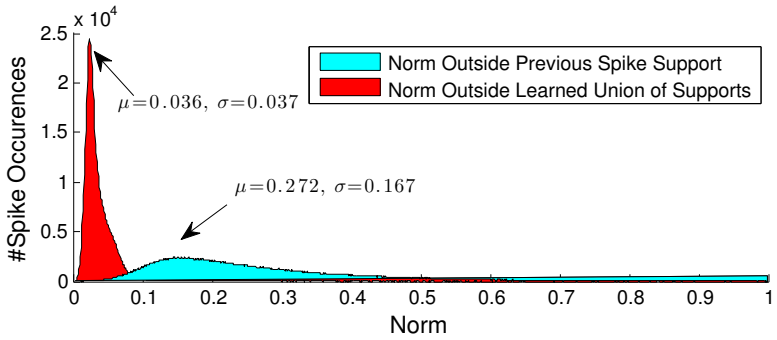

Figure 4.6: Histogram of the norm of signal outside support of previous spike and outside learned union of support of all preceding spikes.

time contribute to Δ_e and inflate $|T|$. That is, $|\Delta_e^u| \geq |\Delta_e^p|$. We therefore need to show when the trade off that exists in Equation (4.4) between the size of Δ_e and the size of Δ is in favor of a larger Δ_e instead of a larger Δ. Figure 4.4 plots the value of the error bound and depicts this trade off with an example set of parameters sweeping the size of the two sets independently and simultaneously. The absolute values illustrated in the curves are not important but rather the observation that the error bound is quite sensitive to the size of Δ but rather insensitive to the size of Δ_e. The principal reason behind this is the $\sqrt{|\Delta|}$ term that appears in the first term of Equation (4.4). This result argues that the seemingly conservative approach of adding all spike supports to a union set may be beneficial if the sizes of the unknown support Δ and the superfluous support Δ_e can be controlled. There is also a key point where the curves meet that emphasizes the tipping point between the sizes of the two sets.

Figure 4.5 shows the progression of the size of the union support set T over our datasets. In order to control the inflation in Δ_e, we reset the union support set to an empty set about every minute (or 1000 spikes). The figure plots the median of $|T|$ over each set of 1000 spikes and shows that a plateau is reached in the support size very rapidly. The 10% and 90% percentile curves emphasize that this result is consistent across all the datasets we have considered. A final rationale behind the intuition of our technique is shown in Figure 4.6, which plots the distribution of the normalized error between a spike and its approximation when the support of the previous spike is used and when the union support

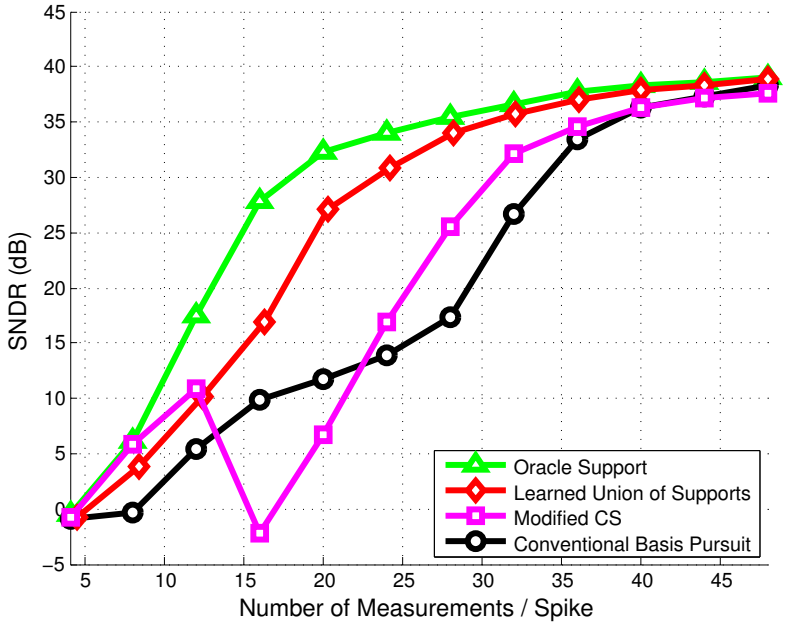

Figure 4.7: Performance comparison of spike recovery using the conventional basis pursuit, using support from the preceding spike, and using a learned union of support of all preceding spikes. Points represent median SNDR over all datasets.

is used. Notationally, this figure plots the PDF of $\|z - z_{T^p}\|_2 / \|z\|_2$ and $\|z - z_{T^u}\|_2 / \|z\|_2$. If we consider that the intuition behind modified BPDN is to enhance sparsity by lowering the energy outside T, we note that the union support technique improves this value by a factor of 8 in the mean across the datasets and a factor of 5 in the standard deviation. The following section demonstrates that the signal quality is consequently higher.

4.4 Results and Discussion

In order to evaluate the accuracy of CS reconstruction, we computed the median SNDR over more than 600 000 spikes from human electrophysiological recordings. The SNDR versus the number of CS measurements for different reconstruction methods is shown in Figure 4.7. This plot shows that the signal recovered using the union of supports method has an SNDR that is on average 6.7 dB greater (maximum 17 dB greater) than the SNDR of the signal recovered using the conventional basis pursuit recovery for the same

number of CS measurements. We can also see that a signal reconstruction with 20-dB SNDR is possible with only 18 CS measurements per spike (where each spike originally had 48 samples). This implies that the data rate is 2.6-times lower than that required for transmission of detected action potentials (up from 1.65× for conventional basis pursuit) and 53-times lower than the data rate required for transmission of raw data (assuming a spike firing rate of 30 Hz). The plot also shows the performance of Modified-CS [LV10a] and an oracle scheme that knows the exact support for recovery. Modified-CS shows some improvement over BP at higher data rates with a seemingly strange drop around the 16 measurement mark. This artifact occurs because the support recovered from each spike is the same until about 12 measurements per spike. This, in turn, is because CS picks the strongest support first and the spikes, though different, are similar enough that their top coefficients are the same. For Modified-CS, the mean improvement in SNDR over conventional basis pursuit is 1 dB with a maximum of 8.2 dB at 28 measurements per spike. It is interesting to note that union support comes fairly close to oracle support, with a seemingly simplistic support selection rule. Statistically, union support was only 2.8 dB off from the oracle on the average.

Besides SNDR, it is also important to evaluate the performance of reconstruction in terms of classification accuracy (CA) of the acquired action potentials. Toward this purpose, we clustered the action potentials using the Osort spike-sorting software package [RM06]. Figure 4.9 shows the color-coded clustering results for the recorded data along with the mean spike shape for each cluster. The clustering process was repeated for the signals reconstructed from a different number of CS measurements, ranging from 4 to 48, using each of the three reconstruction methods. The classification accuracy of the reconstructed spikes was computed by comparing the clustering results for each case with the clustering results of the original action potential waveforms. Figure 4.8 shows the median classification accuracy over the entire set of spikes analyzed for each of the three reconstruction methods. We find that the union of supports technique provides a higher classification accuracy than conventional basis pursuit. The classification accuracy for

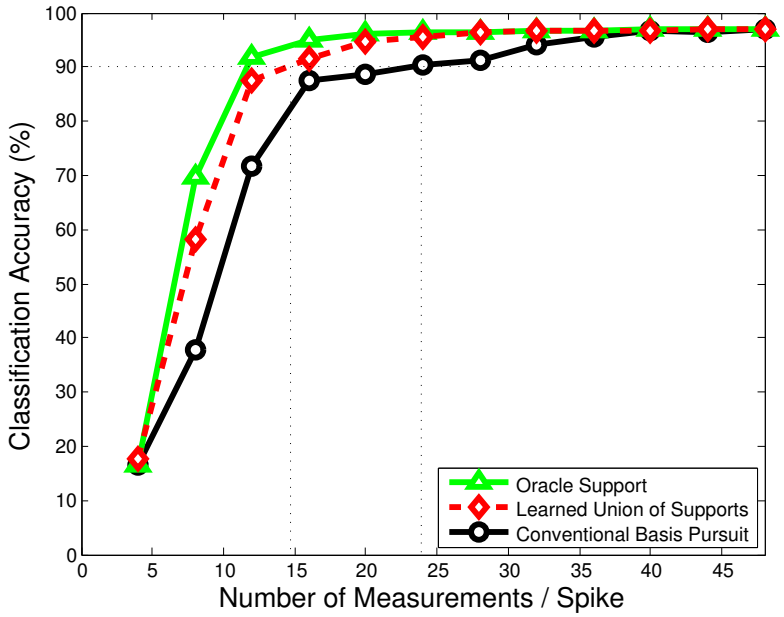

Figure 4.8: Median classification accuracy over more than 600 datasets (1000 spikes per dataset) versus number of CS measurements for conventional basis pursuit, union of supports reconstruction and when using an oracle.

union of supports reconstruction reaches 90% at 15 CS measurements, after which the the classification accuracy increases very slowly with the number of measurements. This relationship between the CA and the number of measurements follows from the behavior of CS reconstruction, which is grossly inaccurate until it reaches a factor proportional to the signal sparsity. Beyond this point, the reconstruction is accurate and improves only slightly with an increased number of measurements. For 24 CS measurements, a median classification accuracy of more than 95% is achieved.

Let us now look into the power savings that this approach can provide. The relationship between the power consumption and the output data rate stems from the trade off between the digital processing core and the radio for a given spike firing rate. In this analysis, the power consumption for digital processing is obtained from synthesis reports for a 90-nm standard-V_T CMOS process. The power estimates for the radio are based on a IEEE 802.15.4.a-compliant radio from Decawave [Dec11]. Figure 4.10 shows a plot of the estimated system power consumption for various system outputs at various spike firing

rates. It can be seen from this plot that transmission of 24 CS measurements provides a 2-times reduction in the system power compared to the transmission of the detected action potential waveforms. If we use 12 CS measurements, for a CA of 85% and an SNDR of 10 dB, the power can be reduced by 3-times compared to the transmission of raw spikes. For spike firing rates below 80 Hz, transmitting CS measurements has the lowest power among the various options considered in Figure 4.10. We have observed that the spike firing rate at a given electrode rarely exceeds 50 Hz, which makes sending CS measurements the most power-efficient choice. The power for transmitting 24 CS coefficients is approximately the same as transmitting 21 discrete derivative features [KGM09]. It should be noted that among all these options, sending CS measurements is the only option that allows accurate reconstruction of the action potentials.

As mentioned in Section 4.2, several alternative approaches to CS have been proposed in literature. A straightforward comparison of these references with our work is not possible because each study used different reconstruction accuracy metrics, different technology nodes, and different datasets. CS has been implemented in 90nm CMOS with a power consumption of 1.9 μW/channel [CCS10], which is lower than the power required for all other data-rate reduction techniques. However, the DWT demonstrated in [KOM09], which consumes 95 μW/channel in a 500-nm CMOS process, may be a competitive alternative to CS---especially when implemented in advanced technologies with aggressive supply-voltage scaling.

4.5 Conclusions

In this chapter, we evaluated the effectiveness of using compressed sensing for transmitting action potentials in wireless neural recording systems. We showed that the action potentials are sparse in the DWT domain. We proposed a union of supports technique for CS recovery and showed that it provides an average SNDR improvement of 6.7 dB and a maximum SNDR improvement of 17 dB compared to conventional basis pursuit

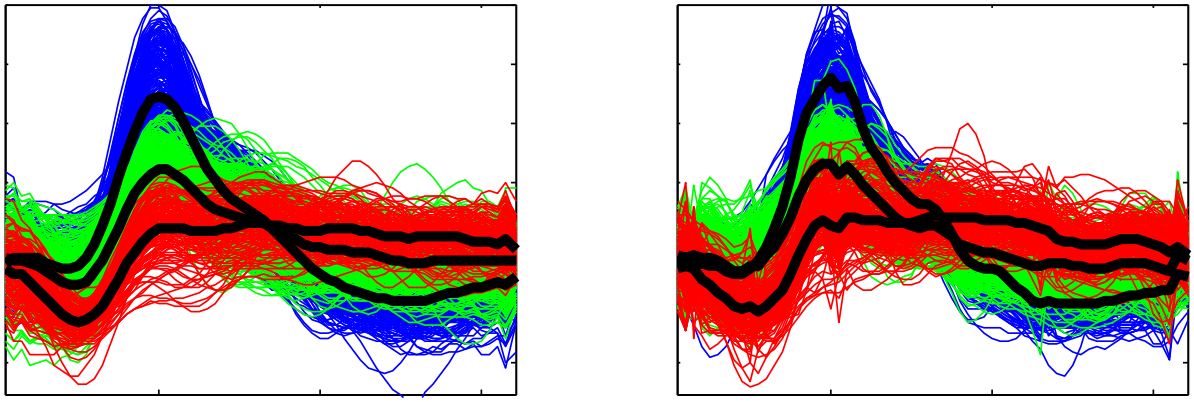

Figure 4.9: Clustered spikes before and after compressive recovery (16 measurements). Each spike has been color-coded according to the cluster to which it was assigned. The cluster mean waveforms are shown in bold.

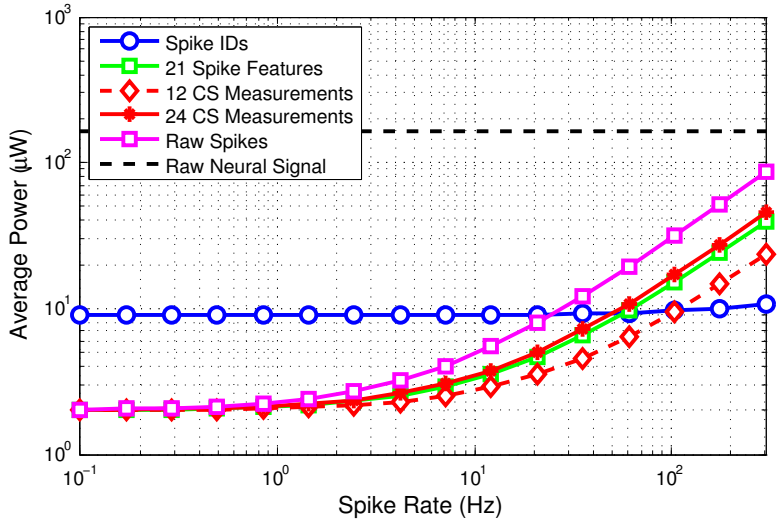

Figure 4.10: Average power consumption per channel for various system designs. Note that full spike morphology is preserved only for the systems represented by the bottom four curves (12 CS Measurements, 24 CS Measurements, Raw Spikes, and Raw Neural Signal).

reconstruction. We also showed that CS with union of supports can be used to provide a 2× reduction in the output data rate and in the system power consumption compared to transmission of raw action potentials. At the same time, sending 24 CS measurements provides a 20-dB SNDR for the reconstructed signal, which corresponds to a 95% classification accuracy for the spikes. A 3-times reduction in system power can be obtained if 12 CS measurements corresponding to a 10-dB SNDR and an 85% classification accuracy are transmitted. Transmitting CS samples is the most power-efficient way to reduce the data rate for spike firing rates below 80 Hz and is the only solution that allows access to individual recorded action potentials.

CHAPTER 5

Optimizing Sampling in an Interference Prone Environment

5.1 Introduction

We shift our focus in the next two chapters to data transport and show how the sampling stage could be leveraged to improve the effectiveness (this chapter) and the robustness (next chapter) of the data delivered to the inference engine. A key purpose for wirelessly networked sensors is for experimentation that furthers our understanding of the natural world and for the estimation and detection of various events within it [ACT, CPG, LRM, VBN]. In the first case, a key metric of performance is the *fidelity* with which a spatio-temporal phenomenon is recovered, while in the second, it is *reliable* detection that is important. In both scenarios, designers strive to ensure that the highest quality data is extracted from the network and it would seem intuitive then, that these metrics should somehow be considered within the data collection process. However, designs for network services in sensor networks have traditionally focused on lower layer metrics such as throughput (or goodput) and fairness [CFX07] and do not capture application-relevant objectives adequately. We argue that application-level feedback is especially important for network protocol design in sensor networks.

The motivation for meticulous sensor placement is similar -- exploiting application

layer domain knowledge while positioning sensors can provide the same information with a fraction of the nodes. Early sensor network research envisioned sensors being deployed, for the most part, in an ad-hoc fashion, requiring estimation from randomly (or uniformly) placed sensors. However, many environmental phenomena vary slowly as a function of space and samples at locations close by are often correlated. By explicitly constructing statistical models of the phenomena, researchers are able to better predict locations that will either maximize the collective entropy or, even better, reduce the entropy at the unselected positions. For some experiments, this technique has led to a 5× [KSG] reduction in node density.

Considering, in particular, the networking stack, numerous works have focused on special medium access protocols for sensor networks. The key argument made in favor of the specialist methodology is that by utilizing *a priori* knowledge of the traffic model, the energy drain due to always-on radio receivers can be effectively mitigated ([YSH], etc.). Designs using this guideline have shown tremendous improvements in network lifetime and are hence regularly used in real deployments ([BISa]Section 3.2.3). Similarly, some routing protocols [FGJ06] leverage the fact that sensing applications typically involve distributed data collection followed by centralized processing and therefore a tree topology with the fusion center at the root is adequate. In the same vein, designers of transport protocols [KFD] have utilized the fact that flows in sensor network applications can be made mutually non-interfering through cooperative scheduling, leading to a simplified design and conditionally high throughput.

However, transport protocols for sensing applications have maintained that ‘‘all bits are created equal’’ and thus data from different source nodes are handled equitably within the network. For traffic engineering in computer networks, researchers introduced the concept of Quality of Service (QoS) as a means of labeling and prioritizing data flows according to a set of static predefined policies to ensure that data from the most important source or flow gets delivered preferentially. For physiological sensing networks, however, this definition of QoS is inadequate because the notion of importance is associated not

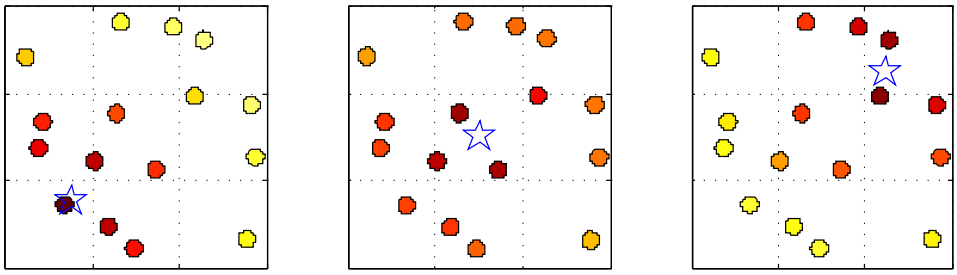

Figure 5.1: An example of foveated sensing. Sensors (dots) closer to the event (blue star) send out more data (darker shades) than ones further away.

with a source, but with the data itself and moreover, on the *value* of the data to the end user of the application. One may interpret this as saying that the importance of a source node and its resultant share of network resources should now depend on the *quality* of data currently being produced (sensed) by the node. This requires a form of dynamic QoS that translates the value of data from a node to its priority on-the-fly.

We dub this "value of sensed data" the Quality of Information (QoI) and we illustrate a mechanism of quantifying and utilizing QoI for allocating network resources at the transport protocol layer. Computing a measure of QoI for a node is hard since it is affected by multiple factors that include its location, its sensing modalities, ambient noise levels, sensing channel conditions, fault status, and physical process dynamics, all of which may vary temporally. Instead, we propose to directly use feedback from the sensor fusion algorithms in an effort to capture these effects holistically on the application itself. Before delving into a case study detailing the use of QoI (Section 5.3), we develop an intuitive appreciation of this concept.

5.1.1 Foveated Sensing

Consider the following example: we wish to track a moving target in a field. The field is instrumented with a set of acoustic sensors [GMP] for object tracking, say. To improve tracking accuracy a heuristic could be applied: once a target is detected and is being actively tracked, *nodes closer to the target should be able to send more data than ones*

further away from it. Intuitively, one would expect that nodes closer to the event have access to better quality data and funneling more data from these nodes would improve the accuracy at the fusion center. Conversely, dropping data from distant nodes under congestion would not affect the end result significantly. However, distant nodes should not be starved so that new events could be detected too. This scheme is depicted pictorially in Figure 5.1. In effect, the heuristic dynamically changes the priority of nodes' transmissions based on an estimate of the event location (one could go further to predict direction and preemptively control rates as well). Interestingly, a mechanism akin to this, termed foveated sensing, occurs in the human vision system to focus our attention on the most salient objects in our field of view [Fri06].

In this case, we could imagine that a node's QoI is higher if it closer to a target, but in general QoI would depend on the end goal of the application. Formally, we may define QoI as *an application dependent objective function that has a monotonic relationship to the accuracy of the final inference.* For example, a good QoI metric for an event detection scenario would be the probability of error (P_e), since it is considered a primary performance benchmark for this application. We posit, then, that adapting our network protocols to make P_e a first-class citizen would result in more efficient network operation for an event detection application. Practically, however, computing error probability explicitly requires ground truth. Instead, we use confidence measures provided by sensor fusion algorithms as a proxy for QoI feedback. We also show in Section 5.3.1 how P_e can be approximately computed for special cases.

Note that by this argument, neither throughput nor fairness are valid QoI metrics. Increased throughput (more data reaching the fusion center) may come at the cost of some nodes being unable to transmit, while fairness may prevent the ''right'' nodes from transmitting at higher rates. On the other hand, a QoI aware approach ensures that the best nodes communicate their information, while unimportant ones are curtailed from clogging the network. This conclusion is studied in more detail in Section 5.4.

5.1.2 Contributions and Assumptions

The contributions of this work are threefold:

- We introduce a notion of QoI that quantifies the *value* of data sensed at a node by observing feedback from the sensor fusion algorithm.
- We exemplify the use of this tool to network resource management in the context of a centralized sensor rate selection mechanism for an event detection application scenario.
- We demonstrate the application-level performance benefits accruing from such a QoI-aware approach and evaluate the costs incurred in adding feedback traffic from the sensor fusion algorithm.

We should also clarify that this chapter describes early work towards identifying objectives that better capture the intent of the transport protocol designer and is not intended as a blue print for a rate control protocol. Also, in order to form an end-to-end argument, we require to make assumptions that simplify our analysis. In particular, we assume that the wireless medium is centrally scheduled so that there is no channel contention and that the fusion center has complete route information to all nodes in the network.

Before we delve into details of our problem formulation in Section 5.3, we describe some work related to our own.

5.2 Related Work

The use of the term QoI is a relatively recent development [Bisb] but much research has been previously conducted for making protocols application-aware under the umbrella of cross-layer design [SRK03, SM05]. There have even been works specifically targeted towards sensor networks [Sic, MCL05], the main motivation behind which, is improving the lifetime of these extremely energy constrained devices. Our philosophy toward a QoI

aware approach is similar in vein but provides a more general framework that can be applied to any objective of interest, or a weighted combination of multiple objectives. The following text lists specific examples that are particularly relevant to our approach.

Gelenbe, et al. [GN] explored routing mechanisms that provide differential service to low-priority high-volume routine sensor measurements and high-priority low-volume unusual event reports by adaptively dispersing the routine traffic to secondary paths so that the event reports can be sent through faster paths with better delay characteristics. For rate control algorithms for sensor networks, Rangwala, et al. (IFRC [RGG]) developed many of the fundamental blocks required to implement a functional protocol. They established a systematic model for flow interference and evolved mechanisms to detect and circumvent it. Later, Kim, et al [KFD] and Paek, et al. [PG] presented Flush and RCRT respectively. Flush is tailored toward reliable bulk transfers and is fully distributed while RCRT transports sensor data reliably from many sources using end-to-end explicit loss recovery, placing rate adaptation functionality at the sinks and resulting in higher efficiency and flexibility. Our approach is similar to that of RCRT in that it exploits a centralized view of the network, but differs in the end objective.

Chen, et al [CFX07] describe a technique that achieves optimal *max-min* fair rate assignments using an iterative linear programming approach. We implement a similar approach to compare our QoI-aware technique to max-min fairness. Finally, Fan et al. [FZS] recently presented optimal maxmin fair schemes for energy harvesting nodes. Though, we do not yet consider nodes' energy conditions within our problem framework, we show how additional constraints could be incorporated easily.

5.3 Problem Formulation

In this section, we construct block-by-block a centralized rate control mechanism for an event detection scenario. In essence, we would like to control sensing and transmission rates for each node from a fusion center that has access to feedback from the sensor fusion

algorithm. We could paraphrase this goal as: *To optimize rate allocation with respect to a tractable Quality of Information metric for transport of sensor measurements in a multi-node multi-hop network with a centralized fusion algorithm.*

5.3.1 Error Probability as a QoI Metric

In a detection scenario, key performance metrics are the probability of false positives (false alarms) and false negatives (missed detections). Minimizing a union of these, termed error probability (P_e) would result in optimal performance. Since we know that network capacity is bounded, nodes may need to curtail the amount of information they send to the fusion center. The question is: How is P_e affected by the transmission rate from each node? This is hard to answer in general, but can be attempted with a specific scenario.

Consider a centralized multi-sensor system with the fusion center performing simple binary hypothesis testing. Each sensor k communicates N_k samples of R_s bits each to the fusion center within an epoch of time Δt at an average bit-rate $R_k = R_s N_k / \Delta t$. Denote the rate allocation vector $R = [R_1, ..., R_M]^T$. The fusion center detects the presence ($\mathcal{H}_1$) or absence ($\mathcal{H}_0$) of the event by performing a likelihood ratio test (LRT) over the received samples. We construct the hypotheses as:

$$\mathcal{H}_0 : r = n \quad \text{and} \quad \mathcal{H}_1 : r = s + n$$

where, $r = (r_1, ..., r_L)^T$ is the L-length sample vector sensed by M sensors collectively and communicated to the fusion center, $L = \sum_{k=1}^{M} N_k$, $s = (s_1, ..., s_L)^T$ is the projection of the event, to be detected in presence of additive white Gaussian noise (AWGN) $n \sim \mathcal{N}(0, \Sigma)$. The LRT reduces to the following *sufficient statistics* decision [Van68]:

$$r^T \Sigma^{-1} s \equiv l \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma \equiv \log(\eta) + \frac{1}{2}\psi^2 \tag{5.1}$$

Where, $\eta = \frac{\pi_0}{\pi_1}$, π_i are the *a priori* probabilities of the hypotheses and the term

$\psi^2 = s^T \Sigma^{-1} s$ represents the signal-to-noise ratio (SNR) for the fusion algorithm. When $l \geq \gamma$ we declare $\mathcal{H}_1$ and when $l < \gamma$ we declare $\mathcal{H}_0$.

The probability of detection, P_d and the probability of false alarm, P_f for the scenario above have the well known form given by [Van68, Kay98]:

$$\begin{aligned} P_d &= \Pr[l \geq \gamma \mid \mathcal{H}_1] = Q\left(\frac{\log(\eta)}{\psi} - \frac{\psi}{2}\right) \\ P_f &= \Pr[l \geq \gamma \mid \mathcal{H}_0] = Q\left(\frac{\psi}{2} + \frac{\log(\eta)}{\psi}\right) \end{aligned}$$

where $Q(\cdot)$ is the complementary CDF of a Gaussian random variable. The probability of error P_e is given by:

$$\begin{aligned} P_e &= \pi_0 P_f + \pi_1 (1 - P_d) \\ &= \pi_0 Q\left(\frac{\psi}{2} + \frac{\log(\eta)}{\psi}\right) + \pi_1 \left(1 - Q\left(\frac{\log(\eta)}{\psi} - \frac{\psi}{2}\right)\right) \end{aligned}$$

Rearranging, we get:

$$P_e(\psi) = \pi_0 Q\left(\frac{\psi}{2} + \frac{\log(\eta)}{\psi}\right) + \pi_1 Q\left(\frac{\psi}{2} - \frac{\log(\eta)}{\psi}\right) \tag{5.2}$$

When noise is independent across samples and sensors, Σ is a diagonal matrix of the form:

$$\Sigma = \text{diag}(\sigma_1^2 I_{N_1}, ..., \sigma_M^2 I_{N_M})$$

where σ_k^2 is the noise variance at sensor k. Then the system SNR can be written as:

$$\psi^2 = s^T \Sigma^{-1} s = \sum_{k=1}^{M} \left\{ \frac{1}{\sigma_k^2} \sum_{i=1}^{N_k} (s_i^k)^2 \right\}$$

where s_i^k is the i-th sample from the k-th sensor in s. The SNR is a function of the rates

allocated to each sensor. Thus,

$$\psi^2(R) = \sum_{k=1}^{M} \left\{ \frac{1}{\sigma_k^2} \sum_{i=1}^{R_k \Delta t / R_s} (s_i^k)^2 \right\} = \sum_{k=1}^{M} \psi_k^2(R_k) \tag{5.3}$$

$\psi_k(R_k)$ can be regarded as the sensor level SNR that contributes to the system level SNR. Our problem of rate allocation can then be formulated as:

$$\min_{R} \qquad P_e(\psi(R)) \tag{5.4}$$

To simplify our analysis, we assume equiprobable hypotheses making the second term in both $Q(\cdot)$ in Equation 5.2 disappear leaving us with:

$$P_e(\psi) = Q\left(\frac{\psi}{2}\right) \tag{5.5}$$

Also, since $Q(\cdot)$ is monotonically decreasing, the minimization can be converted to a maximization on ψ giving us the final form for our QoI objective function:

$$\max_{(R_1,\ldots,R_M)} \qquad \left[\sum_{k=1}^{M} \psi_k^2(R_k)\right]^{\frac{1}{2}} \tag{5.6}$$

This problem is solvable if a form for $\psi_k(R_k)$ is known. For this, we first need to know s_k^i. We assume that the event to be detected has a known signature $s^*(t)$ and that the measurements collected at the sensor node are of the form:

$$s_k^i = a_k s^*(t_i - \tau_k) u(t_i - \tau_k)$$

where, t_i are the time instants that the node samples its sensor, a_k and τ_k represent node specific attenuation factor and propagation delay respectively and $u(\cdot)$ is the unit step function. For many events of interest, the attenuation and propagation are a function

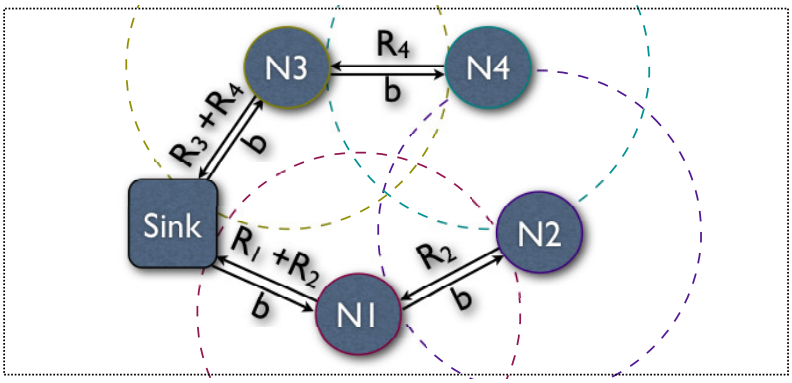

Figure 5.2: Abstracted Network Traffic Model.

of distance to the event, d_k. We assume a squared law attenuation given by: $a_k = \frac{1}{1+d_k^2}$ and a negligible propagation delay, $\tau_k \simeq 0$. When the samples are taken at equispaced intervals at a frequency f_s, the final form of s_k^i is:

$$s_k^i = \frac{1}{1+d_k^2} s^*(\frac{i}{f_s}) u(\frac{i}{f_s}) \quad i \in \{1, .., N\} \tag{5.7}$$

In binary detection scenarios, we seek to determine the instant an event occurred but the most likely location of the event is known apriori. d_k is computed using this and the node position.

Note that in this case we were able to exploit structure in the problem to convert it from one that would use confidence measures from the output of the sensor fusion algorithm to one that uses a function of the input to the algorithm. Next, we develop the problem further to include network constraints.

5.3.2 An Abstracted Network Traffic Model

Let the network consist of M sensor nodes such as in Figure 5.2. In a typical WSN context, each sensor node k senses and sends data to the sink in time epochs of size Δt. We term this *generated data traffic* and quantify it with R_k, the sampling and communication rate. Each node may also receive and forward data from other nodes along the path in a

multi-hop environment. Denote the *forwarded traffic* as F_k. Then, we can say that the *transmitted traffic* for a node is:

$$T_k = R_k + F_k$$

For example, for the 4 node network shown, $F_1 = R_2$ and thus $T_1 = R_1 + R_2$. To compute F_k for a sensor in general, we require to know which nodes transmit data to k for relaying. Using information from the routing layer, every sensing node k can assign some other node, h^k, as its next hop. Alternatively, we could define a next hop matrix H as follows:

$$H(i,j) = \begin{cases} 1 & \text{if node } i \text{ forwards to } j \\ 0 & \text{otherwise} \end{cases} \qquad \forall i,j \in \{1,...,M\} \tag{5.8}$$

Then, F_k is:

$$F_k = \sum_{i=1}^{M} T_i H(i,k)$$

And the relation for T_k can be rewritten as:

$$T_k = R_k + \sum_{i=1}^{M} T_i H(i,k) \tag{5.9}$$

Note that the circular definition in Equation 5.9 imposes a requirement that there be no loops in the routing path.

5.3.3 Interference Costs

For simplified analysis of interference, we assume a constant radius disc propagation model as shown in Figure 5.2. Then, whenever node k transmits, a set of nodes termed the interference neighbors are affected. These nodes cannot correctly decode other transmissions to them since their channel is considered occupied. We specify the interference

neighbors of k as:

$$V(i,j) = \begin{cases} 1 & \text{if } j \text{ is an interference neighbor of } i \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Using this definition, we find that a node's transmission T_k is affected only by its next hop's transmission T_{h^k} and the next hop's neighbors' transmissions, $\sum_{i=1,i\neq k}^{M} T_i V(i,h^k)$, assuming no medium access contention. This is because while a particular node is transmitting:

a. The node's next hop cannot simultaneously transmit (due to a half-duplex radio)

b. *None* of the next hop's neighbors can transmit (or they would interfere)

c. *Any* other node not included in the above can transmit

This implies, in particular, that *neighbors of k can simultaneously transmit with k, as long as these transmitting nodes are not neighbors of h^k*. As an example, note that even though $N2$ and $N4$ are interference neighbors, they can simultaneously transmit to $N1$ and $N3$ respectively. Now, if the link capacity at every node is C, this dictates that the following inequality must hold (in steady state) at every node in order to avoid congestion:

$$T_k + T_{h^k} + \sum_{i=1,i\neq k}^{M} T_i V(i,h^k) \leq C \quad \forall k \in \{1,...,M\} \tag{5.11}$$

where T_k is given by Equation 5.9 and is essentially a function of R. The collection of M inequalities (5.11) forms the first of our network transport constraints.

It should be mentioned that since all constraints have to be met simultaneously, it may lead to a conservative solution. A more aggressive strategy, that is also feasible, masks the k^{th} constraint with an indicator function 1_{R_k}, which is unity if $R_k > 0$ and zero otherwise. This has the effect of discarding the constraint where node k does not participate.

5.3.4 Feedback Traffic

A QoI aware protocol requires feedback from the fusion algorithm and thus we need to model this traffic explicitly in order to evaluate the additional cost it will incur. In Figure 5.2, this is shown with the arrows labeled b assumed to occupy a constant bit-rate of the link capacity. The feedback messages would contain the rate allocation vector computed by the fusion center based on the output of the sensor fusion algorithm. The flow of the feedback traffic is the reverse of data traffic, flowing from the sink to each of the nodes in a multi-hop fashion.

We assume that nodes unicast feedback traffic while relaying. This means that nodes with multiple children in the routing tree communicate with each of them separately. This is quite inefficient but allows us to reuse (5.11) to construct additional constraints.

We consider that a node k requires to transmit feedback traffic to a set $\mathcal{B}_k$. Since this set consists of nodes that forwarded data traffic to k in Section 5.3.2, we use the next hop matrix H from (5.8) to construct $\mathcal{B}_k$ as follows:

$$\mathcal{B}_k : \{j \mid H(j,k) = 1\}$$

As transmissions from k to each element in $\mathcal{B}_k$ are the same as that from k to h^k, we can write a set of constraints for each node k:

$$\tilde{T_k} + 1_b \tilde{T_j} + \sum_{i=1, i \neq k}^{M} 1_b \tilde{T_i} V(i,j) \leq C \quad \forall j \in \mathcal{B}_k \tag{5.12}$$

where,

$$\tilde{T_k} = T_k + |\mathcal{B}_k| \cdot b$$

$|\cdot|$ represents set cardinality in this context. Inequality (5.12) implies that a node's transmission is now dependent on transmissions from each of its routing tree children, which it could ignore earlier in (5.11). The indicator function 1_b ensures that these

additional constraints do not make the optimization conservative when $b = 0$. The constraints can be written compactly as:

$$B\tilde{R} \leq C \quad B \in \mathbb{Z}^{\tilde{M} \times M+1} \tag{5.13}$$

where, $\tilde{R} = (R^T, b)^T$ and $\tilde{M} = \sum_{k=1}^{M} |\mathcal{B}_k|$. The constraints in Inequality (5.11) need to be updated because of feedback traffic as well. They are now written as:

$$\tilde{T}_k + \tilde{T}_{h^k} + \sum_{i=1, i \neq k}^{M} \tilde{T}_i V(i, h^k) \leq C \quad \forall k \in \{1, ..., M\}$$

and can be rewritten more compactly as:

$$A\tilde{R} \leq C \quad A \in \mathbb{Z}^{M \times M+1} \tag{5.14}$$

$$R_k \geq 0 \quad \forall k \in \{1, ..., M\} \tag{5.15}$$

The last constraint set ensures non-negative rate allocations.

5.3.5 Rate Control Policies

A key question that needs to be answered for optimal QoI aware rate allocation is about how the rates would be achieved. One may view the digitization process to consist of three distinct steps: sampling, quantization and encoding as shown in Figure 5.3. The sampling block generates N samples in every time epoch and the quantizer converts the samples to a bit-vector of size NR_s. It is the encoding block that converts this bit-vector to an $R_k \leq NR_s$ sized stream.

There are a multitude of ways in which the encoder can perform this compression, but they can all be classified as either lossless or lossy. Lossless compression may not be able to achieve an arbitrary target rate since it is lower bounded by the entropy of the data. Thus, we will focus on lossy techniques here. Also, since our problem is one

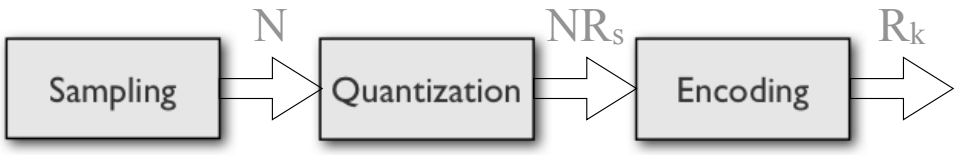

Figure 5.3: The three step digitization process. The encoding block manages the output bit-rate from the system.

of cooperatively detecting an event, the information sent by one sensor may affect the compressibility of data at another. In general, our rate management problem falls under the umbrella of distributed source coding techniques, the theory for which is outside the scope of the current work. It may be worthwhile, however, to mention a few notable works in the regime of sensor networks.

Kansal, et al. [KRS] approached the problem of optimizing network lifetime based on a distortion measure of the final inference. They formulated the encoder block as a Quadratic Gaussian CEO problem [VB97, Ooh98] which was solved using Slepian-Wolf (SW) coding [PTR] providing a feasible sum rate region that would contain the distortion. Li, et al. [LD] converted the problem to a convex one, lending to implementation of the optimization. However, the SW coding bound has only been recently approached [PR03] and to the authors' knowledge, there has been no practical implementation for sensor networks to date.

In fact, Marco, et al. [MDL] proved that even with SW coding, increasing node density does not help reduce the overall bit rate. This was clarified by Kashyap, et al. [KLX] who showed that there does exist a density independent bit rate for the reconstruction of a Gaussian random field. They also demonstrated a sample selection scheme that comes reasonably close.

From our study of past work on algorithms for constrained systems, we discovered a common underlying principle -- *simplex sigillum veri* (simplicity is the seal of truth [Pol73]). Considering the computational unfeasibility of most other encoding techniques, we decided to use pure sample selection ourselves. The problem can be described as

follows: When $R_k < NR_s$, select a subset $\tilde{\mathcal{S}}_k$ of the collected sample set $\mathcal{S}_k$ that minimizes (system-level) P_e. That is:

$$\tilde{\mathcal{S}}_k \subseteq \mathcal{S}_k \quad \left|\tilde{\mathcal{S}}_k\right| = N_k = \frac{R_k}{R_s} \quad \mathcal{S}_k = \{s_k^i\}_{i=1}^N$$

This selection process could be considered homologous to the one for choosing sensor locations, except on the temporal axis. To borrow techniques, however, would require to explicitly construct accurate time domain models of the phenomena, a non-trivial task at best [KSG]. Instead, we consider a set of 'blind' selection (conversely, dropping) policies -- downsampling, token bucket, and random sampling. We compare these with the optimal selection policy.

The downsampling procedure performs low pass filtering and then resamples the sequence at a lower rate $f_s N_k/N$. The token bucket algorithm achieves an average output rate by queueing bursts of traffic at the input. In our case, the input is a constant bit rate stream, so the token bucket will have to drop samples to ensure a finite queue size. Also, we would not like to queue samples across epochs. In effect, the token bucket algorithm will drop $N - N_k$ samples in more or less an interleaved manner. The random sampling technique tests the value of a binary random variable $x \in \{0, 1\}$ $\mu = N_k/N$ at every sampling index, i. If $x(i) = 1$, the sample is selected, otherwise it is dropped.

Though each of these schemes strives to achieve the same average output rate, it is interesting to observe which samples get picked. Figure 5.4 shows this effect. Note that the downsampling scheme results in samples that may not be present in the original vector and that random sampling does not guarantee fewer than N_k samples in every run. Also, random sampling will pick a different set of samples each time. Figure 5.4 only depicts one instance.

Returning to our problem, according to Objective 5.6, we require to maximize the system-level SNR ψ. We would thus like to determine the effect of each rate control policy on the SNR. To do this effectively, we require to fix the event signature $s^*(t)$ and

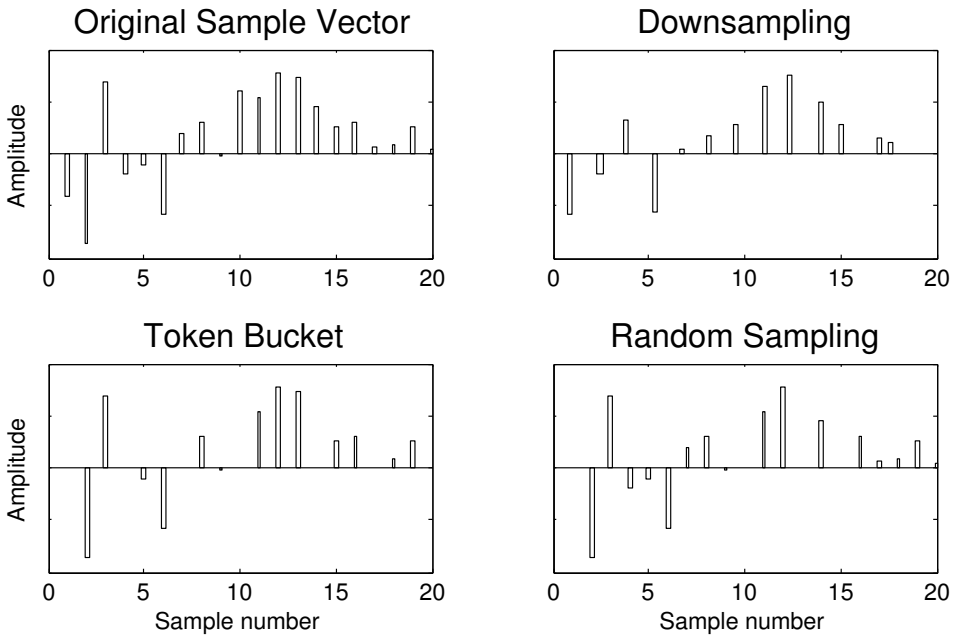

Figure 5.4: Effect of policy on sample selection when target rate is 0.7.

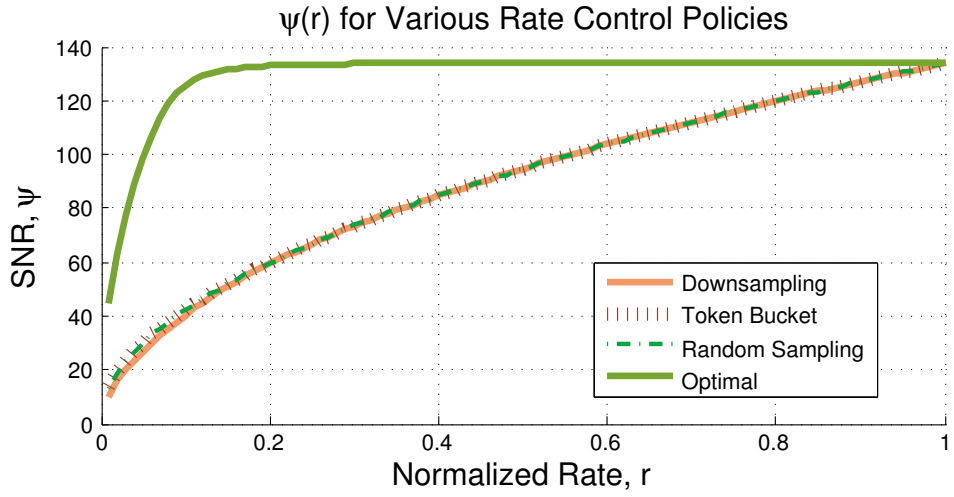

Figure 5.5: Effect of rate control (sample selection) policies on $\psi(r)$.

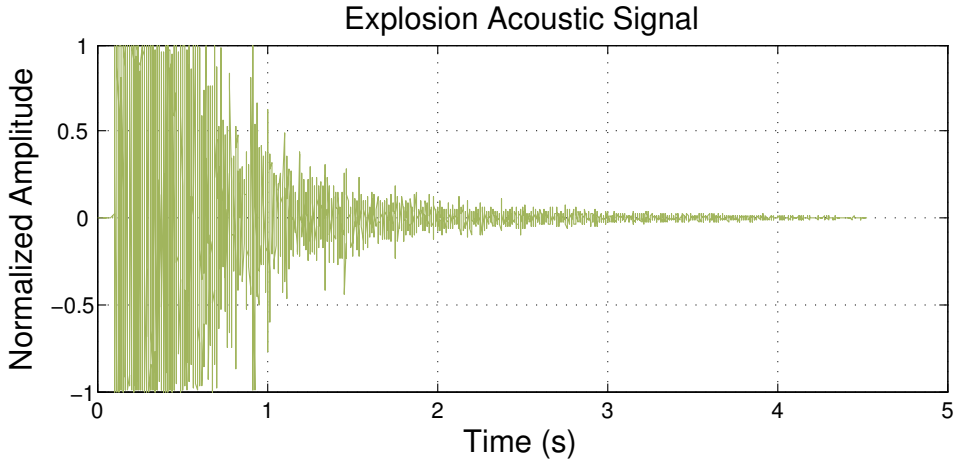

Figure 5.6: Signature of the (explosion) event to be detected *[Courtesy www.free-loops.com]*.

we chose the acoustic sample shown in Figure 5.6 for our application. We computed the node-level SNR, ψ_k for each scheme for a range of target rates. This is equivalent to computing ψ for the trivial case of $M = 1$. The results are shown in Figure 5.5. We also computed the SNR for the optimal case of picking the highest power samples.

Surprisingly, the SNR *for this signal* is quite unaffected by which of the three schemes we pick (downsampling is slightly worse at some rates). The optimal scheme is not blind and we can sadly not pick it. Further, both token bucket and random sampling require to communicate the sample indices along with the samples, increasing the overall rate. Additionally, since they drop samples without low pass filtering, aliasing may occur.

5.3.6 Sensing Model

An additional step remains before we can solve for the QoI objective (5.6) -- a $\psi_k(r)$ function that can be evaluated within the optimization framework. For our example, we used an acoustic event, and in particular an explosion sound sampled at 44.1 kHz. To compute $\psi_k(r)$ for each node k and each feasible sampling rate r, we used a distance based attenuation model with $3dB$ loss per unit distance. Rates were normalized to $[0, 1]$ and arbitrary rates were achieved by a downsampling procedure. The plot of $\psi_k(r)$ for an example node is shown in Figure 5.7. Using empirical assessment, we found that the

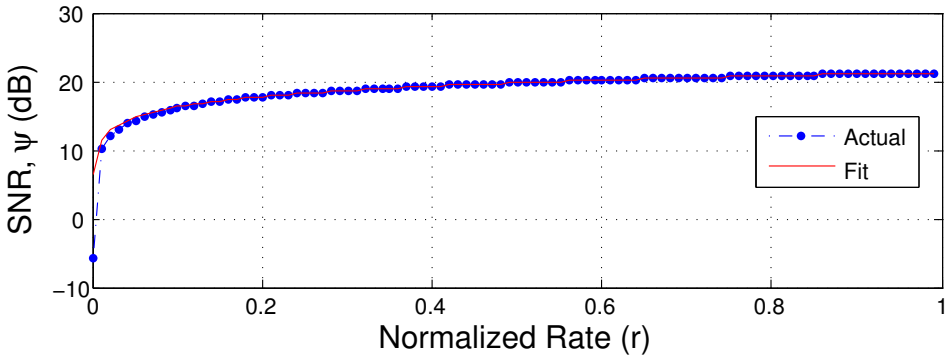

Figure 5.7: Fitting the $\psi(r)$ function using least squares regression.

following function fit the data adequately*:*

$$\psi_k(r) = \alpha_k r^{1/2} \tag{5.16}$$

where, α_k is a node-specific parameter identified through training. This form is especially fortuitous since consequently, $\psi_k^2(r)$ is linear in r and the solution can be found through convex optimization. We must emphasize that this may not hold in general and we are considering the use of more complex sensing models that include the effects of reflections and occlusions. Anyhow, using (5.16) with (5.6) and the constraints (5.13, 5.14 and 5.15) derived form the network topology, we are now in a position to compute a rate vector R that maximizes the QoI delivered to the fusion center.

5.3.7 Optimizing Quality of Information

We end this section with a formal statement of the problem:

$$\max_{(R_1,...,R_k)} \quad \left[\sum_{k=1}^{M} \alpha_k^2 R_k\right]^{\frac{1}{2}} \tag{5.17}$$

$$A\tilde{R} \leq C \quad A \in \mathbb{Z}^{M \times M+1} \tag{5.18}$$

$$B\tilde{R} \leq C \quad B \in \mathbb{Z}^{\tilde{M} \times M+1} \tag{5.19}$$

$$R_k \geq 0 \quad \forall k \in \{1, ..., M\} \tag{5.20}$$

5.3.8 Controlling Rates using Network Feedback

A key requirement for executing the optimization problem above is knowledge of interference constraints (5.14). This is reasonable if the network topology is known in advance or if state variable information from the routing and MAC protocols is available, but not in general. Instead, we could exploit the fact that the constraints essentially embody the network's (in)ability to support an arbitrary set of rates. It may thus be possible to reconstruct A by probing the network with an assignment rate vector (say, R) and perceiving what the network is able to deliver, say $\hat{R}$. In fact, we believe this mechanism could be applied to perform QoI maximization directly, without first estimating A.

Before we describe the procedure, we state a network congestion assumption. We assume that whenever a rate cannot be supported on a link, participating nodes negotiate *locally* to a proportionally reduced data transmission rate that is feasible on that link, given transmissions from interfering nodes. In this way, a bottleneck link will drop packets so that each flow gets a proportional (rather than equal) share of what it transmitted. This behavior occurs naturally when per-packet hop-by-hop ACKs are employed, which is common in many wireless link layer protocols.

Our proposed rate control mechanism uses a greedy algorithm to select rates based on each node's contribution to the QoI. This procedure is listed in Algorithm 5.1. The nodes are first sorted in descending order based on their α_k parameters from (5.16). The top node is then temporarily assigned a rate equal to the spare capacity at the fusion

Algorithm 5.1 Greedy Rate Control

1. Initialize rates: $R_k \leftarrow 0 \quad \forall k \in \{1, ..., M\}$
2. Sort nodes in order of QoI contribution
3. For each node j in the sorted list:
   a. Save rate vector: $R = [R_1, ..., R_M]^T$
   b. Assign spare capacity: $R_j \leftarrow C - \sum R_k$
   c. Allow network to converge to a rate $\hat{R}$
   d. Compare QoI: if $\psi(\hat{R}) > \psi(R)$:
      i. Update rate vector: $R \leftarrow \hat{R}$
4. Output final rate: return R

center. With this assignment rate vector, R, the network is allowed to transport some data and stabilize to some feasible delivery rate vector, $\hat{R}$ ($\leq R$). If the QoI delivered with $\hat{R}$ exceeds that of the previous iteration, the assignment rate vector is updated, in effect fixing the rate for this node. This probe and sense routine is then repeated for the rest of the nodes.

Some notes about this procedure are in order. First, observe that rates are assigned with an all-or-nothing policy -- if there is spare capacity at the fusion center, a node is given everything. If that does not improve the overall QoI (because it interfered with another node), the rate is taken away and the node is ignored in future iterations. This simple strategy works because (a) QoI contributions according to (5.16) are monotonic so that if the maximum rate does not deliver higher QoI, no other combination will do so either and (b) since nodes are tested in order of their potential QoI contribution, 'better' nodes are guaranteed higher rates. Consequently, a node with a lower α_k will be assigned a non-zero rate only when a better node could not utilize capacity completely due to high (multi-hop) interference costs. It should also be mentioned that this strategy fails when two nodes have slight differences in α_k but large disparity in interference costs. The effect of this failure is minimal and is evaluated in Section 5.4.

The second aspect to note is that the rate vector is updated based on what the network could deliver, rather than on what was assigned. This allows us to select subsequent rates based on spare capacity at the fusion center, since that is a known link constraint.

However, this causes problems when packets are dropped ($\hat{R} < R$) due to wireless link errors rather than congestion. In congestion, any rate above a threshold will be capped to that value so assigning a higher rate in future iterations is futile and in fact, tends to increase network stabilization time. Instead, if packet loss was due to link errors, assigning the returned rate in future iterations would result in exponentially decaying delivery rates (assuming link errors cause a constant multiplicative loss).

Third, we are able to sort nodes for QoI contribution because the form of (5.16) assures that a node with a higher α_k will be a larger contributor independent of rate. This does not hold in general, even if the $\psi_k(r)$ are assumed to be convex. We are in the process of understanding what properties of QoI functions enable this feature. Fourth, network topology should remain stable during the entire procedure since any change in interference constraints may lead to erroneous results. And finally, we believe this algorithm can be also used with quasi-static networks by monitoring the delivery rates even after the procedure completes. If the fusion center ever finds $\hat{R} \neq R$, that signals a change in network topology and the entire algorithm can be re-run.

5.4 Simulation Results

In this section, we describe the performance of our QoI aware objective (5.6) and compare it to the traditional objectives of fairness and throughput. We use the *max-min* fairness criterion for the first (i.e. max min R) and maximize the data delivered to the fusion center for the second (i.e. $\max \sum R_k$). The constraints are the same for all three versions of the problem. The next section provides simulation results on a simple configuration before moving to a larger network example.

5.4.1 A 2-node 2-hop Network

Consider a linear network of two nodes and one fusion center or sink. The sink is located at $x = 0$ and nodes $N1$ and $N2$ are located at $x = 5$ and $x = 10$ respectively. For this

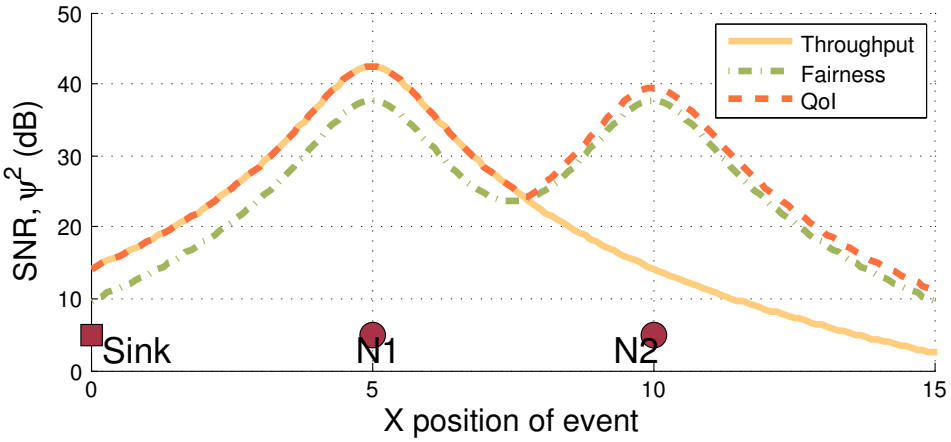

Figure 5.8: Comparing SNRs of received signals for differing objectives.

system, the network constraint matrices are computed using (5.14 and 5.15) are:

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad N = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad R = \begin{bmatrix} S \\ N_1 \\ N_2 \end{bmatrix}$$

We search for the rate vector that optimizes each objective function (fairness, throughput, QoI). We then compute the system level SNR, ψ based on the allocated rates using (5.3). We repeat the experiment for event locations at each point along the x-axis in $[0, 15]$. This provides a measure of performance for events anywhere along the axis. Note that the sink does not participate in the sensing process in our experiment. Figure 5.8 shows the SNR values for the different objectives along the line joining the sink, N_1 and N_2. The noise variance σ_k^2 is assumed to be unity for both nodes.

We observe that ψ for the fairness case is lower. The reason for this is apparent from Figure 5.9. The fairness approach selects rates for both N_1 and N_2 to be $\frac{1}{3}$ of link capacity. This is correct since traffic from N_2 occupies the channel twice (due to forwarding). This allocation is also the most fair one. Since allocating any rate to N_2 costs twice as much channel capacity, the throughput optimizing approach omits it altogether. Thus, when

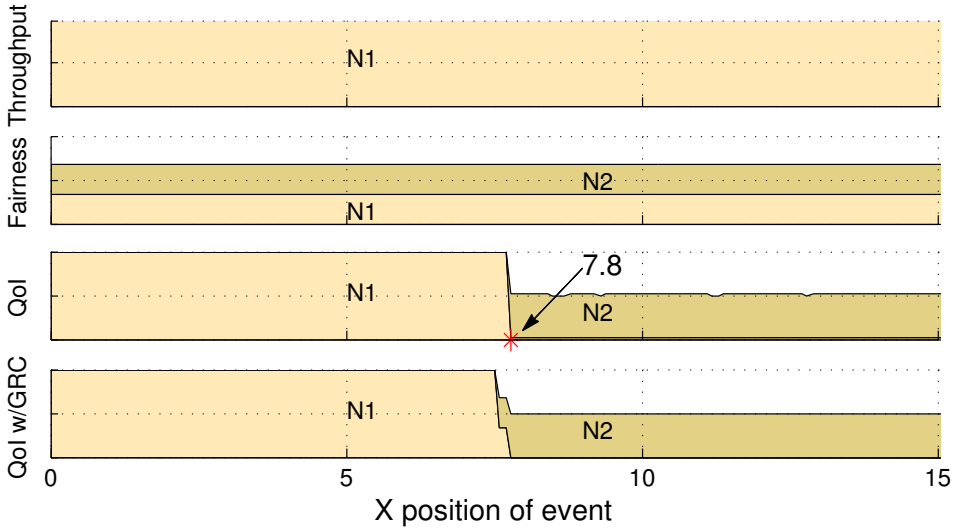

Figure 5.9: Comparing rate allocations at the 2 nodes for differing objectives.

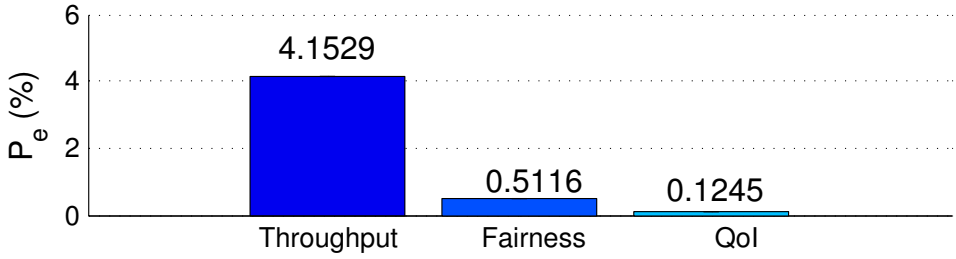

Figure 5.10: Comparing mean probability of error for differing objectives.

the event is nearer N_1 the throughput case performs well (high SNR), but near N_2 SNR drops substantially.

The QoI aware approach behaves quite interestingly -- it allots rates to the node nearer to the event, but not precisely so. One would think that the crossover point would be midway at $x = 7.5$. However, it is actually at $x = 7.8$. The reason for this stems from the asymmetry in the cost of $N2$. Another artifact worth noting is that crossover occurs instantaneously rather than morphing smoothly from one to the other. This is due to the same reasons that the greedy approach described in Section 5.3.8 works. Also shown is the rate vector computed using Greedy Rate Control (GRC) and we find a discrepancy near the crossover point, as expected but the corresponding SNR loss is below 5%. In any case, we see that the SNR from the QoI objective is consistently high and is *never below either fairness or throughput*.

We can also compute the P_e for the SNRs from (5.2). Figure 5.10 reports the mean of the error probability over the axis $[0, 15]$. We see that the *QoI aware rate control results in a* 75% *reduction in mean error probability over fairness*. But QoI comes at the cost of feedback, so its interesting to see how far this keeps up.

5.4.1.1 Effect of Feedback Traffic

Feedback traffic parameter, b represents the fraction of link capacity that a feedback message occupies in each epoch. For example, if $\Delta t = 100$ ms, link capacity $C = 250$ kbps (for 802.15.4) and a feedback message is $m_f = 32$ bytes, $b = \frac{m_f * 8}{C\Delta t} = \simeq 1\%$. In general, m_f depends on the size of the network and Δt on event dynamics (see Section 5.5). Thus, in our evaluation, we examine performance over different b values.

Using the construction in Section 5.3.4, we can develop constraints that include b. In this 2-node system, the most restrictive one is: $R_1 + 2R_2 + 2b \leq C$. Applying this additional constraint to our previous problem with various values of b results in Figure 5.11. For the reasonable case of $b = 1\%$ there is no appreciable loss in SNR and even

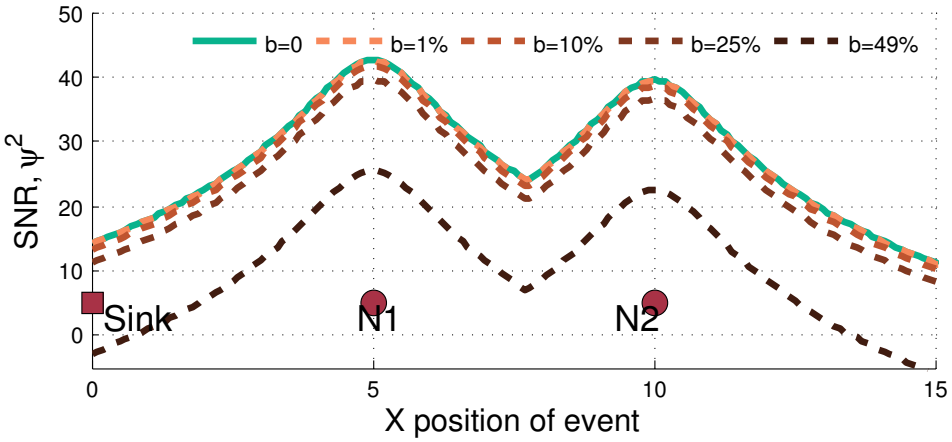

Figure 5.11: Comparing SNR when feedback traffic occupies some percentage of link capacity.

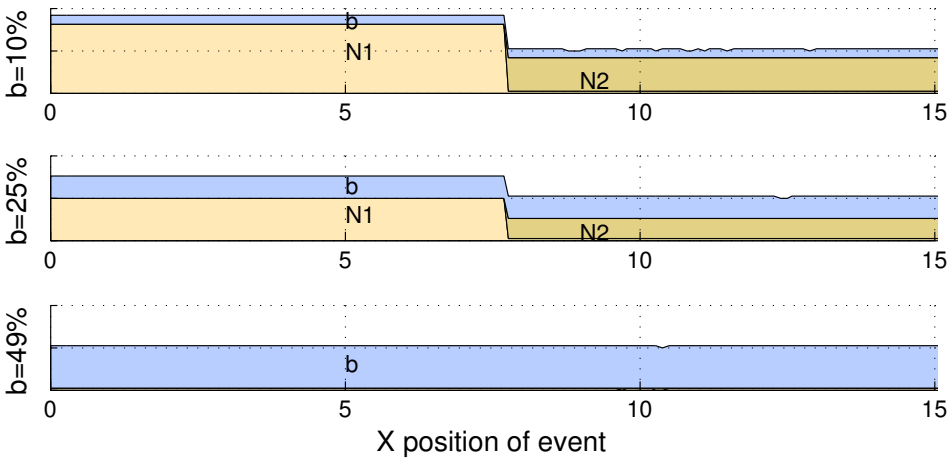

Figure 5.12: Comparing rate allocations at the 2 nodes for differing feedback traffic.

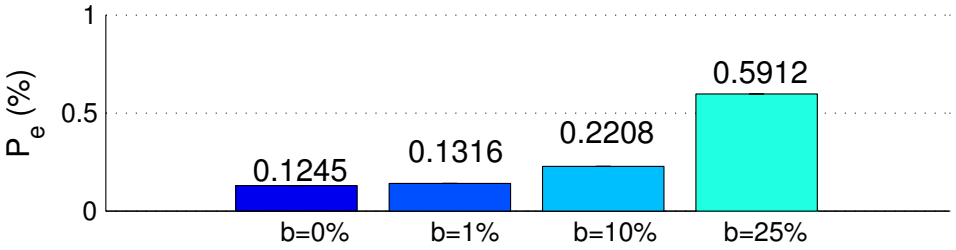

Figure 5.13: Comparing mean probability of error for differing feedback traffic.

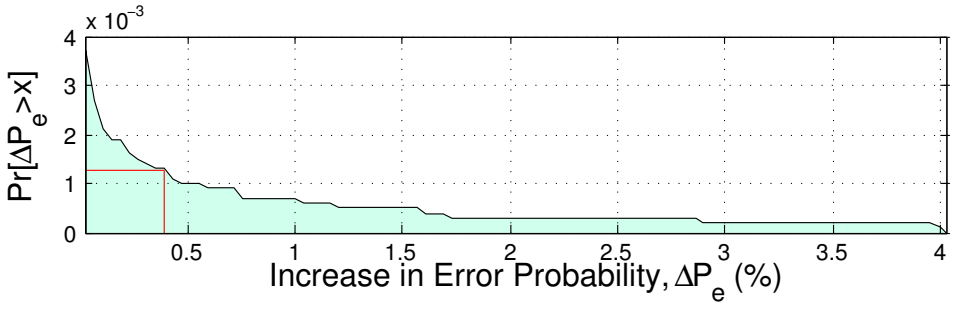

Figure 5.14: Increase in P_e due a positional inaccuracy that is $\mathcal{N}(0,1)$ distributed about a point $x = p$, where p is uniformly distributed over $[0, 15]$.

with $b = 10\%$, SNR is better than fairness. Rate allocations for different b are illustrated in Figure 5.12. The case for $b = 49\%$ represents an extreme case since the channel is swamped with feedback alone -- b is also transmitted twice. This result demonstrates (empirically) the robustness of the objective even under stress. The mean of P_e is reported in Figure 5.13. For this simple system, the QoI approach remains unsurpassed even at $b = 25\%$.

5.4.1.2 Effect of Location Inaccuracy

A question worth pondering is: since the QoI requires knowledge of the event location, what if those estimates were inaccurate? This question is especially relevant in applications where detection and localization are being handled concurrently. We examine this by deliberately injecting $\mathcal{N}(0,1)$ noise into the location estimate. Noting that at $x = 7.8$ there is a sharp rate transition, this is a considerable amount of noise. To ensure statistical significance, we run a 10000 run Monte Carlo simulation over the entire axis with uniform probability. The result is shown in Figure 5.14.

For this plot, the x-axis denotes the increase in P_e due to positional inaccuracy and the y-axis denotes the probability with which this increase may occur. Since the P_e margin between fairness and QoI is about 0.38% , we conclude that QoI will be worse than fairness with this noise no more than 0.12% of the time.

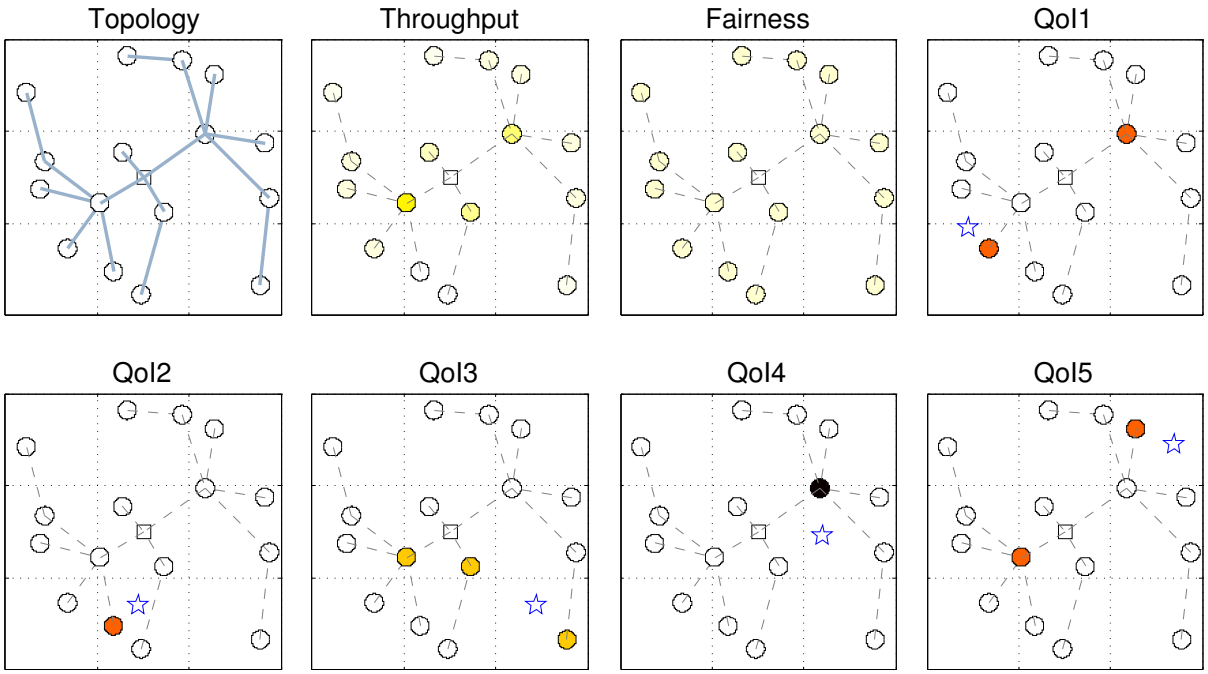

Figure 5.15: Network layout and rate allocations for throughput, fairness and QoI for 5 event (star) locations.

5.4.2 A 16-node 3-hop Network

We now discuss the performance of the QoI aware system with a 16-node network on a larger 60×60 grid shown left-most in Figure 5.15 with the sink at the center and a radio range of 20 units. We set $b = 0$ here and compute the SNR and P_e for the three objectives as before. The SNR across the field is plotted by sorting the values over the grid (Figure 5.16) and the mean P_e is shown in Figure 5.17.

The figures demonstrate that QoI is a better objective especially as networks scale, because QoI can judiciously utilize the the bottleneck link capacity. This is seen explicitly in the rate allocations in Figure 5.15 (darker shades represent higher rates). Relay traffic is not included but can be inferred easily. While the throughput and fairness objectives allocate rates that are either too aggressive or too conservative, the QoI objective ensures that nodes that contribute significantly to the end inference are preferred. Interestingly, in some cases (QoI1, QoI5), nodes further away from the event are allocated rates as well.

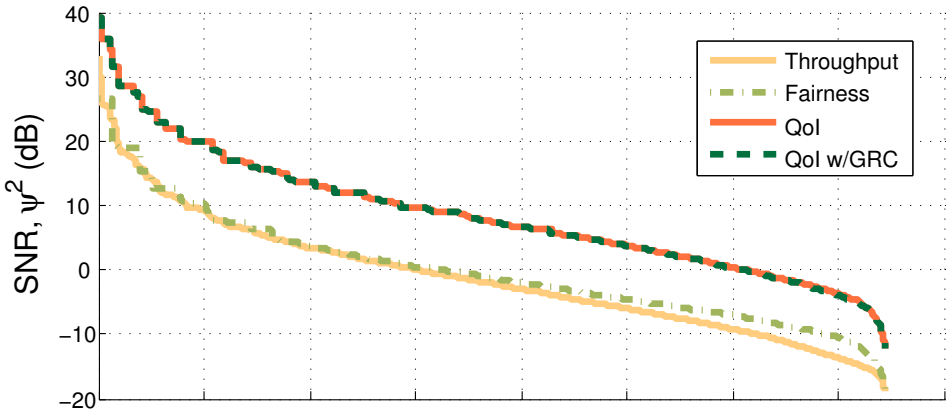

Figure 5.16: Comparing SNR sorted over entire grid.

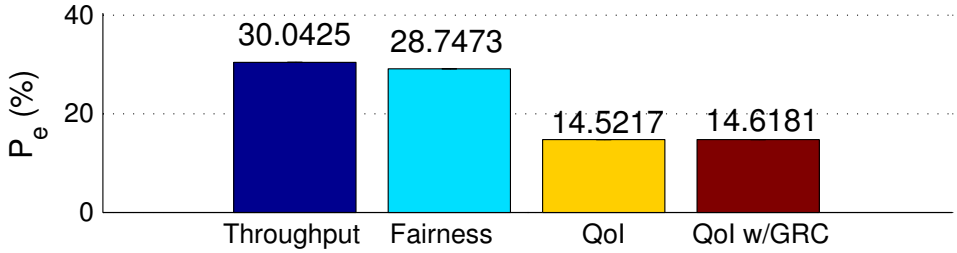

Figure 5.17: Comparing mean probability of error across entire field.

This is an artifact of interference at the node closest to the event and a resultant shift in the bottleneck link away from the sink. For both throughput and fairness, the bottleneck is at the sink. In the case for QoI4, the closest node is one hop away from the sink and can use the link capacity completely.

Also included in the figures are results from using the GRC procedure. While the loss in SNR is hardly visible, the overall P_e rises by less than 0.1%. Note that mean P_e for all cases is higher in this example (vs. Section 5.4.1) because the field is larger and nodes are more spread out.

5.5 Practical Considerations

We side stepped a plethora of significant practical considerations in order to analyze the system fully. Here, we suggest ways in which some of them could be handled. We ground

the discussion in the context of a K-site event detection application.

5.5.1 Network Dynamics

In binary detection, the event is either present or absent and the rate allocation remains static in steady state. In the K-site version, at most one event may occur but its site is unknown. Thus, rates may be continually and dynamically altered so as to minimize P_e (eg. foveal sensing). Until now, we did not consider epoch interval Δt as a variable of interest. In the K-site case, however, it affects network dynamics and P_e tremendously. In a multi-hop network, it can take up to $2D\Delta t$ time for feedback from the fusion center to take effect, where D is the network diameter. Therefore, to improve network response time, Δt must remain as short as possible. However, Δt must also be long enough to accommodate transmissions from nodes that share the medium. In particular, messages contain a constant overhead that is significant when Δt is very short. Additionally, a setting for Δt must also consider the minimum interval between two events. The network must be allowed enough time to recover from rate transients.

Further, since feedback is also sent multi-hop, updated rate vectors come into effect in a rippled manner. This means that our meticulously derived constraints could be violated at some nodes (and may be too conservative at others). To avoid this, the control algorithm has to 'plan ahead' to ensure that rate vector transients do not cause havoc. This can be achieved only because the algorithm can compute the congestion effect of prior allocations in future epochs. Though past allocations cannot be revoked, new ones can accommodate for the rippling effect.

5.5.2 Practical Network Models

Our abstracted network model assumed a well-behaved link model. Effective link capacity, however, shows high variability and predicting it accurately at the fusion center is impossible. One way to address this is to split each rate into an ''always fill'' and ''fill

if possible'' part. Metrics such as ETX could provide link quality statistics to compute these fractions and nodes could exploit current local knowledge to fill each one in. Additionally, some devices use radios with multiple link rates. This exacerbates the problem since link rate settings are typically based on current channel conditions. Resolving this intertwining of layers is precisely what we believe distributed QoI aware algorithms can achieve.

A critical drawback of the proposed greedy rate control algorithm is its inability to handle wireless link errors. We believe this could be handled by the fusion center performing the probe and sense routine twice for every node to estimate the fraction of rate loss due to congestion and link errors independently. We must mention that our current problem formulation does not consider link layer ACKs, multiple path routing or multiple sink nodes. It is conceivable, though, that each of these can be incorporated within the same framework.

5.6 Conclusion

This chapter strives to illustrate that meticulous attention to application relevant objectives can lead to higher networking performance and reduced communications cost by optimizing sampling rates. While many researchers have used error probability as an optimization criterion for event detection applications, we believe this is the first time it is being coupled with the effects of wireless interference and multi-hop forwarding. We attempted to translate heuristics that lead to mechanisms such as foveated sensing into formal objectives using a QoI metric. This not only allowed us to analyze and incline the problem more rigorously, but in some instances, uncovered reasons why particular heuristics work exceptionally. For example, we see from our problem formulation why simple distance weighted rate allocation might work. However, we now possess tools to tweak a multitude of variables simultaneously -- noise variance, sensor reliability, channel conditions, etc. We also showed a practical greedy rate control mechanism that achieves

close to optimal performance.

Our results demonstrate the benefit of using prior information of event location on the probability of error. On an example network, using the QoI objective reduced the P_e by $3\times$ while incurring marginal cost from explicit feedback. Moreover, the effects of inaccuracy in the estimate of event location are examined and found to be contained with high probability. We show interesting results for a larger network that illustrate why QoI is especially important as networks scale. In particular, careful rate selection shifts the bottleneck link away from the sink, allowing the ''best'' nodes to participate more effectively. A fortunate side effect of this is that it relieves nodes closer to the sink, improving mean network lifetime.

In conclusion, the philosophy behind our approach is similar to recent efforts in Content Centric Networking [CH] that endow the networking stack with knowledge of the intent of the communication transaction. The difference is that a QoI based protocol is not only content-aware, but is also cognizant of the *effect* the content has on the application.

CHAPTER 6

Compressive Oversampling for Robust Data Transmission

6.1 Introduction

In this chapter, we focus our attention on improving transport robustness through an innovative sampling procedure. Data loss in wireless sensing applications is inevitable, whether due to exogenous (such as transmission medium impediments) or endogenous (such as faulty sensors) causes. While many schemes have been proposed to cope with this issue, the emerging area of compressive sensing enables a fresh perspective for sensor networks. Many physical phenomena are compressible in a known domain and it is beneficial to use some form of source coding or compression, whenever practical, to reduce redundancy in the data prior to transmission. For example, sounds are compactly represented in the frequency domain whereas images may be compressed in the wavelet domain. Traditionally, compression is performed at the application layer after the signal is sampled and digitized and typically imposes a high computation overhead at the encoder. This cost is the major reason that low-power embedded sensing systems have to make a judicious choice about when to employ source coding [NTG]. Advances in compressive sensing (CS) [CRT06b], however, have made it possible to shift this computation burden to the decoder, presumably a more capable data sink (e.g., a wireless sensor network's

base station), which is neither power nor memory bound. CS enables source compression to be performed inexpensively at the encoder, with a slight sampling overhead[1] and with little or no knowledge of the compression domain.

Compression, however, also makes each transmitted bit of information more precious, necessitating a reliable transport mechanism to maintain the quality of information. To cope with channel disturbances, retransmission schemes have popularly been applied, but they are inefficient in many scenarios, such as on acoustic links used for underwater communication [APM05], where round trip delays and ARQ traffic cost precious throughput. Retransmissions are ineffective in other cases too, for example, in multicast transmissions or when transmission latency is paramount for rapid detection. Forward error correction schemes like Reed-Solomon [RS60], LT [Lub02] or convolutional codes are better suited for these scenarios, but their use in low-power sensing has been limited, primarily because of their computational complexity or bandwidth overhead [RV06].

Fortunately, the computational benefits of CS coupled with its inherent use of randomness can make it an attractive choice for combating erasures as well. A key observation that makes this possible is that reconstruction algorithms for compressively sampled data exploit randomness within the measurement process. Therefore, the stochastic nature of wireless link losses and short-term sensor malfunctions do not hamper the performance of reconstruction algorithms at the decoder. In fact, to the decoder, losses are indistinguishable from an a priori lower sensing rate. We, therefore, propose using compressive oversampling as a low encoding-cost, proactive erasure coding strategy and show, in particular, that employing CS erasure coding (CSEC) has three desirable features:

- CSEC is achieved by nominal oversampling in an incoherent measurement basis. Compared to the cost of conventional erasure coding that is applied over the entire data set from scratch, additional sampling can be much cheaper, especially if random sampling is used. The high cost of CS decoding is amortized over joint source

[1]CS sampling incurs a logarithmic overhead when compared to acquiring the signal directly in the compression domain.

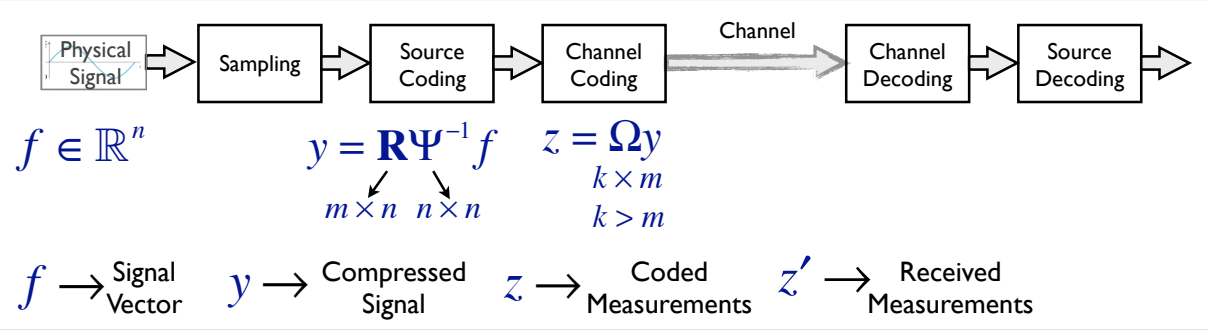

Figure 6.1: The conventional sequence of source and channel coding.

and channel coding and is "free", if CS was already being employed for source decompression.

- The performance of CS erasure coding with random sampling is similar to conventional schemes such as Reed-Solomon and, in general, the BCH family of codes, in that it can recover as many missing symbols for the same relative redundancy in a memoryless erasure channel. This aspect is covered in Section 6.2.4.

- CS erasure coding is robust to estimation error in channel loss probability. For example, if a BCH code of block size n were designed to correct up to t erasures, in a situation where $e > t$ erasures occur, the entire block of n symbols would be discarded. This implies that BCH codes must consider and be designed for a worst-case loss probability for recovery to succeed. An equivalent CS strategy, however, guarantees that even if $e > t$ symbols are lost, the best approximation of the signal is reconstructed from the $n - e$ remaining symbols. This means that even if channel coding fails at the physical layer, CSEC can recover the signal at the application layer.

Despite its advantages, CS erasure coding is not intended as a replacement for traditional physical layer channel codes. It is neither as general-purpose (i.e. it cannot be used for arbitrary non-sparse data), nor is the decoding as computationally efficient (yet). Instead, CSEC should be considered as a coding strategy that is applied at the application layer, where it utilizes knowledge of signal characteristics for better performance. In this regard,

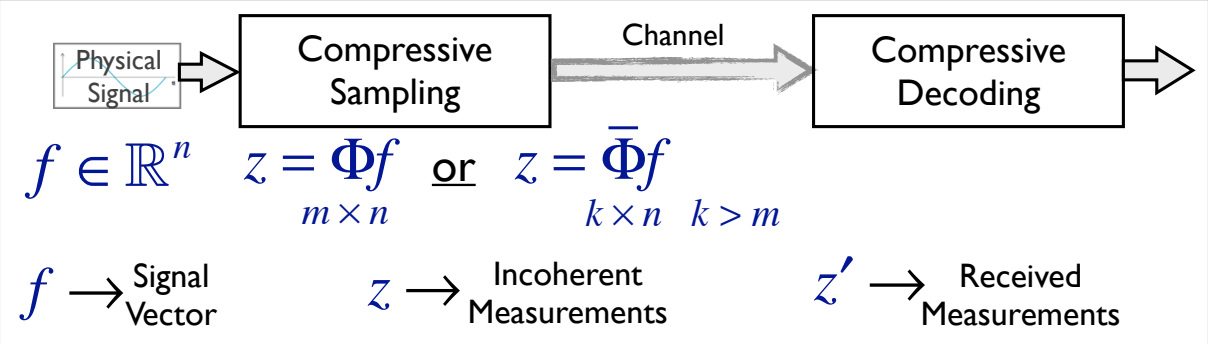

Figure 6.2: Proposed joint source-channel coding using compressive sensing.

it is the reduced encoding cost that makes CSEC especially attractive for low-power physiological sensing. We quantify its energy efficiency benefits in Section 6.3.3.

We highlight the conventional and proposed approaches in Figures 6.1 and 6.2respectively. Notation used in the figures is introduced in the Section 6.2. Typically, source coding is performed after the signal is completely acquired, removing redundancy in the samples through a lossy or lossless compression routine. This step is performed at the application layer and utilizes known signal models to determine the most succinct representation domain for the phenomenon of interest. The compressed data is then handed to the communication protocol stack, where just before transmission, usually at the physical layer, the data may be encoded again to introduce a controlled amount of redundancy. If some transmitted symbols are received in error or not at all, the decoder may be able to recover the original data using this extra information.

If, on the other hand, compressive sampling were to be employed for joint source and channel coding, the sampling stage would itself subsume all the coding blocks. The CS sampling block uses one of a variety of random measurement techniques that ensure that sufficient unique information is captured by the sampling process with high probability. We propose that the CS sampling block should be designed not merely to include prior knowledge of signal characteristics in terms of its sparsity in a specific domain, but consider channel characteristics as well. In particular, we propose tuning the sampling process, e.g., through judicious oversampling, to improve the robustness to channel im-

pairments. We show in Section 6.2.4 that the universality of compressive sensing to the sparsity domain extends to the channel model as well, making CSEC advantageous even when channel characteristics are not precisely known. In particular, we will see in Section 6.3.2 that signal reconstruction performance with CSEC degrades gracefully when the average sampling rate at the acquisition stage is insufficient for exact recovery.

6.2 Compressive Sensing for Erasure Coding

The problem we seek to address is acquiring an n-length vector $f \in \mathbb{R}^n$ at a sensor node and communicating a k-length measurement vector $z \in \mathbb{R}^k$ such that can be recovered accurately at a base station one or more wireless hops away. We assume a generic wireless sensing application, where the signals are sparse or compressible in a known domain, and the data is collected centrally at a capable base-station. To construct our argument, we first briefly discuss both channel coding and compressive sensing. We then propose a compressive coding strategy in which oversampling suffices for robust data transmission.

6.2.1 Channel Coding Overview

If we consider a simple sense-and-send scenario where we send the sensed signal $f \in \mathbb{R}^n$ to a base station over an unreliable communication channel, this is equivalent to setting $z = f$ and $k = n$. However, if a channel coding function $\mathbb{F}_c$ is applied prior to transmission, $z = \mathbb{F}_c(f)$ and since channel coding increases the average transmission rate by adding redundancy, $k \geq n$. Consider a linear channel coding function $z = \Omega f$, where $\Omega \in \mathbb{R}^{k \times n}$ is the equivalent channel coding matrix. When z is transmitted through a lossy channel, some measurements may not be received at the other end. We define the received measurement vector z' of length $k' = k - e$, where e is the number of erasures. The channel may also be modeled as a linear operator $C \in \mathbb{R}^{k' \times k}$ so that $z' = Cz$. In general, C can consist of any values, but for the class of binary erasure channels we consider here, C is a sub-matrix of an identity matrix $\mathbb{I}_k$, where $e = k - k'$ rows have been

omitted denoting the samples that were dropped. Recovering the original signal from the received data is then a decoding operation of the form:

$$\hat{f} = (C\Omega)^{+} z' \tag{6.1}$$

where $X^{+} = \left(X^{T}X\right)^{-1} X^{T}$, is the Moore-Penrose pseudo-inverse. If $C\Omega$ is full rank, the decoding will be successful, else, the signal f cannot be recovered and the measurement vector z' is discarded. Based on the application, the encoder may either re-send z or may re-encode f with a higher redundancy code before retransmitting.

We would like to emphasize a property of erasure channels and linear coding here. Data that is missing from vector z is caused by the channel matrix C, which is generated by omitting rows from an identity matrix $\mathbb{I}_k$ at the indices corresponding to the missing data. But, since z is formed using $z = \Omega f$, one may instead view the combined coding and loss process as one of coding alone, where $\Omega' = C\Omega$ is the equivalent coding matrix generated from by omitting $k - k'$ of its rows at the indices corresponding to the lost data. This means that missing data at the receiver can be considered the same as not having those rows in Ω to begin with. We will use this perspective later when we discuss properties of compressive sensing matrices.

Now, if we knew that the signal f contained redundancy, we could have compressed it before channel coding. We represent f using a sparse vector $x \in \mathbb{R}^n$, by transforming it through an orthonormal basis $\Psi \in \mathbb{R}^{n \times n}$ using $f = \Psi x$. For example, if f was an acoustic waveform and Ψ was the inverse FFT basis operator ($\psi_{\omega,j} = \frac{1}{\sqrt{n}} \exp(\mathrm{i}2\pi\omega j/n)$), x would be the Fourier coefficients of the wave-form. In the traditional (lossy) source-channel sequential coding process, the largest m ($m \ll n$) coefficients of x would be passed to the channel encoder. Let $y = Rx$ be the input to the channel coder, where $R \in \mathbb{R}^{m \times n}$ is a sub-matrix of $\mathbb{I}_n$ that defines the indices of x selected for transmission. The output at the sensor node would then be $z = \Omega y = \Omega R \Psi^{-1} f$, where Ω is now of size $k \times m$. At the receiving end, the channel decoder first recovers $\hat{y}$ and thus $\hat{x}$ using Equation (6.1)

(replacing $\hat{f}$ with $\hat{y}$) and then $\hat{f} = \Psi\hat{x}$.

6.2.2 Compressive Sensing Fundamentals

The theory of compressive sensing asserts that the explicit compression step $x = \Psi^{-1}f$ does not need to be performed at the encoder and that a much smaller ''incoherent'' transformation may be performed instead. We consider a sensing matrix $\Phi \in \mathbb{R}^{m\times n}$ that generates m ($m \ll n$) of these incoherent measurements directly by projecting the signal f in its native domain[2] through $y = \Phi f$. In the usual synchronous sampling regime, Φ is an identity matrix $\mathbb{I}_m$ and $m = n$. When employing compressive sensing, however, the sensing matrix may be generated pseudo-randomly using one of several distributions that ensure that sufficient unique information is captured with high probability. The questions that CS theory answers are: how can f be recovered from y, how many measurements m are required for accurate recovery and what sensing matrices Φ facilitate recovery. We summarize some key results from [Can08] and references therein.

The foundational argument behind much of compressive sensing is that although Φ is not full rank, x and hence f can be ''decoded'' by exploiting the sparsity of x coupled with the sparsity promoting property of the ℓ_1 norm. To accomplish this, we view y as being generated through $y = \Phi\Psi x = Ax$ instead of through $y = \Phi f$. Now, while the solution $y = Ax$ to leads to infinitely many options, the CS reconstruction procedure selects the solution with the least sum of magnitudes by solving a constrained ℓ_1 minimization problem [CT06]:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 1} \quad \text{s.t.} \quad y = A\tilde{x} \tag{6.2}$$

where $\|\tilde{x}\|_{\ell 1} \triangleq \sum_{i=1}^{n} |x_i|$ and, f is recovered using $\hat{f} = \Psi\hat{x}$ as before. To guarantee that the solution from Equation (6.2) is exact, a notion termed the restricted isometry property

[2]The native domain for typical analog-to-digital conversion is time, but in some cases like photonic ADCs [BCJ98], sampling occurs in the frequency domain.

(RIP) was introduced. We will return to the RIP shortly, but first explain how the above procedure could be extended to handle missing data.

6.2.3 Handling Data Losses Compressively

Since the compressively sampled measurements in y are a compact representation of f, a valid scheme to protect y from channel erasures would be to feed it to a channel coding block as before. Thus, the sensor would now emit the coded measurements $z = \Omega y = \Omega\Phi f$, with Ω being of size $k \times m$ $(k > m)$. At the receiver, recovering f proceeds by first recovering $\hat{y}$ from $z' = Cz$ using Equation (6.1) and then using Equation (6.2), if channel decoding succeeds.

If we consider each step in the above process, we see that compressive sampling concentrated the signal information in a set of m measurements and then channel coding dispersed that information to a larger set of k measurements. A natural question to ask is how the dispersion scheme differs in essence from the concentration scheme and whether they can be unified. The answer to this question is the crux of this chapter.

We argue that compressive sensing matrices not only concentrate but also spread information across the m measurements they acquire. This perspective is backed by Theorem (2) (below) and is the primary reason for the logarithmic rate overhead experienced by CS practitioners. Based on this observation, we propose that, an efficient strategy for improving the robustness of data transmissions is to augment the sensing matrix Φ with e additional rows generated in the same way as the first m rows. These extra rows constitute extra measurements, which, under channel erasures will ensure that sufficient information is available at the receiver. Note that oversampling in the native domain of f is a valid strategy too, but is highly inefficient. On the other hand, we will show next that if $k = m + e$ incoherent measurements are acquired through oversampling and e erasures occur in the channel, the CS recovery performance will equal that of the original sensing matrix on a pristine channel with high probability (w.h.p.). We denote the augmented

sensing matrix as $\bar{\Phi} \in \mathbb{R}^{k \times n}$ and the samples received at the decoder would be $z' = C\bar{\Phi}f$. The decoding and recovery procedures for this case are now performed in one-step using Equation (6.2), but constrained by z' (instead of y') to incorporate augmentation and losses: $z = C\bar{\Phi}\Psi x = C\bar{A}x = \bar{A}'x$.

To understand intuitively why such an approach might work, assume that both C and $\bar{\Phi}$ are generated randomly with each element being an instance of an i.i.d. random variable. From our earlier discussion on viewing missing measurements as missing rows in the coding matrix, we observe that with $k = m+e$ and e missing measurements, $\bar{\Phi}' = C\bar{\Phi}$ would be of size $m \times n$. Now, since each element of $\bar{\Phi}$ is i.i.d. and the erasure channel does not modify its value, we can view $\bar{\Phi}'$ as being generated with m rows to begin with, just like Φ. So, while Φ and $\bar{\Phi}'$ will not be identical, their CS reconstruction performance, which depends on their statistical properties and their size, will be equal. We explain this analytically in the following section.

6.2.4 Robustness of CSEC to Erasures

We can show that CS oversampling is not only a valid erasure coding strategy, but also an efficient one. In particular, we would like to show that if we augment the sensing matrix to include e extra measurements and any e from the set of $k = m + e$ measurements are lost (randomly and independently) in transmission, the performance of CS reconstruction is equal to that of the original un-augmented sensing matrix (with high probability). To accomplish this, we rely on results from compressive sensing theory. We define the restricted isometry constant δ_s of a matrix and reproduce a fundamental result from [Can08] that links δ_s to CS performance.

Definition 1. [Can08] For each integer $s = 1, 2, ...$, define the isometry constant δ_s of a matrix A as the smallest number such that

$$(1 - \delta_s) \|x\|^2 \leq \|Ax\|^2 \leq (1 + \delta_s) \|x\|^2 \tag{6.3}$$

holds for all s-sparse vectors x. A vector is said to be s-sparse if it has at most s non-zero entries.

Theorem 2. *[Can08] Assume that $\delta_{2s} < \sqrt{2} - 1$ for some matrix A, then the solution $\hat{x}$ to (6.2) obeys*

$$\|\hat{x} - x\|_{\ell_1} \leq C_0 \cdot \|x - x_s\|_{\ell_1} \tag{6.4}$$

and

$$\|\hat{x} - x\|_{\ell_2} \leq \frac{C_0}{\sqrt{s}} \cdot \|x - x_s\|_{\ell_1} \tag{6.5}$$

for some small positive constant C_0. x_s is an approximation of a non-sparse vector with only its s-largest entries. In particular, if x is s-sparse, the reconstruction is exact.

This theorem not only guarantees that CS reconstruction will be exact if $\delta_{2s}(A) < \sqrt{2}-1$ for an s-sparse signal, but also that if x is not strictly s-sparse, but is compressible, with a power-law decay in its coefficient values, ℓ_1 minimization will result in the best s-sparse approximation of x, returning its largest s coefficients. We will return to this property when we compare the performance of CSEC with traditional erasure coding. Note also, that this is a deterministic result.

CS theory also suggests mechanisms to generate matrices that satisfy the RIP with high probability. For example, it has been shown in [CT06] that if the matrix A is constructed randomly using an *i.i.d.* Gaussian r.v. such that $A_{ij} = \mathcal{N}(0, \frac{1}{n})$ or an equiprobable $\pm\frac{1}{\sqrt{n}}$ Bernoulli r.v., the number of measurements obeys

$$s \leq C \cdot \frac{m}{\log(n/m)} \tag{6.6}$$

with high probability. Using such matrices in low-power sensing devices, however, is difficult since implementing the sensing matrix $\Phi = A\Psi^{-1}$ involves sampling and buffering f and computing $y = \Phi f$ explicitly through complex floating point operations. It was also shown in [CT06] and in [RV06] that if A is constructed by randomly selecting the

rows of a Fourier basis matrix, such as Ψ, the number of measurements obeys

$$s \leq C' \cdot \frac{m}{\log^4(n)} \tag{6.7}$$

with high probability.

This is a significant result indeed because it implies that, if the signal is sparse in the Fourier domain, $\Phi = A\Psi^{-1}$ is essentially an $m \times n$ random sampling matrix constructed by selecting m rows independently and uniformly from an identity matrix $\mathbb{I}_n$. This Φ is trivially implemented by pseudo-randomly sampling f, m times and communicating the stream of samples and their timestamps to the fusion center. Matrix Φ can then be recreated at the fusion center from the timestamps. The limitation on Ψ being the Fourier basis was removed in [RV06], which showed that the bound in Equation (6.7) extends to any dense orthonormal basis matrix Ψ with uniform random sampling.

Assume that the transmission channel can be modeled using an independent Bernoulli process with mean loss probability p. Thus, the likelihood of any measurement being dropped in this memoryless erasure channel is equal and is p. To show now that CSEC with random sampling is efficient for this channel model, we need to show that reconstruction performance with $A = \Phi\Psi$, where $\Phi \in \mathbb{R}^{m \times n}$ and $\bar{A}' = C\bar{\Phi}\Psi$, where $\bar{\Phi} \in \mathbb{R}^{k \times n}$ is equal with high probability when $k = m/(1-p)$. The factor $1-p$ denotes the ratio of measurements lost in the channel. However, note that since Ψ is an orthonormal basis matrix, it is equivalent to show that sensing performance with Φ and $\Phi' = C\bar{\Phi}$ is identical w.h.p.

Our approach considers the Fourier random sampling strategy, where Φ is constructed by selecting m rows independently and uniformly from an $n \times n$ identity matrix $\mathbb{I}_n$ and $\bar{\Phi}$ is constructed by selecting k rows independently and uniformly from $\mathbb{I}_n$. For the bound in Equation (6.7) to hold for matrix Φ, two conditions need to be met:

a. At least m samples need to be selected

b. The indices should be selected using a uniform random distribution so that each sample is equiprobable.

To show that $\bar{\Phi}'$ results in identical performance (w.h.p.) to Φ, we need to show that the above conditions hold equally. We first show that $\bar{\Phi}'$ satisfies condition (b). When the channel is memoryless with a loss probability p, an average of kp samples are lost in transmission and only k' samples are received. This can be modeled as a channel matrix C that is constructed by selecting k' rows independently and uniformly from $\mathbb{I}_k$. Let $\mathcal{S}_\Phi$ denote the set of unique sample indices that were selected using $y = \Phi f$ for the random sampling case. Therefore, the cardinality of $\mathcal{S}_\Phi$ would be $|\mathcal{S}_\Phi| = m$. Similarly, let the set of indices chosen in $\bar{\Phi}$ be labeled as $\mathcal{S}_{\bar{\Phi}}$ and the sample indices received by the decoder be $\mathcal{S}_{C\bar{\Phi}}$, where $|\mathcal{S}_{\bar{\Phi}}| = k$ and $|\mathcal{S}_{C\bar{\Phi}}| = k'$. Since Φ is constructed randomly and uniformly, the probability of a particular sample i from f being selected is:

$$\Pr\left[i \in \mathcal{S}_\Phi |\ \ |\mathcal{S}_\Phi| = m\right] = \frac{m}{n} \tag{6.8}$$

and for the oversampling case is:

$$\Pr\left[i \in \mathcal{S}_{\bar{\Phi}} |\ \ |\mathcal{S}_{\bar{\Phi}}| = k\right] = \frac{k}{n} \tag{6.9}$$

Proposition 3. *If we transmit k randomly chosen samples over an independent Bernoulli channel with a probability of lost transmission over the channel as p, the probability of the sample i being received in the samples is:*

$$Pr\left[i \in \mathcal{S}_{C\bar{\Phi}} |\ \ |\mathcal{S}_{C\bar{\Phi}}| = k'\right] = \frac{k'}{n} \tag{6.10}$$

Proof. This result is intuitive and straightforward to prove.

$$\Pr\left[i \in \mathcal{S}_{C\bar{\Phi}} \mid |\mathcal{S}_{C\bar{\Phi}}| = k'\right] \tag{6.11}$$

$$= \frac{\Pr\left[i \in \mathcal{S}_{C\bar{\Phi}},\ |\mathcal{S}_{C\bar{\Phi}}| = k'\right]}{Pr\left[|\mathcal{S}_{C\bar{\Phi}}| = k'\right]} \tag{6.12}$$

$$= \frac{\Pr\left[\text{receiving sample } i \text{ correctly}\right] \cdot \Pr\left[|\mathcal{S}_{C\bar{\Phi}}| = k' - 1\right]}{Pr\left[|\mathcal{S}_{C\bar{\Phi}}| = k'\right]} \tag{6.13}$$

$$= \frac{(1-p) \cdot \Pr\left[\text{selecting sample } i\right] \cdot \Pr\left[|\mathcal{S}_{C\bar{\Phi}}| = k' - 1\right]}{Pr\left[|\mathcal{S}_{C\bar{\Phi}}| = k'\right]} \tag{6.14}$$

$$= \frac{(1-p) \cdot \Pr\left[i \in \mathcal{S}_{\bar{\Phi}} \mid |\mathcal{S}_{\bar{\Phi}}| = k\right] \cdot \begin{pmatrix} k-1 \\ k'-1 \end{pmatrix} p^{k-k'}(1-p)^{k'-1}}{\begin{pmatrix} k \\ k' \end{pmatrix} p^{k-k'}(1-p)^{k'}} \tag{6.15}$$

$$= \frac{(1-p)k/n}{k/k'} \cdot \frac{p^{k-k'}(1-p)^{k'-1}}{p^{k-k'}(1-p)^{k'}} = \frac{k'}{n} \tag{6.16}$$

□

This means that, if the channel is modeled as an independent Bernoulli process and the input sample distribution is equiprobable over n samples, the output index distribution is also equiprobable over the set of correctly received samples. This proves condition (b). If we increase the number of samples by the ratio lost in the channel such that $k = m/(1-p)$, $\mathrm{E}\left[k'\right] = m$ and thus:

$$\Pr\left[i \in \mathcal{S}_{C\bar{\Phi}} \mid |\mathcal{S}_{C\bar{\Phi}}| = k'\right] = \Pr\left[i \in \mathcal{S}_{\Phi} \mid |\mathcal{S}_{\Phi}| = m\right] = \frac{m}{n} \tag{6.17}$$

This shows that to the decoder, the set of samples $\mathcal{S}_{C\bar{\Phi}}$ received through the memoryless binary erasure channel is indistinguishable from the original sample set $\mathcal{S}_{\Phi}$ and thus oversampling by a factor of $1/(1-p)$ achieves the same performance bound as the original sampling rate (with high probability).

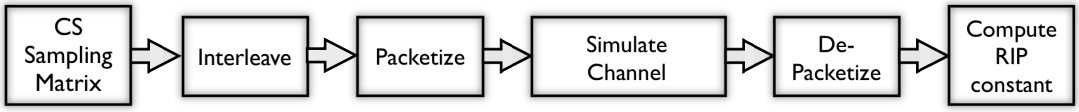

Figure 6.3: Steps followed in evaluating the RIP constant for different sampling matrices under various channel conditions.

6.2.5 CSEC Reconstruction when Redundancy is Insufficient

While the above result indicates that signal recovery using CSEC is exact if the redundancy $k-m$ is higher than the number of erasures $k-k'$, it can also be shown that when $k' < m$, recovery can still proceed but results in an approximation of the signal. This is in contrast to traditional erasure coding schemes that necessitate that the matrix $C\Omega$ be invertible for any reconstruction to occur. To prove this we use Theorem (2) when applied to compressible signals. Assume that the signal of interest f in its compressed form x has its ordered coefficients decaying according to a power law such that $|x|_{(l)} \leq C_r l^{-r}$, where C_r is a signal dependent constant, $r \geq 1$ and $|x|_{(0)} \geq |x|_{(1)} \geq \cdots \geq |x|_{(n)}$.

Assume also that the bound in Equation (6.7) is satisfied in equality for some choice of m and s. Now, when $k' < m$ because of heavy data loss, k' will not meet the bound for sparsity s. However, k' is guaranteed to satisfy the bound for some lower sparsity $s' \leq s$ (by extension from [RV06]). For the set of k' measurements received, Theorem (2) guarantees that CS reconstruction will result in the best s'-sparse approximation of x, returning its largest s' coefficients. This implies, then, that the increase in the ℓ_1 norm of the reconstruction error with k' measurements will be limited to $\epsilon \leq C_0 \left\| x_s - x_{s'} \right\|_{\ell_1}$ with high probability. We study the effect of this reconstruction error on the probability of recovery in Section 6.3.2.

6.3 Evaluating CS Erasure Coding

In order to verify the performance of compressive erasure coding, we analyze the sampling matrix $\bar{A}'$ that identifies which measurements were received at the decoder.

6.3.1 Verifiable Conditions using RIP

6.3.1.1 For a Memoryless Erasure Channel

We model the erasure introduced by the transmission channel with an average loss probability, $p \in [0, 1]$. We initially assume an independent Bernoulli process so that the probability of any sample being dropped is equal to p. The question we would like to address is how the loss of $k \cdot p$ packets dropped in this way affects CS reconstruction. From Theorem 2, we understand that reconstruction accuracy depends on the RIP constant δ_{2s} of A. To evaluate the extent of performance loss through the erasure channel, we thus rely upon quantifying $\delta_s(A)$. Computing $\delta_s(A)$ exactly from Definition 1, however, is exhaustive because it is defined over all s-sparse vectors. We approximate it by evaluating the eigenvalues of the Grammian [BDD08] over 10^3 random $s \times n$ sub-matrices. Increasing this number to 10^6 results in little improvement.

The entire process used to evaluate the efficacy of CS erasure coding is shown in Figure 6.3. We first generate a random sampling matrix $A = \Phi\Psi$ of size $m \times n$ as described in Section 6.2.2. The samples may then undergo an interleaving process to randomly shuffle the data before transmission. Since samples are not sent in isolation but may be packetized before transmission, a packetization block is included before transmission through the lossy channel. The un-augmented sensing matrix A will be modified by the channel so that $A' = CA = C\Phi\Psi$ and we also have an augmented sensing matrix $\bar{A} = \bar{\Phi}\Psi$ generated at the source of size $k \times n$ with $k > m$ and its received counterpart $\bar{A}' = C\bar{A}$. Testing the performance of CSEC numerically then proceeds by comparing whether $\delta_s(\bar{A}') \leq \delta_s(A)$. Equality ensures that the CS decoder would be able to achieve reconstruction accuracy identical to an un-augmented sensing matrix with a pristine channel. If $\delta_s(\bar{A}') < \delta_s(A)$, it means that the decoder has more measurements through $\bar{A}'$ than through the original A and would lead to a higher probability of exact recovery.

The result from this calculation for 1000 256×1024 randomly generated random sampling matrices with the Fourier basis for reconstruction is shown in Figure 6.4. The

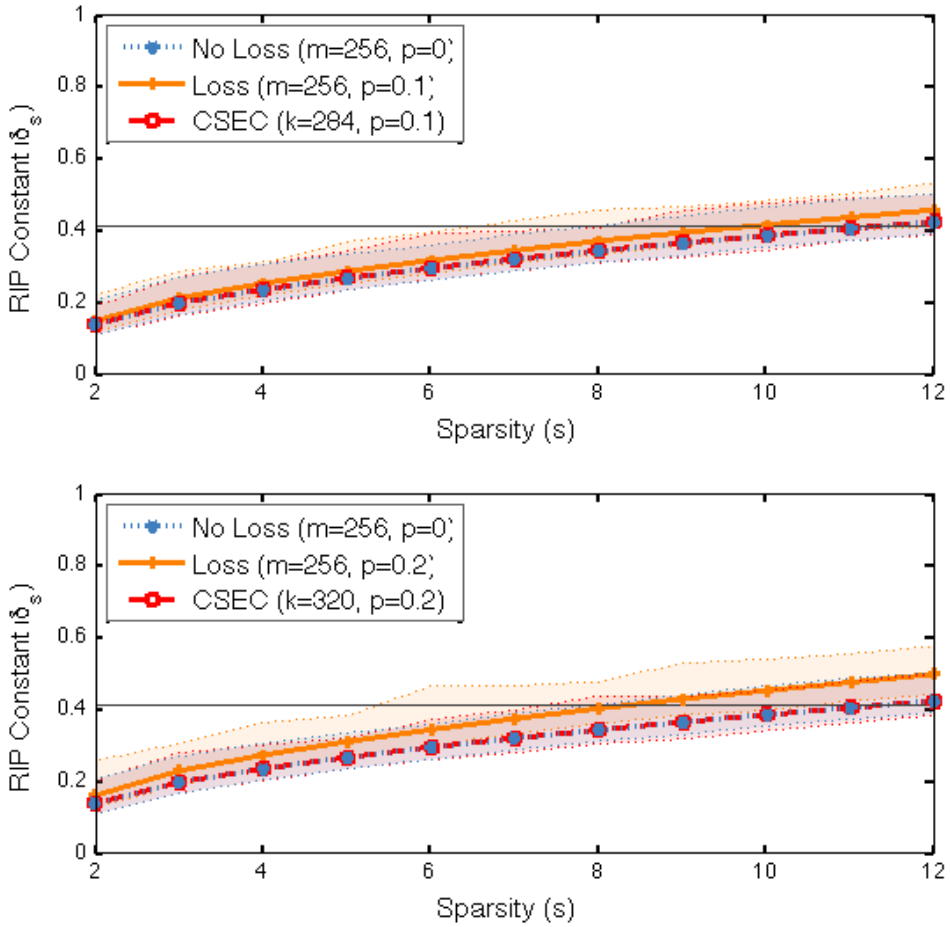

Figure 6.4: Effect of data loss on RIP constant with average loss probability p in a memoryless Bernoulli channel. Also shown is the improvement in RIP constant by increasing rate to $m/(1-p)$ and shuffling samples prior to transmission. Shading indicates the min-max across 1000 Monte-Carlo runs. The $\delta_{2s} < \sqrt{2}-1$ bound is included for reference.

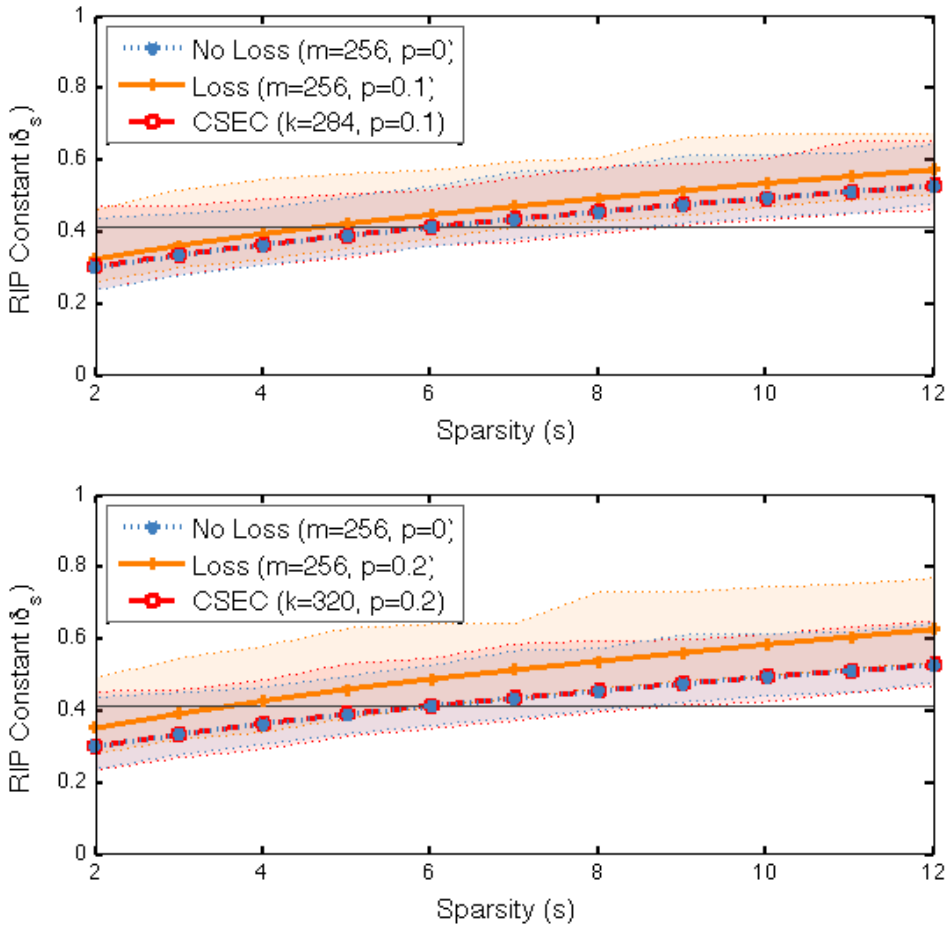

Figure 6.5: Effect of data loss on RIP constant with average loss probability p in a memoryless Bernoulli channel with Gaussian random sampling.

solid blue curve labeled "No Loss" indicates $\delta_s(A)$ and the shading illustrates the min-max values over all Φ about the mean. With loss probability $p = 0.2$, we see an increase in RIP constant, which implies that the sparsity for guaranteed ℓ_1 reconstruction drops (from about 6 to about 4 based on the bound $\delta_{2s} < \sqrt{2}-1$) (gray horizontal line) from Theorem (2). Note, though, that while this bound is known to be conservative, enumerating the RIP constant in this way clearly indicates the loss in reconstruction accuracy that may be expected by losing 20% of the sampled measurements.

From Proposition (3), we see that the probability distribution of time-stamps extracted from Φ and $\Phi' = C\Phi$ are identical when C comes from a memoryless (independent

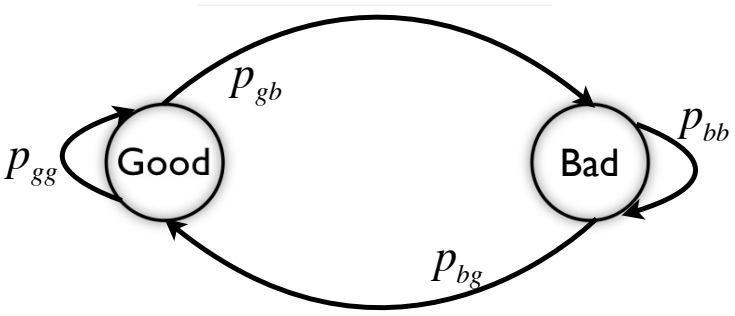

Figure 6.6: Steps followed in evaluating the RIP constant for different sampling matrices under various channel conditions.

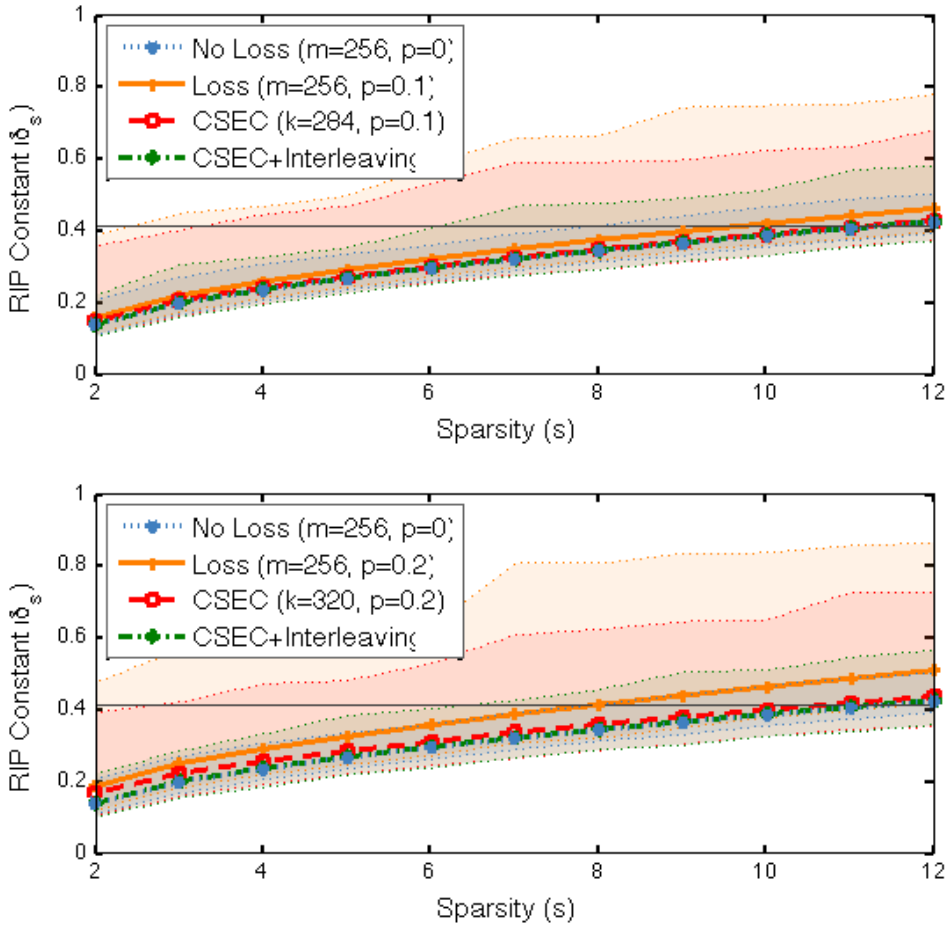

Figure 6.7: Effect of data loss on RIP constant with average loss probability $p = 0.1$ in a Gilbert-Elliott Channel ($p_{g\to b} = 0.0312$, $p_{b\to g} = 0.125$). Also shown is the improvement in RIP constant by increasing rate to $m/(1-p)$ and shuffling samples prior to transmission. Shading indicates the min-max across 1000 Monte-Carlo runs. The $\delta_{2s} < \sqrt{2} - 1$ bound is included for reference.

Bernoulli) channel. This means that losses due to the channel are indistinguishable from an *a priori* reduced random sampling rate at the sensor node. This in turn means that, if the channel is not congested, increasing the sensing rate by a factor of $p/(1-p)$ will restore the delivery rate to $k' = m$. The effect of this increase is substantiated in Figure 6.4 and establishes $\delta_s(\bar{A}') \approx \delta_s(A)$ for the independent Bernoulli channel. We see that not only does the mean RIP constant improve to its original value but that the range of variation also recovers to the "No Loss" baseline. Note also, that the minimum values of $\delta_s(\bar{A}')$ are below $\delta_s(A)$ suggesting that some instances of $\bar{\Phi}$ coupled with channel loss actually deliver better-than-baseline performance.

To compare performance across different sampling schemes, we further evaluate the RIP constant for a sensing matrix that is constructed using the random Gaussian method as described earlier. In this case, the reconstruction is performed in the identity domain with $A = \Phi$. It has been shown in [CT06] that the Gaussian sampling technique has equivalent performance across any orthonormal reconstruction basis and the identity matrix was chosen for computational ease. The result of this computation is shown in Figure 6.5 for a memoryless channel. Here too, we observe that while the RIP constant is slightly higher for the Gaussian sampling case, increasing the rate by the amount lost in the channel recovers the performance guaranteed by compressive sensing.

6.3.1.2 Interleaving for Bursty Channels

Realistic wireless channels exhibit bursty characteristics [SKA08]. To estimate the effect of CSEC performance with bursty channels, we use the popular Gilbert-Elliott (GE) model [Ell63], which is both tractable to use and accurate in describing many wireless channels (including those in mobile environments [WM95]). GE channels are modeled using a stationary discrete-time binary Markov process as shown in Figure 6.6. Within the two states, marked good and bad, the probability of loss in the good state is $p_g = 0$ and the probability of loss in the bad state is $p_b = 1$. The probability of transition from one state to another is marked as $p_{g\to b}$ and $p_{b\to g}$. To maintain consistency with the

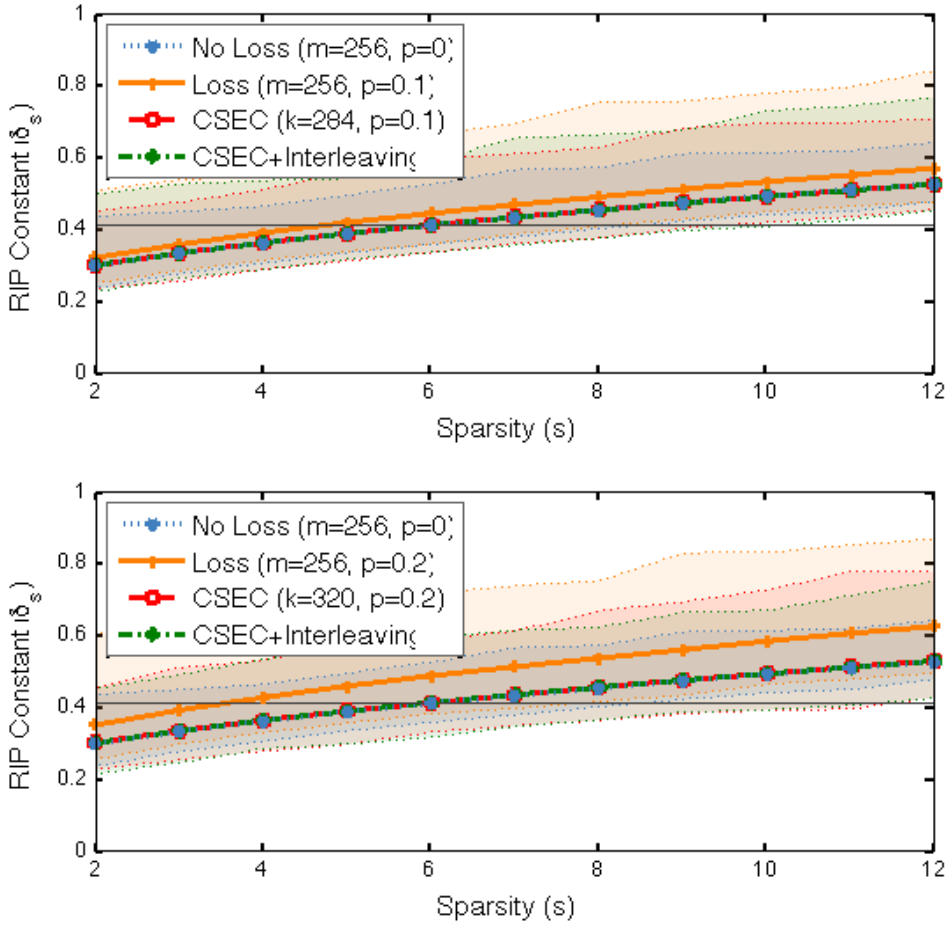

Figure 6.8: Effect of data loss on RIP constant with average loss probability $p = 0.1$ in a Gilbert-Elliott Channel ($p_{g\to b} = 0.0312$, $p_{b\to g} = 0.125$) with Gaussian random sampling.

memoryless channel performance studies, we compute the transition probabilities based on the average loss probability p and the expected burst size b, using the relationships:

$$p_{g\to b} = \frac{p}{b(1-p)} \tag{6.18}$$

$$p_{b\to g} = \frac{1}{b} \tag{6.19}$$

To test the effect on $\delta_s(A')$ and hence CS reconstruction, we constructed a GE model with an expected loss burst of $b = 8$ samples and $p = 0.1$ and $p = 0.2$. While $b = 8$ constitutes an extreme condition of burstiness, it is instructive to see its effects on CS recovery. Thus, the same number of samples are delivered to the fusion center as with the Bernoulli channel, but with a modified index distribution. The effect of this change is immediately evident in the spread of δ_s in Figure 6.7 indicating that some Φ matrix choices will be particularly bad for a GE channel. While an increase in sensing rate improves the mean RIP constant (though not reaching the baseline), the spread still remains high.

This variation issue can be resolved by also applying randomized interleaving [MB89] prior to transmission, which results in a roughly uniform distribution of the sample losses. It can be shown that interleaving recovers the original time-stamp distribution (up to a bound) and Figure 6.7 and 6.8 illustrates this empirically. Note, however, that interleaving requires buffering y (though not f), which increases decoding latency. Interestingly, we observe that the Gaussian random projection technique in Figure 6.8 is unaffected by interleaving. In fact, using Gaussian projections delivers near baseline performance with or without interleaving (min-max variation is not perfect). This is because the sensing matrix Φ is dense (compared to the one for random sampling) with each element within it being i.i.d Gaussian. This means that every measurement in y has a random, independent but statistically identical contribution from every element in f. Since interleaving the measurements y is equivalent to shuffling the rows of the matrix Φ, interleaving does not affect the statistical properties of Φ.

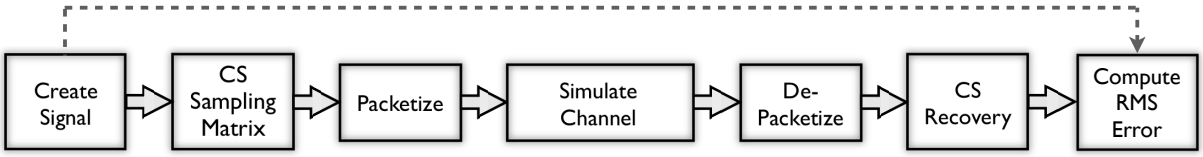

Figure 6.9: Steps followed in evaluating the probability of recovery from the RMS error for different sampling matrices under various channel conditions.

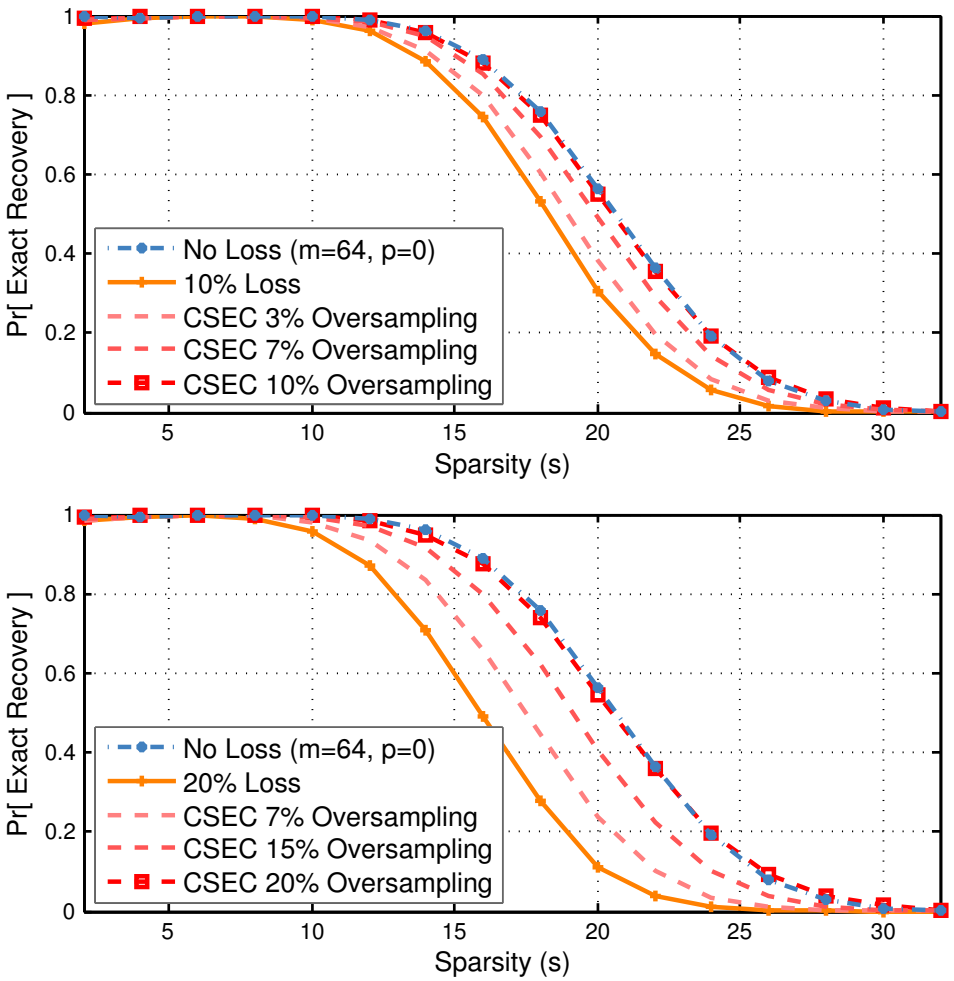

Figure 6.10: Effect of data loss on the probability of recovery with average loss probability p in a memoryless Bernoulli channel with Fourier random sampling.

6.3.2 Signal Reconstruction Performance

Evaluating the RIP constant provides theoretical insight into what the performance gain would be when using CSEC. In this section, we study the practical implications by evalu-

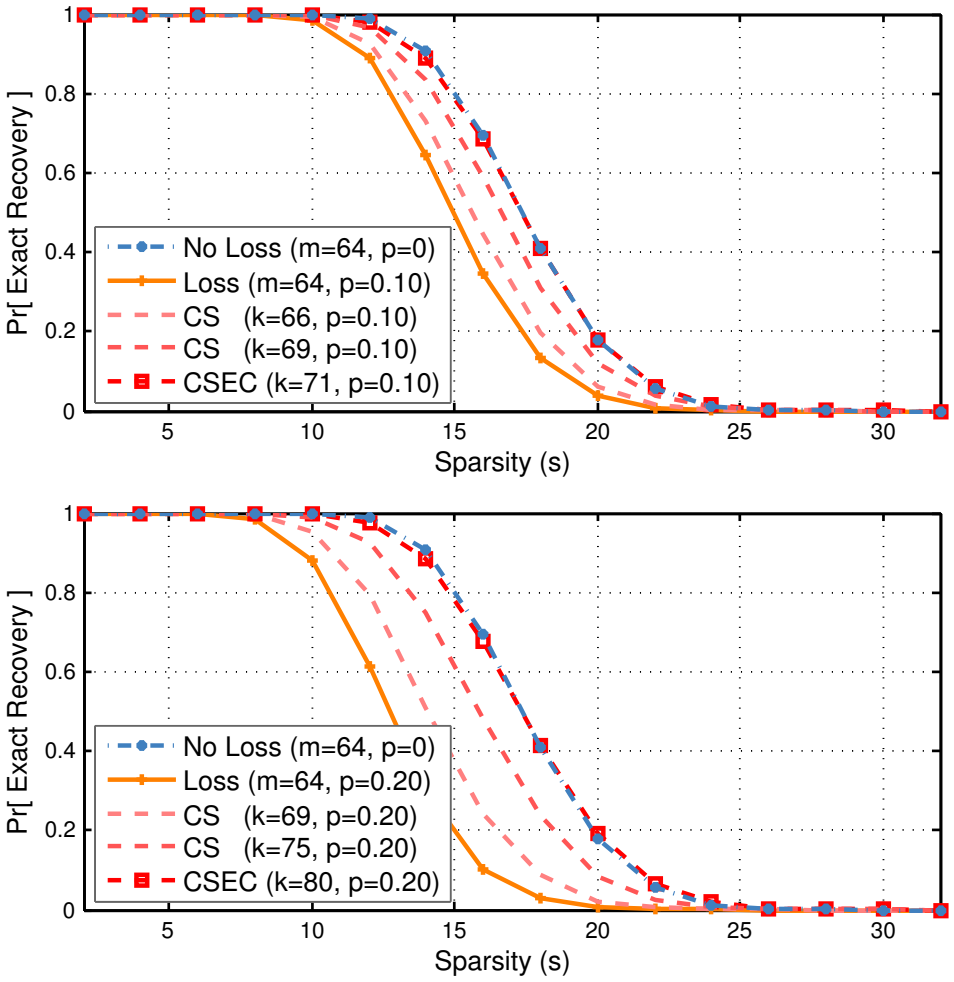

Figure 6.11: Effect of data loss on the probability of recovery with average loss probability p in a memoryless Bernoulli channel with Gaussian random sampling.

ating the probability P_{ex} with which CSEC could deliver the original signal exactly. We do this by performing a Monte-Carlo simulation over 10^4 random instances of a length 256 sparse signal and computing how often CS erasure coding results in exact recovery. Figs. 6.10 to 6.13 illustrate the comparative performance of using Fourier random sampling and the Gaussian projection method for CSEC. For each plot, the "No Loss" curve indicates the baseline with $m = 64$ and the "Loss" curve indicates the probability when no over-sampling is performed. The "CSEC" (red) curve indicates P_{ex} with over-sampling at $k = m/(1-p)$ and the two "CS" curves indicate intermediate values with $m < k < m/(1-p)$. The x-axis indicates the number of non-zero coefficients in x.

Three channel models have been used to generate these figures. Figs. 6.10 and 6.11 mimic the channel model used in Figs. 6.4 and 6.5, a memoryless erasure channel modeled as an independent Bernoulli process. We see for both Fourier random sampling and Gaussian projections that, when $k - k' = 16$ measurements are lost on the average, $k = 80$ recovers performance to the original $m = 64$ level. Observe that if the bound in Equation (6.7) is not met (beyond about $s = 10$), the performance for a particular k drops gradually with s. Note, also, that P_{ex} decays quicker to 0 in the case of Gaussian projections and we see while comparing Figs. 6.4 and 6.5 that this is because the RIP constant is also higher for the latter. The intermediate values of k in both cases deliver intermediate levels of quality as predicted in Section 6.2.5.

Figs. 6.12 and 6.13 use the same Gilbert-Elliott channel model as Figs. 6.7 and 6.8 with $p = 0.1$ and $p = 0.2$ with $b = 8$. It is striking to note that due to the burstiness of the channel, the performance of neither Fourier random sampling nor Gaussian projections reaches the baseline for low sparsity levels. Further, the highest s for which $P_{ex} = 1$ has gone down substantially for the lossy scenarios and the slope of the curve is also reduced. The reason for this is that the distribution of received sample lengths k' across the Monte-Carlo runs is skewed and asymmetric about the mean for bursty channels, whereas it is symmetric about $k(1-p)$ and is Gaussian for a memoryless channel . As shown in Figure 6.14, the mean value of k' for CSEC across runs is $\mu^{k'} \approx 64$ for both the

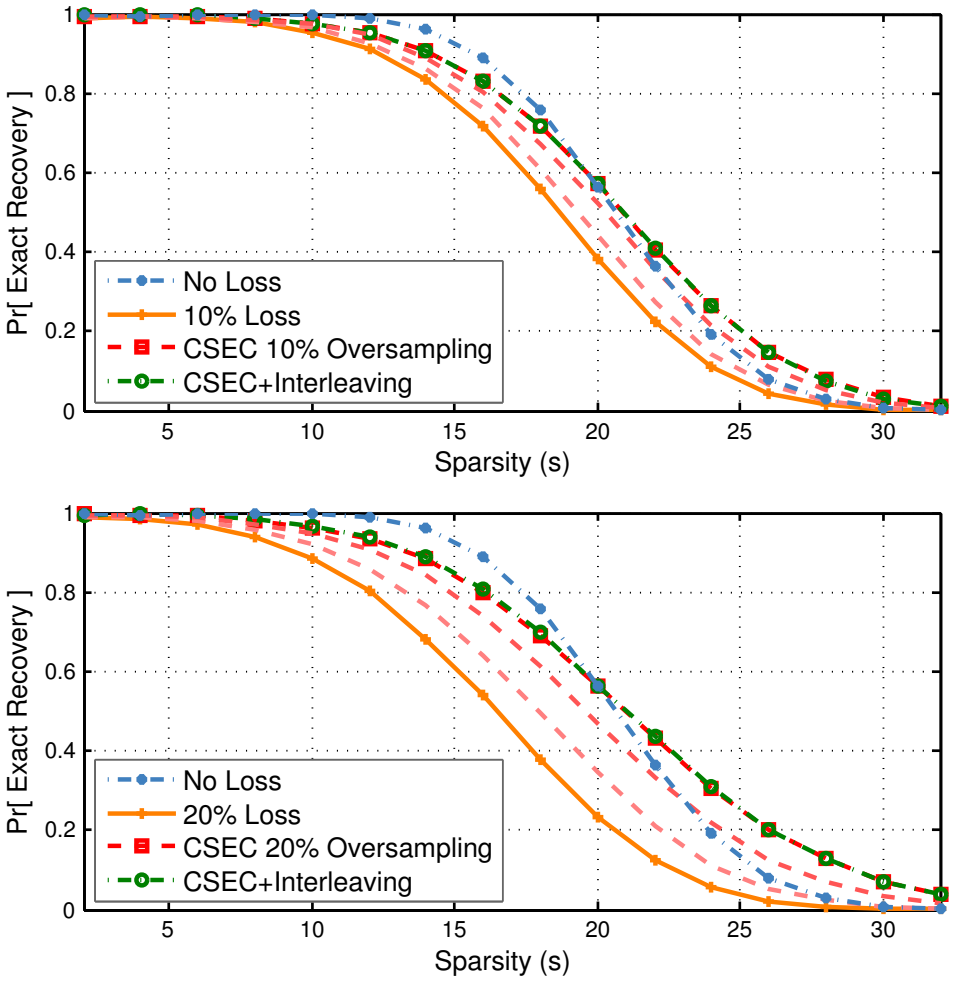

Figure 6.12: Effect of data loss on the probability of recovery with average loss probability $p = 0.1$ in a Gilbert-Elliott Channel ($p_{g\to b} = 0.0312$, $p_{b\to g} = 0.125$) with Fourier random sampling.

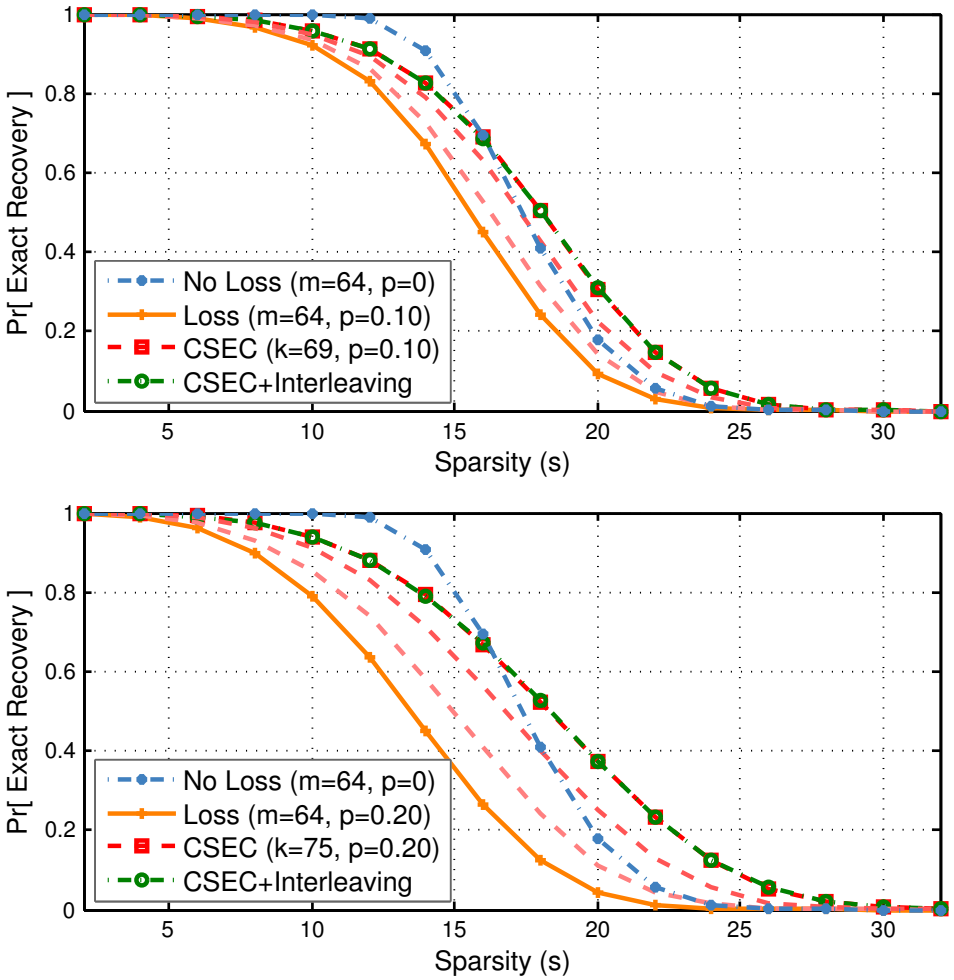

Figure 6.13: Effect of data loss on the probability of recovery with average loss probability $p = 0.1$ in a Gilbert-Elliott Channel ($p_{g\to b} = 0.0312$, $p_{b\to g} = 0.125$) with Gaussian random sampling.

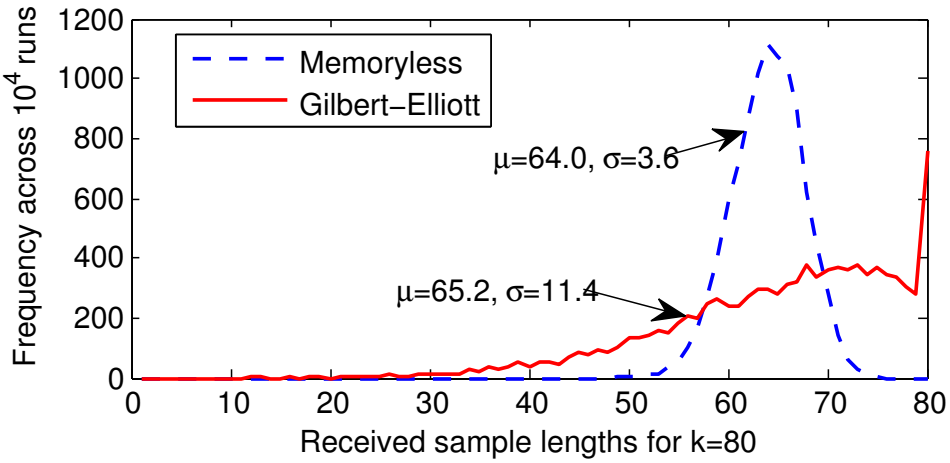

Figure 6.14: Change in the distribution of sample lengths when passed through a memory-less channel and a Gilbert-Elliott channel.

Bernoulli channel and the GE channel but their standard deviations are $\sigma^{k'}_{Bern} \approx 3$ and $\sigma^{k'}_{GE} \approx 10$ respectively. It's interesting, though; the sample length distribution is skewed toward higher k' and the result is that the probability of reconstruction at larger sparsity levels is actually higher than the baseline. An unexpected result from Figure 6.12 is that interleaving makes little or no practical difference to Fourier random sampling.

Figure 6.15 has been generated using a wireless network trace from the CRAWDAD database [ISK08]. The particular trace we selected used sensor nodes with an IEEE 802.15.4 radio transceiver placed about 12m apart between two different floors of a university building. This trace had the highest loss probability and burstiness across the 27 traces collected with $p \approx 0.15$ and $b = 1.2$. We built a GE channel model based off the trace and simulated the probability of exact recovery as before. There is very little burstiness in the channel and Figure 6.15 shows that CSEC will be able to deliver near baseline performance with either Fourier random sampling or Gaussian projections.

6.3.3 CSEC Implementation Costs

We can quantify the energy efficiency gains that CS promises too. In particular, we use random sampling with $n = 256$ and compare it to two cases – first, where a standard

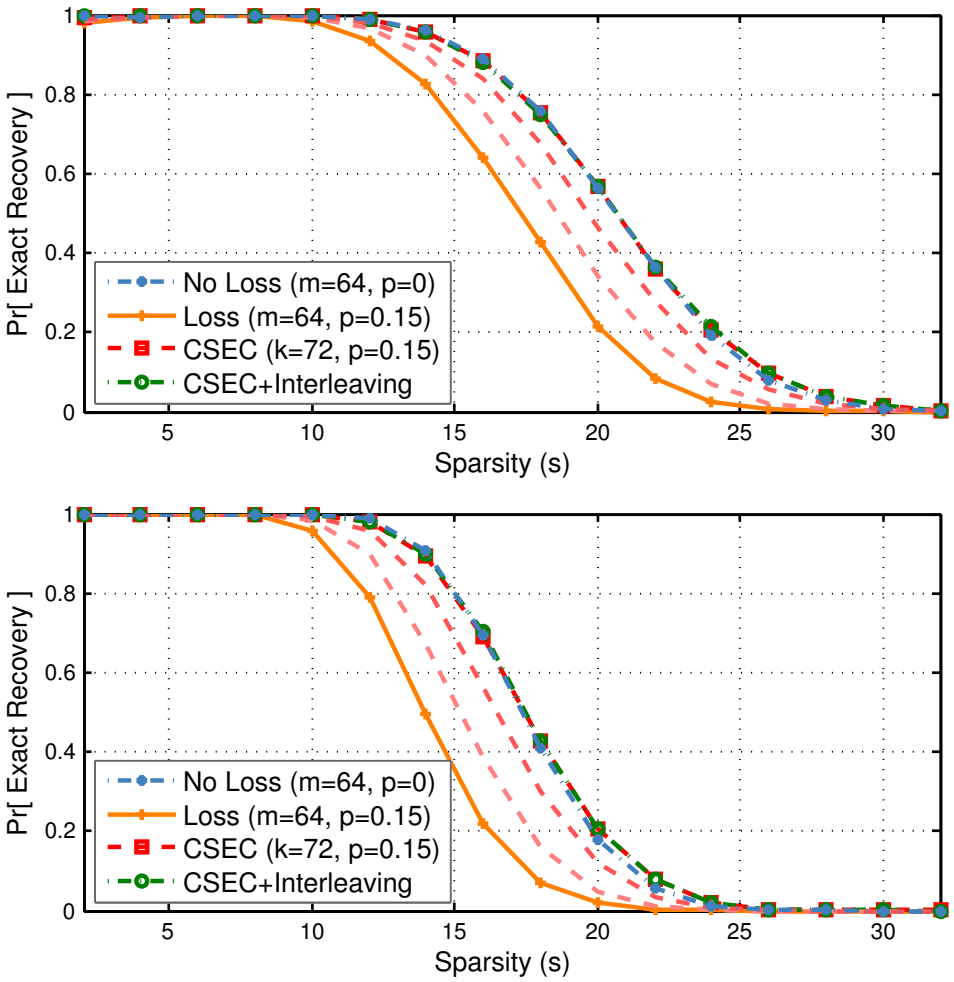

Figure 6.15: Effect of data loss on the probability of recovery with average loss probability $p = 0.15$ with a real wireless network trace from CRAWDAD database with Fourier (top) and Gaussian random sampling (bottom).

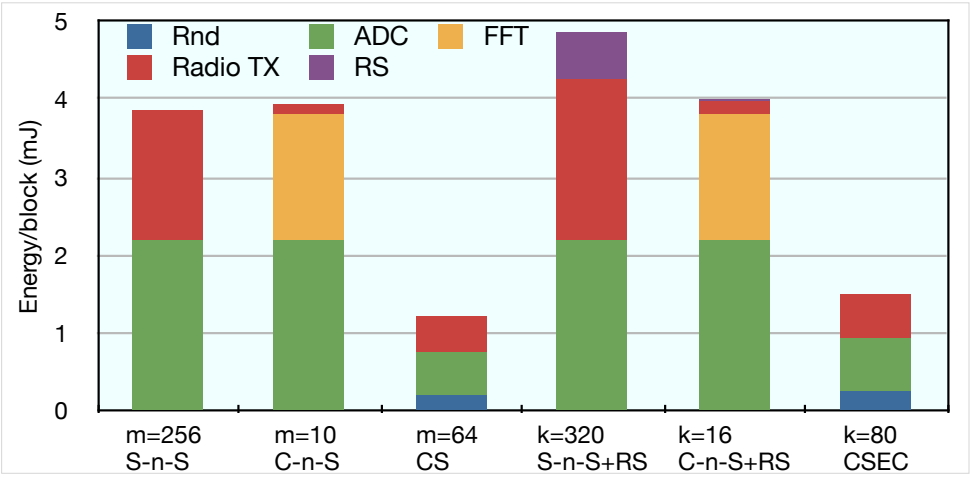

Figure 6.16: Energy consumption comparison for different sampling strategies (Sample-and-Send, Compress-and-Send and Compressive Sensing).

$(255, 223)$ Reed-Solomon (RS) [RS60] code is applied to a set of 256 raw 16-bit samples and second, where RS is applied to a compressed version of the signal. We assume the signal is sparse ($s \leq 10$) in the Fourier domain and use 256-point FFT for source compression. Figure 6.16 shows this comparison, which also includes energy consumption costs without RS. The data has been extracted using a cycle and energy accurate instruction-level simulator [TLP05] available for the popular MicaZ sensor platform. While the numbers are specific to this platform, the insight from these results can be applied more generally.

We have split the costs among five blocks, which are significant for the comparison – random number generator (for CS), ADC, FFT processing, radio transmission and Reed-Solomon coding. The total energy consumption of sample-and-send and compress-and-send is almost equal without RS, with the radio taking a large chunk of the former and the FFT routine consuming half of the latter. Notice also, that the ADC energy consumption is substantial since both these techniques need to operate on the entire signal vector. On the other hand, the CS routine at $m = 64$ requires a fraction of the ADC and radio, but incurs an overhead for generating random numbers. The current implementation uses an inexpensive 16-bit LFSR for pseudo-random number generation. Figure 6.17 illustrates the marginal improvement that can be realized using more computationally

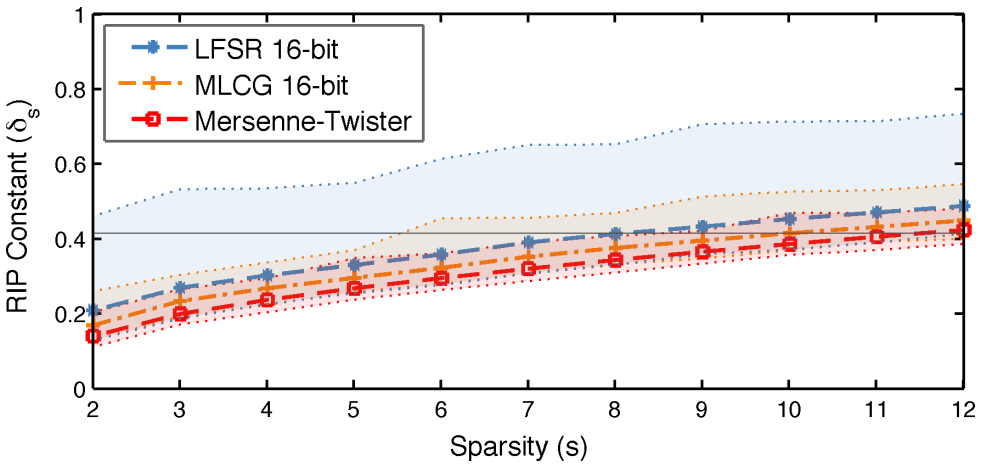

Figure 6.17: Comparison of RIP performance of different Pseudo-random number generators for Fourier Random Sampling.

complex generators.

When the data is RS encoded before transmission, the energy consumption of the sample-and-send strategy jumps considerably, whereas the increase for compress-and-send is negligible, because it is sending at most 10 coded symbols (with 6 parity symbols). We chose $s \leq 10$ since that is the threshold below which $m = 64$ in the lossless case and $k = 80$ for a memoryless erasure channel result in exact CS recovery (refer Figs. 6.10 and 6.11). This means that, with a $p = 0.2$ memoryless channel, all three strategies on the right would deliver equivalent recovery performance. When comparing encoding cost, however, CS erasure coding is 2.5× better than doing local source compression and 3× better than sending raw samples.

6.4 Related Work and Discussion

Recovering from erroneous and missing data in transmission systems is a classic issue that has received much attention over the years, with ARQ retransmissions and forward error correction (FEC) being used routinely today, sometimes simultaneously, at different layers of the communication protocol stack. For sensor networks, however, the simplicity of ARQ has retained it as the dominant form of error recovery. Many researchers have

questioned this recently and evaluated FEC techniques through lab experiments.

For example, Schmidt, et. al. [SBW09] focused on convolutional coding to show that a modified Turbo code is quite feasible on a MicaZ platform. While they report that the energy consumption using Turbo codes is about 3.5× of its un-coded counter-part, the overall energy efficiency is better considering retransmission costs, especially on high loss links. They also show that the computational complexity of Turbo encoding is practical, but only with low-rate data transfers (they tested with one packet every second). Jeong, et. al [JE07] proposed using a simpler error correcting code for only single and double-bit errors to reduce this computation burden. They illustrate, through experimental data that long error bursts are rare in static sensor networks and argue that the complexity of RS or LT codes is unwarranted. They show that their error correcting code reduces the packet drop rate almost to zero for outdoor deployments. However, due to a higher frequency of multiple-bit errors indoors, recovery remains imperfect there. We've shown that CSEC can provide both computational benefits as well as recovery performance that parallels state-of-the-art erasure correcting codes. To use CSEC, however, one must have a good understanding of the physical phenomena being acquired and the domain it can be compressed in.

Further, Wood, et. al. [WS09] recently reported the use of online codes in low-power sensor networks. Online codes are a form of digital fountain codes, such as the LT codes [Lub02], but are simpler to encode and decode. They propose a lightweight feed-back mechanism that allows the encoder to cope with variations in link quality rapidly and efficiently. They point out, however, that multiple parameters need to be tuned in order for the coding to be efficient. This is similar to the sensitivity of LT codes to the degree distribution [Lub02]. While our current work has focused on block wise decoding and makes analogies to other linear block coding strategies such as BCH codes, CSEC can be used in a ''rate-less'' mode as well, similar to fountain codes. Asif, et. al. [AR08] demonstrate this as a way of streaming incoherent measurements and describe a homotopy based approach that performs iterative decoding as measurements are received.

We have described CSEC for handling erasures in a channel, but CSEC can be extended to correct for errors in the sensor transduction process too. This means that a controlled amount of sensor noise can be cleaned from the acquired measurements during the decompression process. It is achieved by using Basis Pursuit De-noising [CDS98], which changes the equality constraint in Equation (6.2) to an inequality to account for variations due to noise. Note, however, that since CSEC utilizes features of the physical phenomenon and operates on the acquired signal, and not on the modulated symbols transmitted through the wireless channel, CSEC is not useful for correcting symbol errors at a communication receiver. A better approach to tackling the latter using ℓ_1 minimization techniques is discussed by Candes and Tao in [CT05].

In Section 6.3.1, we used the RIP constant of the sensing matrix as way of verifying its reconstruction performance. Another technique that was recently proposed, namely the null-space property [dG08], could also have been used. Until much recently, however, the null-space property was as difficult to compute as the RIP constant.

And finally, we note that the evaluation studies in Section 6.3 assumed that measurements are streamed to the receiver as they are acquired. If one packetizes the measurements for transmission, in a memoryless channel, the sample losses will no longer be independent and instead show high burstiness. An example of this is shown in Figure 6.18 and 6.19, which shows the probability of recovery when 8 measurements are transmitted in every packet.

6.5 Conclusion

We have explored the application of compressive sensing to handling data loss from erasure channels by viewing it as a low encoding-cost, proactive, erasure correction scheme. We showed that CS erasure coding is efficient when the channel is memoryless and employed the RIP to illustrate, that even extreme stochasticity in losses can be handled cheaply and effectively. We showed that for the Fourier random sampling scheme, oversampling

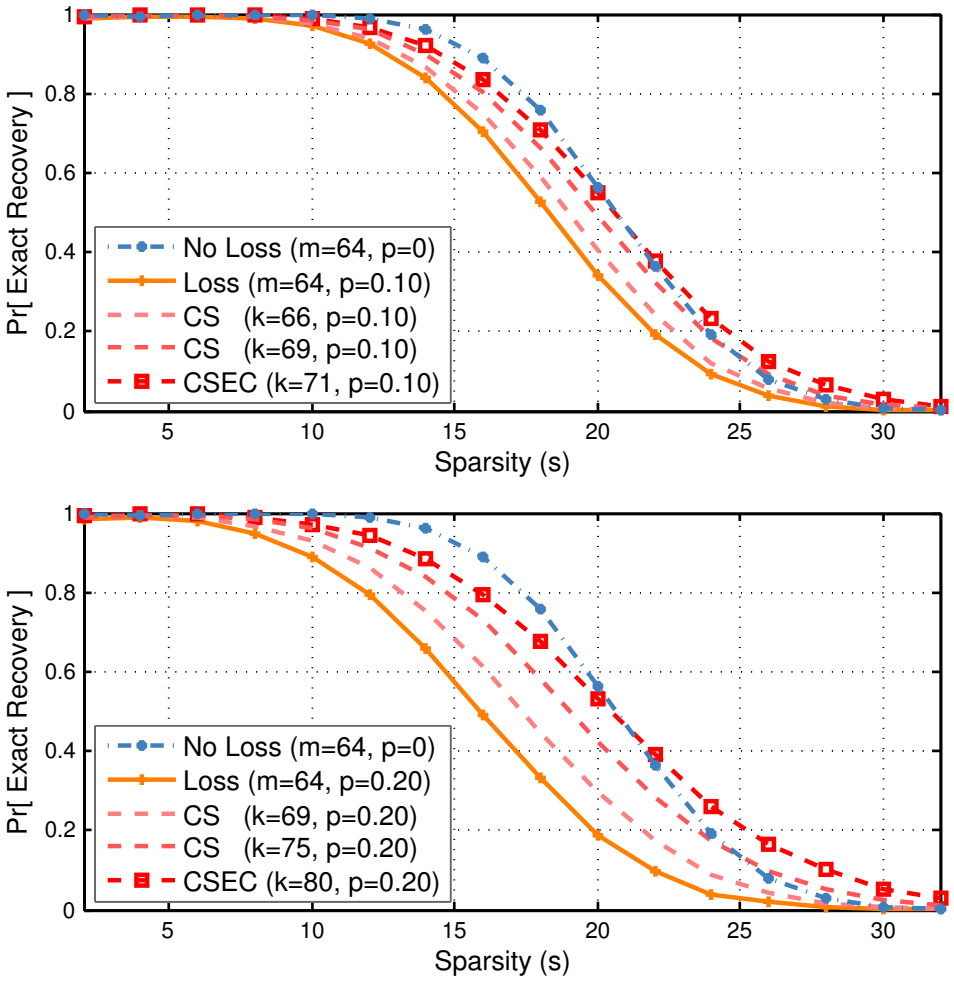

Figure 6.18: Effect of data loss on the probability of recovery with average loss probability with 8-sample packetization with Fourier random sampling.

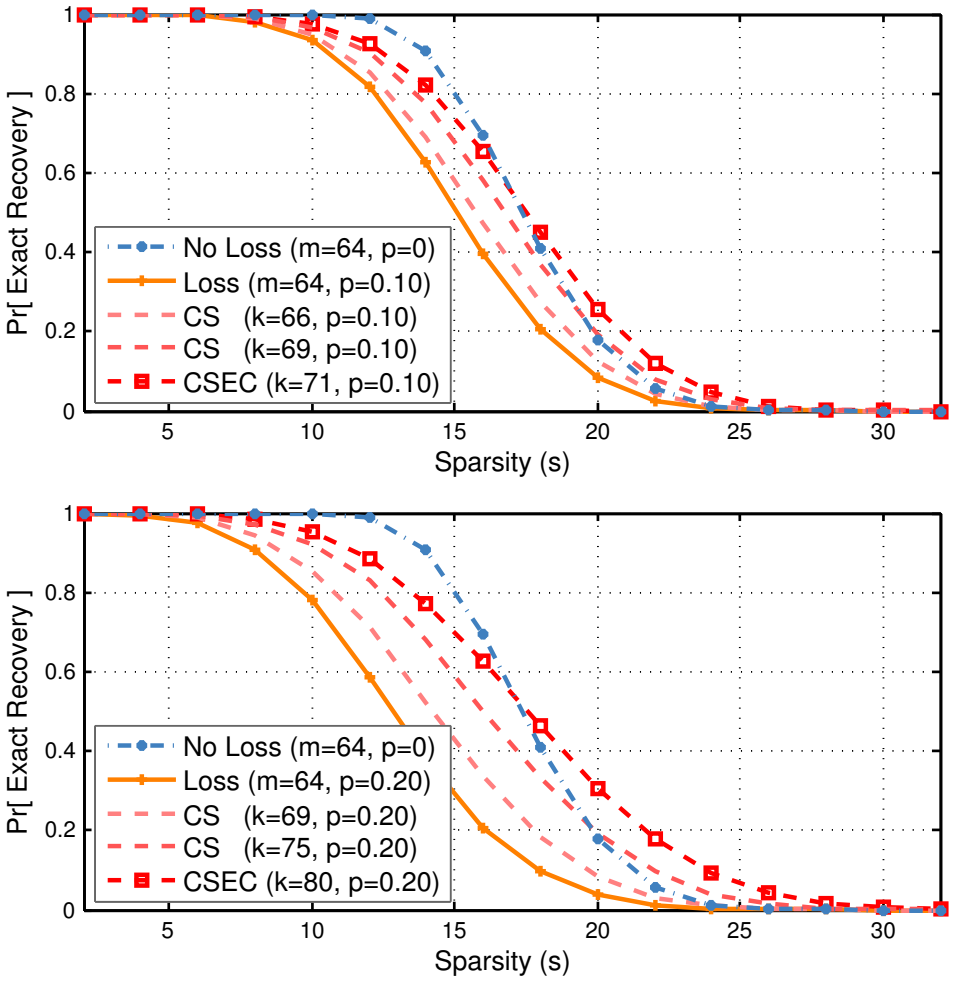

Figure 6.19: Effect of data loss on the probability of recovery with average loss probability with 8-sample packetization with Fourier random sampling.

is much less expensive than competing erasure coding methods and performs just as well. This makes it an attractive choice for low-power embedded sensing where forward erasure correction is needed.

CHAPTER 7

Energy Efficient Inferencing Through Compressive Sensing

7.1 Introduction

In the final chapter of this dissertation, we turn to the inferencing or the application block to consider how sampling optimization can be exploited for higher performance analytics. This work is projected to a context of generic wireless sensing for estimation and detection of various events. Since the primary use of sensing has been in hard-to-reach infrastructure-less environments, many wireless sensor network platforms are battery operated, leading to extreme energy constraints. Achieving high system lifetime, therefore, requires a concerted effort in reducing the sensor sampling, processing and radio communication costs while maintaining application level objectives.

For detection applications, WSN's to date bifurcate into those which can only pursue simple detection schemes and those which are not really low-power. In the latter case substantial computing and communication elements supported by large energy buffers and harvesting means replace tiny low-cost nodes. These systems are too costly to provide broad coverage and often last merely on the order of weeks. The former either push the detection problem into analog hardware – sleeping until woken by the analog trigger signal – or use a secondary digital processor to manage sampling and initial detection. The result

is either a large number of false alarms or an inability to detect more sophisticated trigger conditions (such as a specific acoustic signature).

In this chapter we present a novel approach – Weighted Basis Pursuit (WBP) – which eliminates this dilemma by providing continual coverage, sophisticated signal detection, and high accuracy (low false positive/negative rates) even at low signal-to-noise ratios (SNR) – all *without an additional power penalty.* This is achieved by tailoring work in compressive sensing (CS) to the challenges of WSN. For validation, a WBP-CS TinyOS module was developed and deployed on a testbed of MicaZ nodes. Demonstrable findings in this work clearly illustrate WBP-CS' utility. With a 30× sampling reduction, a deployment otherwise obtaining an impractical 1 month lifetime might only need a battery change once a year.

7.1.1 Compressive Sensing Overview

Many natural signals are compressible by transforming them to some domain -- eg. sounds are compactly represented in the frequency domain and images in the wavelet domain. But, compression is typically performed after the signal is completely acquired. Advances in compressive sensing [CRT06a, CDS98] suggest that if the signal is sparse or compressible, the sampling process can itself be designed so as to acquire only essential information. CS enables signal acquisition with average sampling rates far below the Shannon-Nyquist requirement and eliminates the explicit compression step altogether. This not only saves energy in the *ADC subsystem* through reduced sampling, the *processing subsystem* through reduced complexity (no explicit compression), and the *communication subsystem* through reduced transmission, but also enables the capture of substantially more complex signals where it would not be possible otherwise. For example, in applications interested in high-frequency acoustic signals, low power sensor network platforms, including MicaZ motes, can not sample at Nyquist rates [AGN].

Compressive sensing involves taking sample measurements in an ‘‘incoherent domain’’

through a linear transformation [BDD08]. This step may be viewed computationally equivalent to compression if this transformation sparsifies the signal. However, the key insight underpinning CS mechanisms is that, though the incoherent domain does not sparsify the signal directly, it describes the signal sufficiently uniquely for perfect recovery to succeed from a fraction of measurements. The computational advantage of doing this comes from the fact that some incoherent transformations can be done implicitly and cheaply at the source. To achieve this, however, the designer needs to (a) fabricate a domain that is incoherent with the sparsifying one and (b) transform the signal to it through sampling. Researchers have shown [BDD08], quite remarkably, that taking appropriate random projections of the signal before sampling satisfies both these requirements adequately for a large class of compressible signals.

The issue with applying CS in embedded systems is that while acquisition is cheap, reconstruction algorithms are computationally severe. Interestingly, it is this asymmetric architecture that makes CS an excellent choice for low-power distributed sensing with wireless sensor networks. This is because WSN deployments usually include a back-end data collection and fusion center (where the event is ultimately reported) that is endowed with a considerable amount of computing and storage ability. This means that, if the sensor nodes are able to take random projections of the sampled signal and communicate them to the fusion center, it is possible to reconstruct the signal with high probability using a fraction of what the Nyquist rate would have required.

7.1.2 Compressive Event Detection

While many CS mechanisms have focused on signal reconstruction, some researchers [HN07, DDW07, DBH08] have found that the number of samples needed to reliably detect features in the signal, even in a noisy and interference prone environment, can be considerably lower if full CS recovery is not required. A recent algorithm, IDEA [DDW06], demonstrates this by utilizing knowledge of *where* the event may be present in the sparse domain. For example, if the event of interest is an acoustic signature of known frequen-

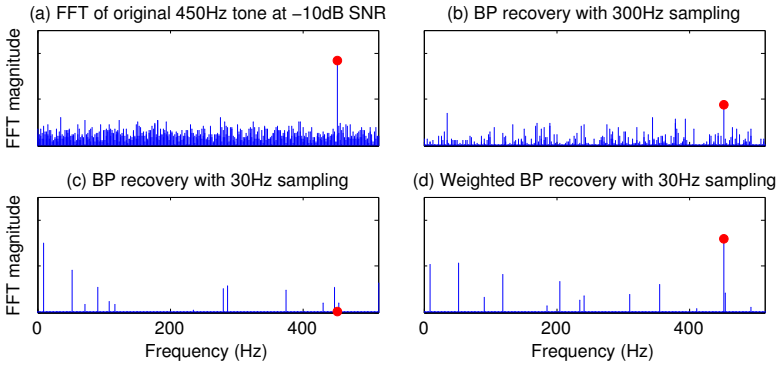

Figure 7.1: Frequency (FFT) coefficients for the CS reconstruction of a 450Hz tone at -10dB SNR with different sampling rates and recovery strategies.

cies, IDEA would look for the signal at those frequencies first. IDEA employs a greedy search procedure called Orthogonal Matching Pursuit (OMP [TG07]) to look for the best fitting frequency coefficients and furnishes its gains from being able to terminate OMP as soon as the desired signature is found in the signal.

An alternative to OMP often used for full CS recovery is called Basis Pursuit (BP), which poses the search for the sparse solution as a linear programming problem. Interestingly, it has been demonstrated that BP performs better than OMP in practice [CDS98]. Intuitively, this is because BP attempts to find the global minimum while OMP might get caught in a local dip. There are two drawbacks to using BP directly for detection, however. First, though BP can complete in polynomial time, the computational requirement is far higher than OMP [TG07]. And second, since BP attempts to reconstruct the signal completely, the number of measurements and hence energy required of the sender for comparable detection performance may actually be higher than IDEA. Assuming that the first drawback can be overlooked in a deployment with a capable back-end fusion center, this chapter focuses on overcoming the second.

Our solution tailors BP's linear programming problem to include prior knowledge of the event signature, similar in concept to IDEA. This is done by biasing components of the solution through a weighting matrix (details in Section 7.2) that prioritizes the search to prefer solutions with the known frequency indices. The effect of this weighting procedure

is that the biased components 'stand out' because they are artificially enhanced against background noise. This idea was inspired by the recent work of Candes, et al. [CWB08] that applies an iterative re-weighting procedure around BP to improve the quality of the compressive decoding solution.

Our proposed Weighted Basis Pursuit (WBP) is visually depicted in Figure 7.1 for the detection of a sinusoidal tone at 450 Hz in the presence of white noise. The reconstruction is performed in the Fourier (frequency) domain from randomly collected samples at different rates. When no weighting is applied, the average sampling rate needs to be as high as 300Hz to detect the tone -- the red dot in Figure 1.12b is just above the noise floor. While this is below the Nyquist rate of 900Hz, the gains are not impressive. If the sampling rate is lowered to 30Hz, no detection is possible (1.12c). However, if weighting is applied, the frequency tones immediately stand out (1.12d), implying a near 30× benefit over the Nyquist rate. A detailed evaluation of both simulated and experimental performance for different sampling rates in various noisy environments is deferred until Section 7.4.

7.1.3 Implementing Compressive Detection

Perhaps the most important aspect of implementing CS is the random linear transformation for sampling. Note that this transformation must not only be incoherent with the domain in which the signal is sparse, but it must also be substantially cheaper to implement than explicit compression. Much research has been undertaken in the CS community to search for suitable pseudo-random transforms but most require some form of additional front-end hardware before the ADC or some software oriented techniques that assume Nyquist sampling once more. A key contribution of this chapter is a demonstration of compressed detection mechanisms on commercial MicaZ sensor nodes without additional hardware. To achieve this, we use a uniform random sampling procedure that is known to be incoherent with any orthogonal basis [RV06], such as the Fourier basis. However, this random sampling is inherently non-causal and may also violate ADC hold times. In Section 7.3, we show how both these limitations can be overcome effectively

and inexpensively.

7.2 Weighted Basis Pursuit

Before we describe our detection procedure, we outline the BP estimation problem briefly. Assume that the signal of interest x is of length n and that a set of measurements z of length k, where $k \ll n$, are available to us, such that $z = \Phi x$, and Φ is the $k \times n$ measurement (random non-invertible transformation) matrix. Then, under the condition that x is sufficiently sparse, the solution to the following combinatorial optimization problem is known to recover the signal exactly:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 0} \quad \text{s.t.} \quad z = \Phi\tilde{x} \tag{7.1}$$

where $\|x\|_{\ell 0} \triangleq |\{i : x_i \neq 0\}|$. Equation 7.1 is NP-hard in general. Instead, a relaxed version of the problem that is convex is proposed:

$$\hat{x} = \underset{\tilde{x}}{\operatorname{argmin}} \|\tilde{x}\|_{\ell 1} \quad \text{s.t.} \quad z = \Phi\tilde{x} \tag{7.2}$$

where $\|x\|_{\ell 1} \triangleq \sum_{i=1}^{n} |x_i|$. It is shown in [CRT06a] that under the sparsity condition and that Φ satisfies the so-called restricted isometry property, the reconstruction $\hat{x}$ is exact with overwhelming probability [CW06]. Practically, this means that if the signal is sparse in the sensing domain, then taking k measurements through a suitable linear transformation Φ will be sufficient to reconstruct the signal. If the signal is not sparse in the sensing domain, but in another known domain, the reconstruction must be performed in two steps. Assume an invertible linear transformation Ψ of size $n \times n$ which compresses the signal using $x = \Psi y$, where y is also of length n but has very few non-zero coefficients. For example, if the signal x was a set of sinusoidal tones, it is not sparse in the time domain, but with Ψ as the inverse Fourier transform, y is sparse.

Then, under the condition that x is sufficiently compressible and that $\Phi\Psi$ satisfies the

restricted isometry property (or are mutually incoherent), the reconstruction $\hat{x}$ from the following optimization problem is exact with high probability [CW06, CRT06a]:

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmin}} \|\tilde{y}\|_{\ell 1} \quad \text{s.t.} \quad z = \Phi\Psi\tilde{y} \tag{7.3}$$

$$\hat{x} = \Psi\hat{y} \tag{7.4}$$

where $\|y\|_{\ell 1} \triangleq \sum_{i=1}^{n} |y_i|$, the sum of magnitudes (ℓ_1 norm) of the sparse coefficients. The above problem is termed Basis Pursuit. The intuition behind BP is that from the infinitely many solutions of $\tilde{y}$ that satisfy $z = \Phi\Psi\tilde{y}$, it is the simplest one, the one with the minimum sum of magnitudes that is most likely the right one. One may think of this as applying Occam's Razor, albeit rather uniquely. The mathematical theory behind BP is well developed and we refer the interested reader to [CW06], which offers an excellent introductory treatise on the subject.

While it is understood that the ℓ_1 regularization above performs quite well when the sparsity condition is satisfied, the question we wish to investigate here is whether a known event signature can be identified from fewer measurements in the presence of noise and interference. Along these lines, we propose to modify BP to include a weighting matrix W within the minimization objective of Equation 7.3 as follows:

$$\hat{y} = \underset{\tilde{y}}{\operatorname{argmin}} \|W\tilde{y}\|_{\ell 1} \quad \text{s.t.} \quad z = \Phi\Psi\tilde{y} \tag{7.5}$$

This weighting matrix serves to bias components of the event signature so that they are preferentially picked by the ℓ_1 minimization routine. This is done by defining W as a diagonal matrix as follows:

$$W = diag(w_1, ..., w_n), \quad \begin{array}{ll} w_j < 1 & \forall j \in \Omega \\ w_j = 1 & \forall j \notin \Omega \end{array} \tag{7.6}$$

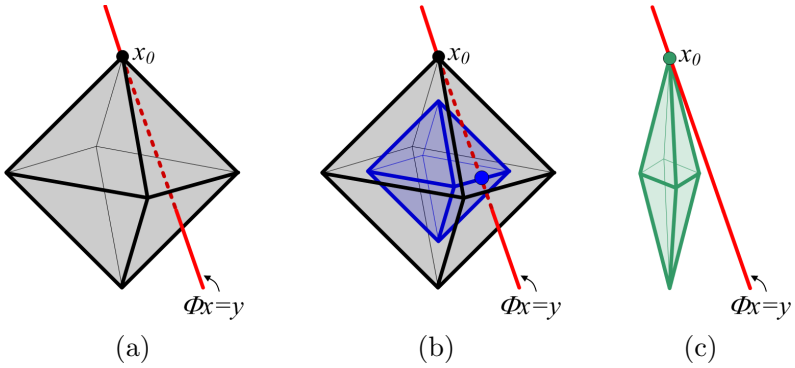

Figure 7.2: (Reproduced from [CWB08]) Weighting ℓ_1 minimization to improve sparse signal recovery. (a) Sparse signal x_0, feasible set $\Phi x = y$, and ℓ_1 ball of radius $\|x_0\|_{\ell 1}$. (b) There exists an $x \neq x_0$ for which $\|x\|_{\ell 1} < \|x_0\|_{\ell 1}$. (c) Weighted ℓ_1 ball. There exists no $x = x_0$ for which $\|Wx\|_{\ell 1} < \|Wx_0\|_{\ell 1}$.

where, Ω is the set of indices in the sparse domain where the event of interest may be present. For the detection of an acoustic event, this would correspond to the major frequency components of that signature. Note that W is constructed such that coefficients of interest have a smaller weight attached to them. The effect of this is an (artificial) reduction in the ℓ_1 norm of any solution $\tilde{y}$ that contains these coefficients, leading those solutions to be chosen over others. Because the solver has to meet the constraints $z = \Phi\Psi\tilde{y}$, it ensures that the solution is not arbitrary and that it is bounded in energy.

A geometric interpretation of the above weighting technique can be borrowed from Candes, et. al. [CWB08], which inspired the current work. They show ([CWB08], reproduced in Figure 7.2) how the weighting factor skews the previously symmetric ℓ_1 norm ball to direct it towards a preferred solution. They also show that as the weighting value $w_{j\in\Omega}$ decreases, the solutions tend to stabilize, which means that a weighting less than a certain value gives an almost identical result. This is partly because the norm ball $\|W\tilde{y}\|_{\ell 1}$ hits the same point on the polyhedra $z = \Phi\Psi\tilde{y}$ beyond a point.

In a detection scenario, it is also interesting to see what happens when the event is not present, because then the weighting violates the assumption that the signature is present at those entries. If the event is absent, noise or interference at indices corresponding to Ω will be erroneously enhanced. This means that while $w_{j\in\Omega} \to 0$ is a valid selection, when

the weights are very small the solver will enhance even small amounts of noise, resulting in false alarms (or false positives). In our empirical evaluations (details in Section 7.4), we found that the detection performance was fairly insensitive to the precise value of $w_{j\in\Omega}$ and that values between 10^{-1} and 10^{-3} gave equivalent results. The reader is referred to [KXA09], which optimizes weighting for a specific detection scenario.

7.2.1 Detection Functions

In an event detection application, it is important to consider how the event hypothesis will be decided. For example, if we assume that the hypothesis of an event being present is $\mathcal{H}_1$ and absent is $\mathcal{H}_0$, one may declare a hypothesis by computing a detection function $\mathcal{D}(\hat{y}, \Omega)$, where $\hat{y}$ is the solution to Equation 7.5 with weighting using (7.6) and Ω is a non-empty set of coefficient indices we care about. While many forms of $\mathcal{D}$ are possible, in this section we consider three forms of detection functions: precomputed threshold testing (PTT) , a proxy of the classical Likelihood Ratio Test (LRT) [MW95] and winner takes all (WTA). PTT and LRT are both defined as:

$$\mathcal{D}_{LRT/PTT}(y, \Omega) = \begin{cases} \mathcal{H}_1 & \text{if } y_j > \theta_j \ \ \forall j \in \Omega \\ \mathcal{H}_0 & \text{otherwise} \end{cases} \tag{7.7}$$

where $\theta_{j\in\Omega}$ represents a threshold for each component in y that the event signature is composed of. In classical detection theory, the thresholds are computed based on the noise power level (and we term this PTT), but we adopt a more general training based strategy to handle non-Gaussian noise sources as well as narrow-band interference (we term this learning based approach LRT). WTA is defined as:

$$\mathcal{D}_{WTA}(y, \Omega) = \begin{cases} \mathcal{H}_1 & \text{if } \mathcal{M}(y) \in \Omega \\ \mathcal{H}_0 & \text{otherwise} \end{cases} \tag{7.8}$$

where $\mathcal{M}(y)$ returns the *index* of the maximum component in y. The WTA procedure, as defined, declares the event present if the index of $\max(y)$ belongs to Ω. A more conservative version, kWTA may compare the largest k values of y. When $|\Omega| = 1$, these are equivalent.

Detection performance is measured in terms of the probability of missed detections, P_{MD} and probability of false alarms, P_{FA}, which are defined as:

$$P_{MD} = \Pr[\mathcal{D}(\hat{y}, \Omega) = \mathcal{H}_0 \mid \mathcal{H}_1] \tag{7.9}$$

$$P_{FA} = \Pr[\mathcal{D}(\hat{y}, \Omega) = \mathcal{H}_1 \mid \mathcal{H}_0] \tag{7.10}$$

We extensively evaluate the performance of event detection in both simulation and through experiments in Section 7.4, but first describe the implementation of compressive sensing on low-end sensor network platforms.

7.3 Low-Power CS Implementation

A critical aspect of implementing CS is the random projection matrix Φ (in Equation 7.5) through which the sensor node collects sample measurements. Since the matrix is constructed pseudo-randomly, the node need not communicate the complete matrix to the fusion center. Instead, if the random number generator being used and the initial seed are known, the fusion center can regenerate the matrix locally.

From [CW06], we learn that a number of random distributions may be used to develop Φ, though not all lend themselves to low-power implementations easily. The two most popular ones that have been shown to satisfy the restricted isometry property [CRT06a] are (a) when the elements of Φ are independent realizations of a Gaussian random variable, $\Phi_{ij} = \mathcal{N}(0, \frac{1}{n})$ and (b) when they are independent realizations of an equiprobable $\pm\frac{1}{\sqrt{n}}$ Bernoulli random variable.

Random projections may be computed in software by generating Φ and performing the

matrix multiplication, $z = \Phi x$. This step, however, requires the sensor node to possess x a priori, which means that it needs to sample above the Nyquist rate and store n samples in memory. Further, using the Gaussian distribution requires $\mathcal{O}(kn)$ (ideally, floating-point) multiply and add operations to compute z. Though this computational burden is relaxed when using the Bernoulli distribution, which only needs additions (the $\frac{1}{\sqrt{n}}$ scale factor can be performed post facto at the fusion center), the promised ADC rate reductions have been lost.

The device described in [KLW06] is a hardware based approach that consists of a bank of k analog front-ends, each of which performs signal multiplication with a Bernoulli random stream generated at the Nyquist rate. The result from each multiplier is integrated and sampled simultaneously at a much lower rate. While this is an attractive general purpose technique, it has two drawbacks for low-power implementation. First, the extra power consumed by the continuous analog operation is non-trivial, especially because of linearized low-noise multiplier blocks and second, strict time synchronization is required with the fusion center so that the regenerated Bernoulli stream matches that at the node.

7.3.1 Causal Randomized Sampling

A technique that avoids both these issues is randomized sampling. Sampling at uniformly distributed random instants was shown to satisfy the restricted isometry property when the sparse basis Ψ is orthogonal [CW06, RV06], and has been employed successfully in [BHF07, DDW07, DBH08]. The Φ matrix is constructed by randomly selecting one column index in each of the k rows to be set to unity, but in practice all that is required is being able to sample at arbitrary times and storing the k samples for subsequent communication. This means that the node no longer samples above the Nyquist rate nor does it perform any arithmetic operation to compute z.

This form of uniform random sampling, however, is non-causal if the random numbers are generated on-the-fly. To ensure causality, one would have to generate, sort and store

the k numbers in memory. Further, this technique has the disadvantage that two sample times may be closer together than the hardware can handle. Dang, et. al. [DBH08] circumvented this problem by applying a scaling factor before and after generating the random sample indices. To avoid the quantization effects introduced by this scaling, they apply a normally distributed jitter to the resulting sampling instants, which works acceptably well.

A simpler technique that solves these issues and is a good approximation to the uniform distribution is mentioned in Bilinskis and Mikelsons [BM92]. Let us define the k sampling instants as $t_i,\ i \in \{1, ..., k\}$. Then, the sampling instants are generated using the additive random sampling process, that is:

$$t_i = t_{i-1} + \tau_i \tag{7.11}$$

where $t_0 = 0$ and τ_i are independent realizations of a Gaussian random variable $\sim \mathcal{N}(\frac{n}{k}, \frac{r^2n^2}{k^2})$. It turns out that the PDF of $\{t_i\}$ converges to a uniform distribution rather quickly. Here, $\frac{n}{k}$ represents the desired average sampling interval and r determines the width of the bell and the resulting speed of convergence. The effect of the additive random sampling procedure is visually depicted in Figure 7.3 (adapted from [BM92]) with $r = 0.25$. The top 5 plots represent the PDFs of each $t_i,\ i \in \{1, .., 5\}$ and the bottom plot represents the PDF of realizations of all t_i, which approximates the uniform distribution as required.

We use this procedure in our implementation to generate random sampling times on-the-fly, with r fixed at 0.25. For causality and feasibility reasons, we ensure that $\tau_i > T_{ADC}$, the ADC sampling latency.

It is noteworthy to add that this causal randomized sampling procedure is as general purpose as the other techniques mentioned while reducing hardware, sampling, storage and computation requirements substantially. The only downside to using randomized sampling is that its domain basis is not incoherent with signals that are sparse in the

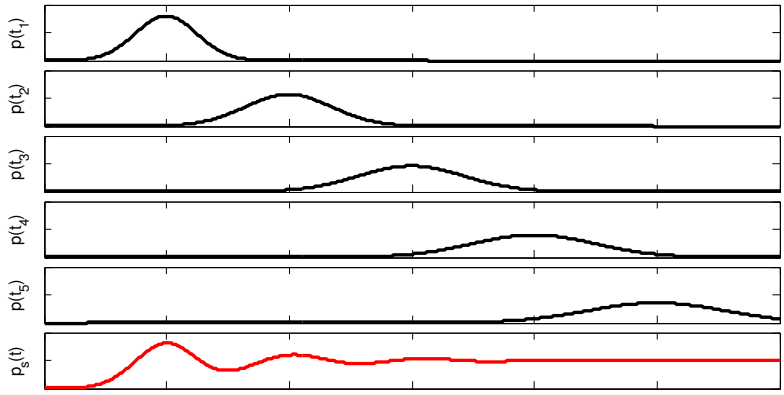

Figure 7.3: The effect of an additive Gaussian random sampling process

time domain, such as EKG signals, precluding its use from this particular sub-class of signals.

7.3.2 Quantifying Power and Duty Cycle Gains

To test our proposition in practice and quantify the gains and performance it could deliver, we implemented the solution using MicaZ sensor motes running the TinyOS operating system. We motivate the application of a CS based approach through acoustic signature detection similar to [GT] using the Fourier basis for reconstruction. Since the Fourier basis is incoherent with the time-spike basis, our randomized sampling procedure is well suited to this application.

MicaZ sensor motes contain an 8 MHz 8-bit ATMEGA128 processor with a built-in 10-bit ADC and an IEEE 802.15.4 compliant radio. They have been reported to sustain sampling rates of a few hundred Hz, limited mainly due to the absence of a DMA unit. Since detecting the signature via Fourier domain analysis on the mote itself or through sampling and collecting data wirelessly at the Nyquist rate would have been infeasible, we used a combination of empirical modeling and simulation to quantify the gains from our CS approach. In particular, we modeled four blocks, which are significant for the comparison -- random number generator (for CS), ADC, FFT processing and radio transmission.

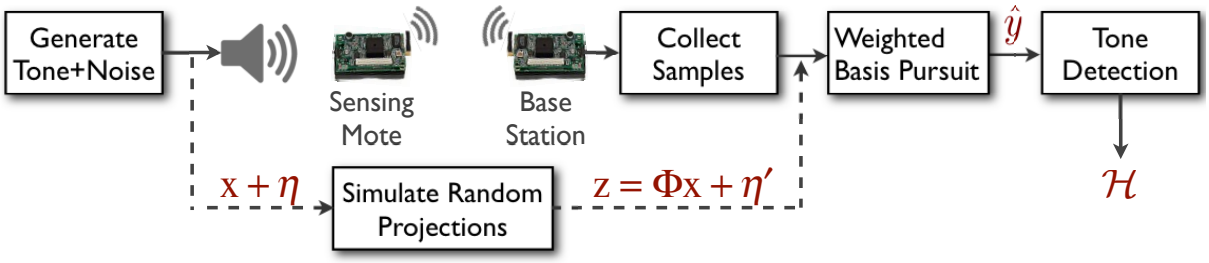

Figure 7.4: Schematic representation of detection process with MicaZ motes and in simulation

We model the energy consumption and running time of each block with simple first order linear functions that depend on the data rate flowing through them. Model parameters were extracted using a cycle and energy accurate instruction-level simulator available for the MicaZ [TLP05]. The Gaussian random variable for causal randomized sampling is computed by approximating it to an order-12 Irwin-Hall distribution using a 16-bit MLCG [Le88] based uniform random number generator. FFT processing was performed using an 1024-point implementation optimized for 16-bit operation. The FFT library routines occupy 2KB of the 4KB RAM available on the ATMEGA128.

7.4 Results

Figure 7.4 depicts a schematic representation of the compressive detection process used for evaluation. We chose to generate and detect a single frequency tone at 450 Hz for our experiments. While identifying the presence of a single frequency tone may be a trivial detection problem, we chose it as a case study for three reasons. First, it provides us with a baseline for comparison using a well known basis. Second, the solution is easily extended to event signatures sparse in other bases, with no change to the node's implementation. And finally, single frequency tones have little structure to be exploited by the detection function and thus the false alarm rates reported in Section 7.4 may be considered worst case. Results for a multi-tone case are reported in [CKZ09a].

A host machine generates the 450Hz signal and white noise at a specific SNR at a high

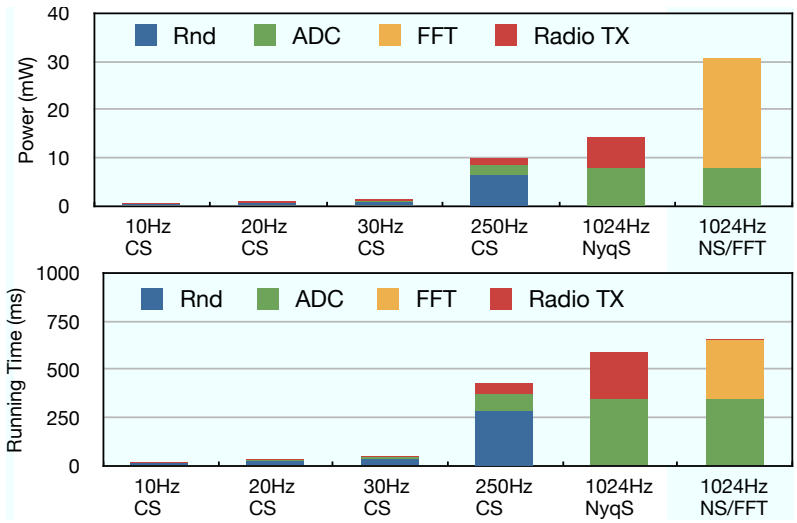

Figure 7.5: Power and Duty Cycle costs for Compressive Sensing versus Nyquist Sampling w/ local FFT.

sampling rate. This audio stream is played out over a speaker and recorded through the microphone of a *sensing* MicaZ mote using random projections as described in Section 7.3.1. One second long segments of recorded samples are then wirelessly transmitted to a *base-station* mote connected to the fusion center, which performs weighted basis pursuit to recover the signal in the frequency domain using 1024-point FFT. The FFT coefficients are fed into the detection function along with the indices Ω to produce the hypothesis decision. We also run a simulation version of the process, which emulates the recording and collection process by applying the same random projection matrix as would have been computed on the sensing mote.

Figure 7.5 reports results for the resource costs incurred by the sensing MicaZ node. While the numbers are specific to this platform, the insight from these results can be applied more generally. The top plot shows the power consumed by each block for different sampling rates. Also included are the simulated power consumption numbers when periodic sampling is applied above the Nyquist rate and when the FFT detection procedure is performed on the node itself. Performing local detection, while computationally expensive, reduces the radio transmission burden, which is especially beneficial in a multi-hop network scenario. For example, in this case, the relatively favorable results for the

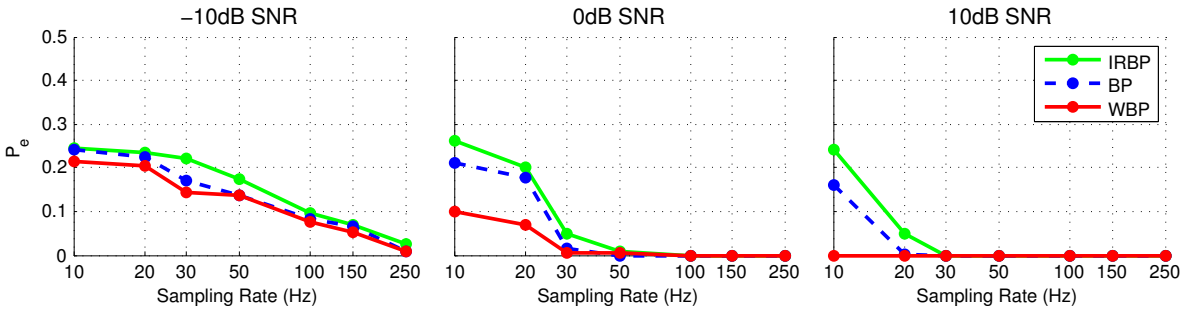

Figure 7.6: Comparing the detection performance of IRBP and BP with WBP in simulation.

radio transmission power would overshadow FFT computation cost if the base-station was further than two hops away.

For the compressive sensing cases shown left most, the biggest power consumer is the random number generator and, in particular, the MLCG implementation, which uses software emulated 32-bit arithmetic extensively. If a lower cost LFSR based implementation was used instead, the consumption would be substantially reduced at the cost of fewer unique random numbers [Le88]. In terms of energy, using 30Hz CS is over 10× more efficient than sampling and communicating at 1024Hz over a one-hop wireless link. For comparison purposes, a 250Hz CS implementation was also simulated and found to result in 30% reduction in power.

The bottom plot of Figure 7.5 illustrates the running time of each block for every one second window. This equates to the achievable duty cycle of the node, lower values for which further improve the overall energy efficiency. The ADC latency is clearly visible here as the dominant component for high rate sampling. This stems from the lack of a DMA unit, which causes the CPU to be interrupted constantly.

While Figure 7.5 emphatically demonstrates that using low rate compressive sensing can achieve long node lifetimes, Figures 7.6 and 7.7 shows that detection performance is also exceptionally good. We show results from 250-run Monte Carlo simulations (Figure 7.6) and hardware experiments (7.7) at five different average sampling rates (10, 20, 30, 50 and 100 Hz) and at three different SNRs (-10, 0 and 10 dB). The y-axis denotes the overall

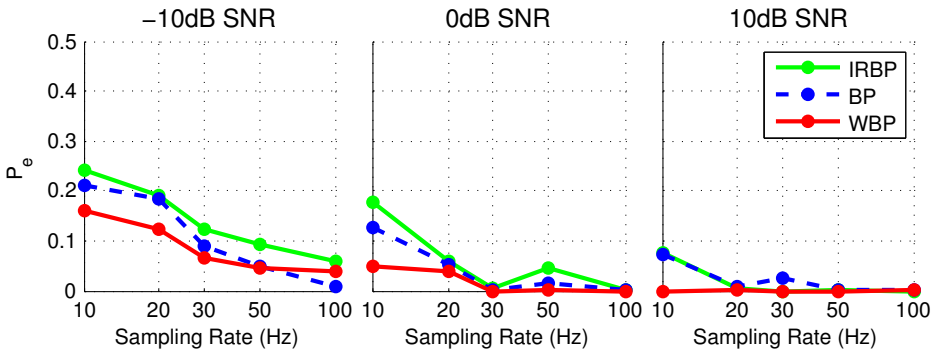

Figure 7.7: Comparing the experimental detection performance of IRBP and BP with WBP .

error probability, P_e, which is an equally weighted sum of P_{MD} and P_{FA} (Equation 7.9-7.10). All results reported here use $w_{j\in\Omega} == 0.1$. To select the thresholds θ_Ω (in Equation 7.7), we use a 10-fold cross validation training approach with Neyman-Pearson detection [MW95] setting the maximum false alarm rates to 10%.

We evaluate our biased weighting approach (WBP) against conventional basis pursuit (BP) and the iterative re-weighting technique (IRBP) described in [CWB08]. We observe some general trends right away -- increasing SNR or sampling rate reduces P_e for all three techniques. This is expected, since a higher quality signal (or the lack of it) as well as additional samples improve both the detection and rejection performance of the system. Comparing first BP and IRBP, we observe that the latter is always worse (or no better) at all SNRs and sampling rates. This seems counter-intuitive at first because IRBP has been shown to perform especially well when the signal is non-sparse [CWB08]. The reason for this poor detection performance at low sampling rates arises from the way the algorithm applies iterative weighting. Fundamentally, IRBP assumes that the solution from a previous iteration is a good one and strengthens that solution in subsequent iterations. Thus, if the solution in the first iteration (which is unbiased) is a bad one, IRBP gets caught in a local trap, never attempting other possibilities. Our implementation of IRBP assumes a zero valued initial condition as suggested in [CWB08]. This approach turns out worse than conventional BP because the iterative strengthening leads to a

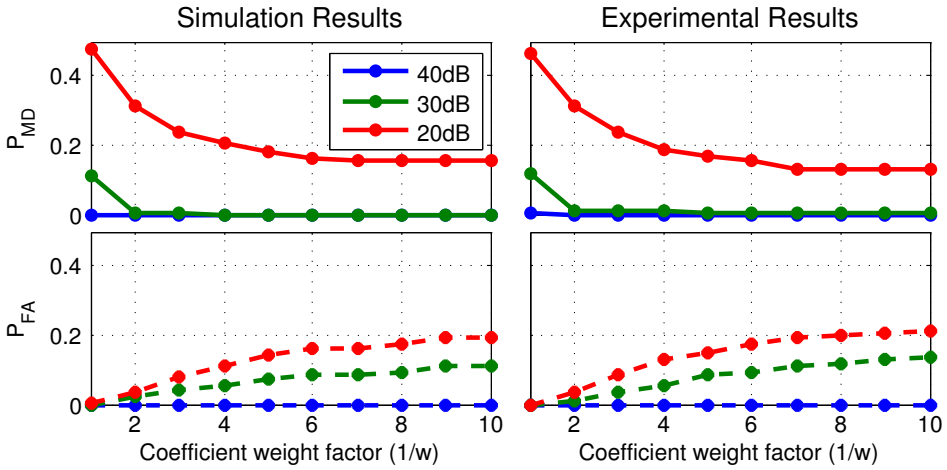

Figure 7.8: P_{MD} and P_{FA} for PTT for various SNR at 30 Hz sampling.

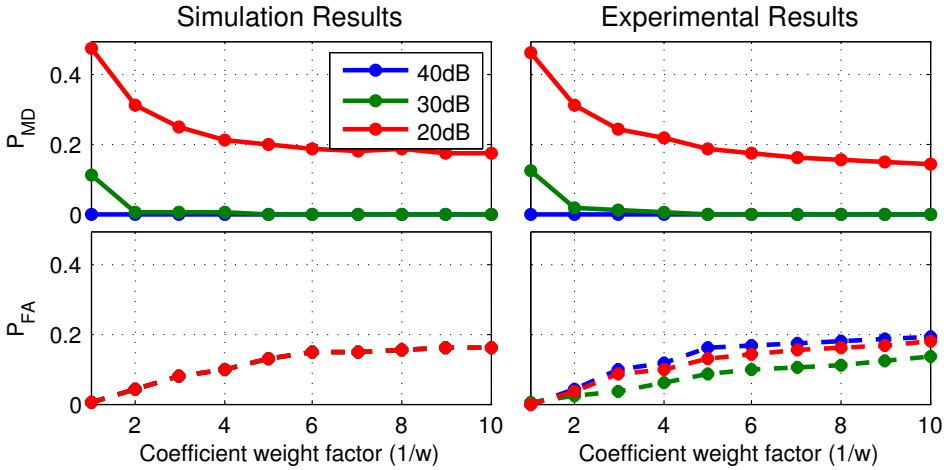

Figure 7.9: P_{MD} and P_{FA} for WTA for various SNR at 30 Hz sampling.

large number of false alarms. With WBP, only those indices that form part of the event signature are biased. Thus, false alarms occur only in the unlikely scenario that significant noise or interference energy is present at those indices.

7.4.1 Using SNR Dependent Thresholds

Instead of using training, we could have used a fixed SNR dependent threshold value (PTT), as is commonly done in likelihood ratio testing and this has an interesting effect as the parameter $w_{j\in\Omega}$ is varied. It can be shown that P_{MD} is a monotonically *non-*

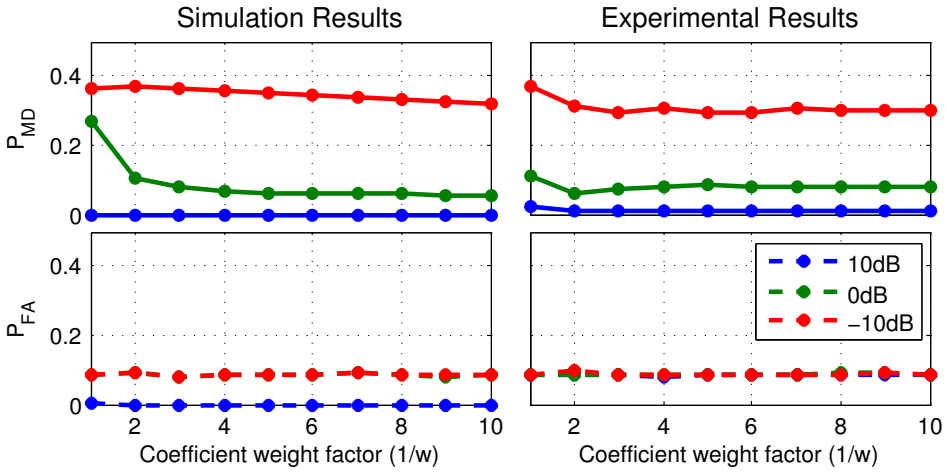

Figure 7.10: P_{MD} and P_{FA} for PTT for various SNR at 20 Hz sampling.

decreasing function of w for fixed SNR and sampling rate and P_{FA} is a monotonically *non-increasing* function of w. This conclusion is intuitive, owing to the fact that as w reduces, so does the ℓ_1 norm, promoting those indices in the solution (even if the signal was not present). Figures 7.8 and 7.9 illustrate results at 30 Hz across varying coefficient weights. Included in this set of plots are results from simulated runs of the same experiments. Note that at high SNR, false alarm rates are negligible even at $w = 10$ for PTT, but are quite high for WTA. However, when SNR is low, WTA outperforms PTT slightly. We believe this is because WTA picks the maximum component in the FFT coefficients and even in low SNR regimes, the signal component is inclined to stand out.

It must be mentioned here that the improved performance of PTT has a cost associated with it. Selecting the right thresholds θ_Ω is non-trivial in cases where the signal is not completely captured within the samples being processed. Since the reconstruction is performed on a block of received data (in our case, we used 1 sec worth of samples) each time, the event signature may not be aligned with the block. In order to avoid this complexity and to demonstrate the asymptotic performance of the system, we applied a preprocessing step for experimental data to recover the correct alignment by padding the event signature with silence.

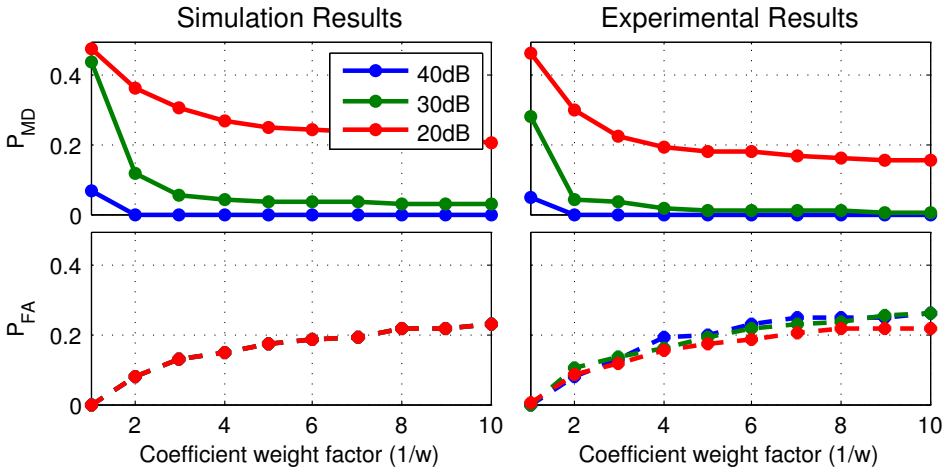

Figure 7.11: P_{MD} and P_{FA} for WTA for various SNR at 20 Hz sampling.

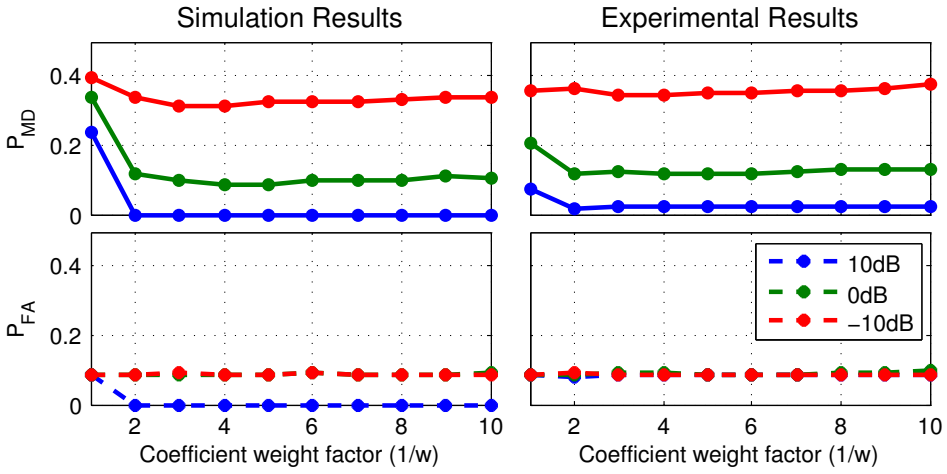

Figure 7.12: P_{MD} and P_{FA} for PTT for various SNR at 10 Hz sampling.

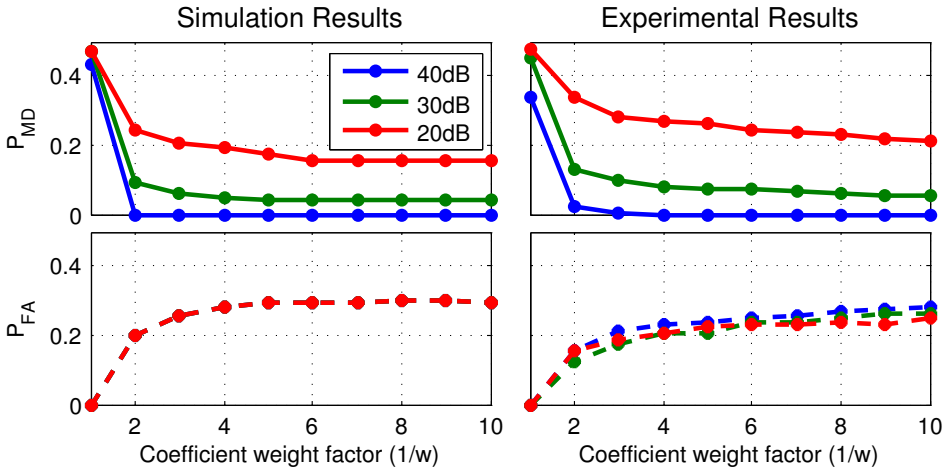

Figure 7.13: P_{MD} and P_{FA} for WTA for various SNR at 10 Hz sampling.

Another aspect worth noting is that with WTA, all SNRs result in the same P_{FA} performance, in simulation results. This is not a discrepancy. As the SNR increases, the chances of detecting the signal are higher if it is present (resulting in lower P_{MD}). But, when there is only noise (which happens for false alarms), SNR is effectively 0, so the same noise component gets picked every time regardless of SNR. Note that this result is also a side effect of maintaining the same random noise seeds across the Monte Carlo simulation runs. This behavior is not present in the experimental results though the random noise generated is be the same because the noise in the recorded samples is affected by multiple factors, including ambient and circuit noise.

Figures 7.10 - 7.13 show the performance of the system for average sampling rates of 20 Hz and 10 Hz respectively. We observe that both P_{MD} and P_{FA} performance worsens as the sampling rate is reduced. This is because the feasible solution space that conforms to the polyhedra $x = \Phi\Psi\tilde{y}$ expands to include many points that may be classified wrongly. Further details of this are included in [CKZ09a].

In summary, Figure 7.7 illustrates results for our weighted BP approach, which consistently outperforms the former two recovery techniques for detection. The performance improvement is more obvious at lower sampling rates and higher SNRs. There are slight discrepancies for experimental results that, we believe, are an artifact of the inevitable en-

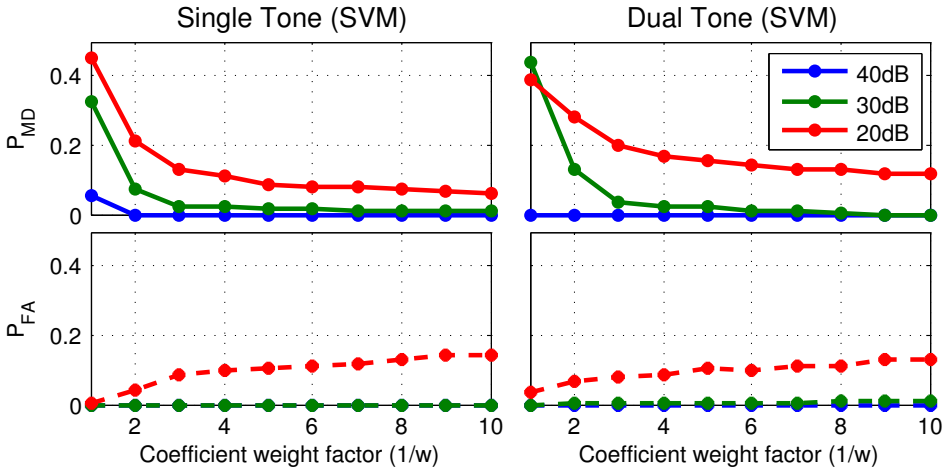

Figure 7.14: P_{MD} and P_{FA} detecting a dual tone signal using an SVM classifier.

Relative Power Gains → SNR	-10dB	0dB	10dB
Over 1024Hz sample-and-send	30%	90%	96%
Over 1024Hz sample-FFT-detect	67%	95%	98%

Table 7.1: Relative power consumption gains using WBP CS with comparable detection performance.

vironmental differences between runs. Notice that at 0dB and 10dB SNRs, the detection performance is near-perfect with 30Hz and 10Hz respectively. For the poor -10dB SNR environment, the sampling rate has to be elevated considerably to extract the same level of performance. A snapshot of the relative power gains achieved by compressive sensing with weighted basis pursuit for detection performance comparable to Nyquist sampling is listed in Table 7.1.

7.4.2 Event Signatures with Structure

In Figure 7.14, we illustrate results that test our conjecture that signals with structure may be detected more easily, with possibly fewer false alarms. A trivial signal structure results from two frequency components, so we add an equal amplitude tone at 150 Hz to the original event signature and repeat the experiments. To ensure that we recognize and exploit the structure correctly, we used a support vector machine (SVM [CL01]) classifier

for detection rather than PTT or WTA.

Using this classifier required model selection, which was performed through randomized 10-fold cross validation and training, which was conducted with data from 50% of the simulation runs. The features used for classification were the magnitudes and angles of the complex FFT coefficients. To establish a fair comparison, the same detection procedure was also performed on the single frequency tone signature. Both events were randomly sampled at 30 Hz.

Surprisingly, we observe little or no improvement in the dual tone detection scenario. In fact, we observe some deterioration in P_{MD} performance. Upon inspection, we understand two reasons for this behavior. Firstly, since SNR is computed using the total signal power, which is now shared between two frequency components, individual indices constitute a lower power contribution against the same noise power. And secondly, by introducing an extra frequency tone, we have changed the sparsity of the event signature. From [CRT06a], we know that the number of measurements required for reconstruction is proportional to the sparsity of the signal. We surmise that it is the simple structure in the signature that offsets the deterioration resulting from both these issues.

7.4.3 Events in Narrowband Interference

A final set of results in Figure 7.15 show that WBP is comparable to conventional BP in the presence of narrow-band interference. To emulate interference, we generate a high amplitude tone at a randomly selected frequency such that the signal-to-interference-plus-noise ratio (SINR) is between -10dB and -30dB. The noise power was maintained at 0dB.

7.5 Conclusion

We have presented a novel modification to the basis pursuit reconstruction procedure for known-signature event detection from sparse incoherent measurements. We show through simulations and an implementation on MicaZ sensor nodes that this strategy is not only

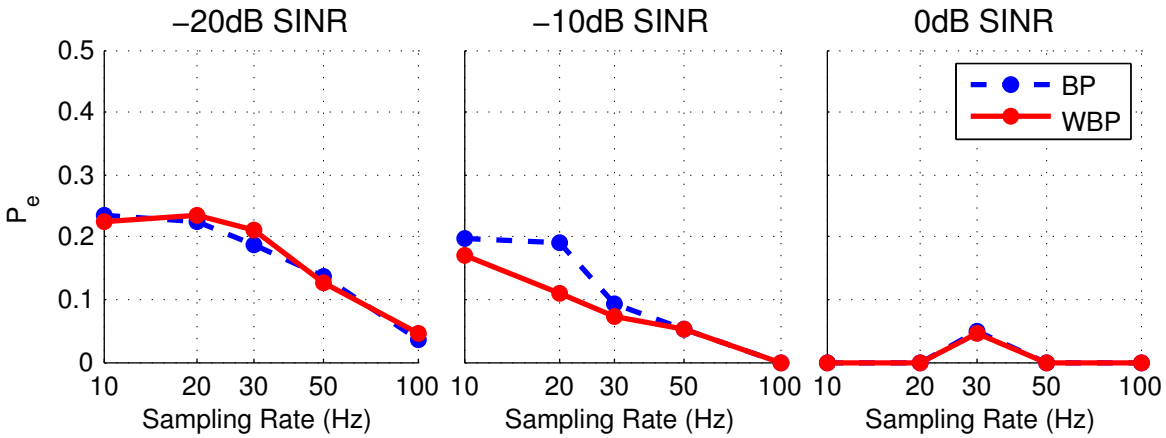

Figure 7.15: Detection performance of BP and WBP in narrow-band interference with 0dB noise power.

feasible at rates 30× below the Nyquist requirement but that it delivers comparable detection performance with up to 10× increased energy efficiency. Our empirical study also shows that the computational complexity of good random number generation is non-trivial for these low-power embedded devices.

CHAPTER 8

Conclusion

We exemplified the principle of prudent sampling in a number of different ways that cover each of the sensing stages: from sensor interface to inference engine. We take this opportunity to recapitulate our key findings, emphasize some caveats and pinpoint some open questions that we hope to explore in future research.

Filters That Remember

In the first part of this dissertation, we focused on the interface between the sensed phenomena and the processing logic. For a physiological monitoring system, we found that the energy consumption of the analog front end, the circuit that conditions the raw signal from a transducer, accounts for a large proportion of overall system energy. Since the circuit is only needed part of the time when the signal is actually sampled or during a period of known activity, one considers whether the circuit can be duty cycled the same way as its digital counterparts. Duty cycling analog front ends comes at a cost, however, because they usually have filters with long settling times. In Chapter 2, we proposed the Filters That Remember (FTR) approach to reduce this settling delay and showed its performance on an off-the-shelf electrocardiography monitor.

A key limitation of the FTR approach, when applied to activation only during specific events is that the system would miss asynchronous events and anomalies. FTR is better suited at acquiring (quasi-)periodic signals and if Inequality 2.10 is satisfied, turning off

the AFE between sample acquisitions. Unfortunately, Inequality 2.10 is not satisfied for the ShimmerFTR design with a 256Hz sampling rate due to charge injection effects and because of the cost associated with re-energizing the circuit each time between samples. That said, a 28% duty cycle (active for 210ms at 80bpm) is sufficient to capture most (but not all) anomalies. FTR is also unsuitable when the switch series resistance would affect the filter response unacceptably or when the required duty cycle is too high -- when acquiring long S-T segments or P-wave segments, for example. Furthermore, a fundamental limitation of FTR is that the filter would be oblivious to any changes in the signal during the off time. This has implications for the filter response if the filter characteristics mandated that the current value of the signal be dependent on the signal's recent historical changes. This holds, for example, with anti-aliasing filters. Applying FTR, therefore, between samples was not found to be a valid strategy when employed on anti-alias filters. As others have noted, there is an additional processing burden associated with this form of duty cycling to predict the right instant to wake up, but we believe that our switch based filter architecture has shifted the scales in favor of AFE duty cycling.

The FTR technique is not exclusively relevant to the ambulatory ECG monitoring circuits that we analyzed. The analog front end is beginning to influence the power budget for other biophysical sensors, like EEG [HKC09] and accelerometers as well. We argue that whenever an application requires only a part of the signal waveform -- to compute strides with accelerometers or spikes in EEG -- duty cycling the analog front end will lead to substantial benefit. As an example, consider the ADXL335, a popular low-cost low-power 3-axis accelerometer. The device is rated at 350μA in active mode and provides analog outputs that must be filtered for noise reduction. An application such as stride analysis, which requires information in the 2Hz - 5Hz band [BBS08], may opt to sample the signal at 10Hz, a sample every 100ms. However, this requires a filter capacitance of 1μF resulting in a practical wake up delay of 160ms (see note 7 in Table 1 in [Ana10]) precluding the use of duty cycling. With FTR, even with a conservative estimate of 10ms for start up, the accelerometer could be running at a tenth of its power.

CapMux

Chapter 3 introduced the concept of compressed sensing (CS), a technique that is used extensively in later chapters too. CS offers a brand new perspective on sampling that obviates the need for explicit compression post sampling. Chapter 3 explores how one could implement compressed sensing right in the analog domain. The advantage of doing so is a reduction in the ADC sampling energy and energy required for computing explicit matrix multiplications in digital hardware. The novel architecture, which we call CapMux, uses a time multiplexed approach to computing equivalent matrix multiplications in micro-power analog circuits. We showed a proof-of-concept design of a 16-channel CapMux board that is capable of compressing sparse and compressible signals with a greater than 30 dB SNR at less than 20μA quiescent current. Other state-of-the-art architectures cost at least 3 orders of magnitude more in terms of current consumption and cannot be scaled down easily.

The main performance parameter of the CapMux system is the bandwidth of signals it can correctly acquire. The signal bandwidth is in turn dependent on the chipping rate, f_c but the maximum achievable chipping rate in CapMux depends on the total integration time for one round (one column of the sensing matrix). With sparse binary sampling matrices, the chipping rate is given by $f_c = 1/d\tau$. Sampling density d and per channel integration time τ should, therefore, be kept as small as possible for high f_c. Integration time is harder to reduce because it is governed by the speed (slew rate) of the opamp, the leakage characteristics of the switches and capacitors, and the overall noise of the circuit. A low τ would also mean a reduced integrated charge and a lower signal-to-noise ratio (SNR), unless the capacitors are small too. But, one must ensure that total switch parasitic capacitance ($\sim$2-5pF per switch) should be a tiny fraction of integration capacitance.

In head-to-head tests with low power micro-controllers, it turns out that CapMux is only marginally better at low sampling rates because of the high integration times required

($>30\mu$s) due to the large integration capacitors we needed to use. When CapMux is integrated in a monolithic design, much smaller integration capacitors can be used and much tighter noise performance can be be achieved [Tex09]. Another aspect worth noting is that with advances in digital logic, the energy required for the digital logic for CapMux switching is not substantially lesser than the energy consumed for the explicit matrix multiplication. Effectively the difference between CapMux and a conventional sample-and-CS routine boils down to how much more the ADC costs relative to the cost of the CapMux analog hardware.

Scaling the CapMux architecture to a larger number of channels allows for accurate reconstruction of signals of higher dimensionality with little to no increase in the quiescent current of the analog circuitry, due to time-multiplexed access of a single integration subsystem. Towards demonstrating the efficacy of the system as a scalable solution, we presented empirical results for a scaled version of the current CapMux circuit up to 64 channels. The 64 channel simulations suggest an almost six-fold increase in the ability to recover sparse signals without decreasing the sampling rate of the compressor.

Union of Supports Compressed Sensing

In Chapter 4, we turn to the processing stage within the context of a wireless neural recorder. It is interesting to observe that the neuronal action potentials that scientists seek to capture are slightly different from each other due to physiological reasons. Neuroscientists use this diversity in action potentials to separate them into clusters so that they can associate each cluster with a specific neuron and its corresponding function. We approached this problem from the other side of the coin, noting that there are sufficient similarities in the shape of the spike to learn and exploit a model. We instantiated this in the context of a compressed sensing system that uses knowledge of the support of previously recovered spikes to improve the signal to noise ratio of subsequent spikes. Alternatively, this could be used to reduce the number of compressed measurements for a particular SNR, improving the overall compression ratio and system energy footprint.

In particular, we illustrated how action potentials are sparse in the DWT domain and how their supports overlap in the DWT domain. We proposed a union of supports technique for learning the support over time and showed that it provides an average SNDR improvement of 6.7 dB and a maximum SNDR improvement of 17 dB compared to conventional basis pursuit reconstruction. We also showed that CS with union of supports can be used to provide a 2× reduction in the output data rate and in the system power consumption compared to transmission of raw action potentials. At the same time, sending 24 CS measurements provides a 20-dB SNDR for the reconstructed signal, which corresponds to a 95% classification accuracy for the spikes. A 3-times reduction in system power can be obtained if 12 CS measurements corresponding to a 10-dB SNDR and an 85% classification accuracy are transmitted. Transmitting CS samples is the most power-efficient way to reduce the data rate for spike firing rates below 80 Hz and is the only solution that allows access to individual recorded action potentials.

The learned union of supports methodology can be employed whenever there is temporal similarity in the signal being acquired. Fortunately, this holds true for many physiological signals because of various rhythms the body is tuned to. A good example is the electrocardiography (ECG) application targeted in Chapter 2. ECG signals have high temporal similarity in terms of the shape of the waveform but the ECG pulses themselves are only quasi-synchronous because of changing heart rate. We believe that CS with a learned union of supports will help not only with compressing ECG signals but can also help identify anomalies because it can quickly discover when a pulse does not meet an expected model.

Interference and QoI-aware Sensing

The following two chapters focus on transport mechanisms used in the sensing chain. These chapters have remained application agnostic due to the nature of the transport stage. In Chapter 5, we undertake the ambitious goal of improving network efficiency by incorporating everything from sampling rates to application level objectives, something

we term the Quality of Information to guide rate control in congested networks. We formulated the problem in terms of interference constraints set by the node topologies and coupled that with knowledge of how inferences are affected by the data rate allotted to each node in the network.

We illustrated how meticulous attention to application relevant objectives can lead to higher networking performance and reduced communications cost by optimizing sampling rates. We attempted to translate heuristics that lead to mechanisms such as foveated sensing into formal objectives using a QoI metric. This not only allowed us to analyze and incline the problem more rigorously, but in some instances, uncovered reasons why particular heuristics work exceptionally. The work also produced a formulation that allows us to tweak a multitude of variables simultaneously -- noise variance, sensor reliability, channel conditions, etc. We also showed a practical greedy rate control mechanism that achieves close to optimal performance.

Our results demonstrate the benefit of using prior information of event location on the probability of error. On an example network, using the QoI objective reduced the P_e by $3\times$ while incurring marginal cost from explicit feedback. Moreover, the effects of inaccuracy in the estimate of event location are examined and found to be contained with high probability. We show interesting results for a larger network that illustrate why QoI is especially important as networks scale. In particular, careful rate selection shifts the bottleneck link away from the sink, allowing the ''best'' nodes to participate more effectively. A fortunate side effect of this is that it relieves nodes closer to the sink, improving mean network lifetime.

We side stepped a plethora of significant practical considerations in order to analyze the system fully. Here, we suggest ways in which some of them could be handled. We ground the discussion in the context of a K-site event detection application. In binary detection, the event is either present or absent and the rate allocation remains static in steady state. In the K-site version, at most one event may occur but its site is unknown. Thus, rates may be continually and dynamically altered so as to minimize P_e (eg. foveal

sensing). In a multi-hop network, it can take up to $2D\Delta t$ time for feedback from the fusion center to take effect, where D is the network diameter. Therefore, to improve network response time, Δt must remain as short as possible. However, Δt must also be long enough to accommodate transmissions from nodes that share the medium. Additionally, a setting for Δt must also consider the minimum interval between two events. The network must be allowed enough time to recover from rate transients.

Further, since feedback is also sent multi-hop, updated rate vectors come into effect in a rippled manner. This means that our meticulously derived constraints could be violated at some nodes (and may be too conservative at others). To avoid this, the control algorithm has to ''plan ahead'' to ensure that rate vector transients do not cause havoc. This can be achieved only because the algorithm can compute the congestion effect of prior allocations in future epochs. Though past allocations cannot be revoked, new ones can accommodate for the rippling effect.

Our abstracted network model assumed a well-behaved link model. Effective link capacity, however, shows high variability and predicting it accurately at the fusion center is impossible. One way to address this is to split each rate into an ''always fill'' and ''fill if possible'' part. Metrics such as ETX could provide link quality statistics to compute these fractions and nodes could exploit current local knowledge to fill each one in.

A critical drawback of the proposed greedy rate control algorithm is its inability to handle wireless link errors. We believe this could be handled by the fusion center performing the probe and sense routine twice for every node to estimate the fraction of rate loss due to congestion and link errors independently. We must mention that our current problem formulation does not consider link layer ACKs, multiple path routing or multiple sink nodes. It is conceivable, though, that each of these can be incorporated within the same framework.

Compressive Oversampling for Robust Transport

In Chapter 6, we introduced the idea that if a signal is compressible, one could use compressed sensing not only to improve the energy efficiency by including compression within the sampling stage, but that transport over erroneous wireless channels could also be made robust through a proportionate oversampling during CS. We propose using compressive sensing for handling data loss from erasure channels by viewing it as a low encoding-cost, proactive, erasure correction scheme. We showed that CS erasure coding is efficient when the channel is memoryless and employed the RIP to illustrate, that even extreme stochasticity in losses can be handled cheaply and effectively. We showed that for the Fourier random sampling scheme, oversampling is much less expensive than competing erasure coding methods and performs just as well. This makes it an attractive choice for low-power embedded sensing where forward erasure correction is needed.

We described CS erasure coding (CSEC) for handling erasures in a channel, but CSEC can be extended to correct for errors in the sensor transduction process too. This means that a controlled amount of sensor noise can be cleaned from the acquired measurements during the decompression process. It is achieved by using Basis Pursuit De-noising [CDS98], which changes the equality constraint in Equation (6.2) to an inequality to account for variations due to noise. Note, however, that since CSEC utilizes features of the physical phenomenon and operates on the acquired signal, and not on the modulated symbols transmitted through the wireless channel, CSEC is not useful for correcting symbol errors at a communication receiver. A better approach to tackling the latter using ℓ_1 minimization techniques is discussed by Candes and Tao in [CT05].

Note that our evaluation studies assumed that measurements are streamed to the receiver as they are acquired. If one packetizes the measurements for transmission, in a memoryless channel, the sample losses will no longer be independent and instead show high burstiness. Despite its advantages, CS erasure coding is not intended as a replacement for traditional physical layer channel codes. It is neither as general-purpose (i.e.

it cannot be used for arbitrary non-sparse data), nor is the decoding as computationally efficient (yet). Instead, CSEC should be considered as a coding strategy that is applied at the application layer, where it utilizes knowledge of signal characteristics for better performance. In this regard, it is the reduced encoding cost that makes CSEC especially attractive for low-power physiological sensing.

Compressive Detection

In the final chapter, we target the inference engine and show in particular how knowing the structure of an event to be detected can improve energy efficiency through smarter sampling. In particular, this chapter also uses compressed sensing and illustrates its versatility in handling an event detection application. While many CS mechanisms have focused on signal reconstruction, we found that the number of samples needed to reliably detect features in the signal, even in a noisy and interference prone environment, can be considerably lower if full CS recovery is not required. We presented a novel modification to the conventional basis pursuit reconstruction procedure for known-signature event detection from sparse incoherent measurements. We show through simulations and an implementation on MicaZ sensor nodes that this strategy is not only feasible at rates 30× below the Nyquist requirement but that it delivers comparable detection performance with up to 10× increased energy efficiency. Our empirical study also shows that the computational complexity of good random number generation is non-trivial for these low-power embedded devices.

We used a version of randomized sampling that is both causal and practical. It involves simply using a Gaussian distributed sampling period to determine the next sampling instant. It is noteworthy to add that this causal randomized sampling procedure is as general purpose as other techniques while reducing hardware, sampling, storage and computation requirements substantially. The only downside to using randomized sampling is that its domain basis is not incoherent with signals that are sparse in the time domain, such as EKG signals, precluding its use from this particular sub-class of signals.

The key tunable parameters with the compressive detection approach are identifying the correct structure in the event and setting a good weighting matrix that balances the false positive and false negative rates. In a detection scenario, it is also interesting to see what happens when the event is not present, because then the weighting violates the assumption that the signature is present at those entries. If the event is absent, noise or interference at indices corresponding to Ω will be erroneously enhanced. This means that while $w_{j\in\Omega} \to 0$ is a valid selection, when the weights are very small the solver will enhance even small amounts of noise, resulting in false alarms (or false positives). In Chapter 4, we had a similar issue, which was resolved through temporal similarity. In the event detection scenario as well, if one assumes that training data is available, a simpler implementation ensues, both to set the weights and to choose the detection thresholds.

Sensing is an important aspect of physiological monitoring and the other applications we have described here, but is also imperative for any closed-loop system, whether in the natural or artificial world. The main point of this work is to show how the sensing process, which begins at sampling, can be made more efficient if one knows before hand some feature of the signal or the purpose for acquiring the signal or both. While the Whittaker-Shannon-Nyquist theorem provides a fundamental result about the sampling rate for synchronous acquisition, it does not enable any additional optimizations that could come with extra knowledge of the signal or the application. In this work, we have explored how one may be able to use this extra knowledge and put it to use for various applications and scenarios. We find that, indeed, by taking into consideration the entire information chain, the process of sensing can be made much smarter.

APPENDIX A

CapMux Schematic

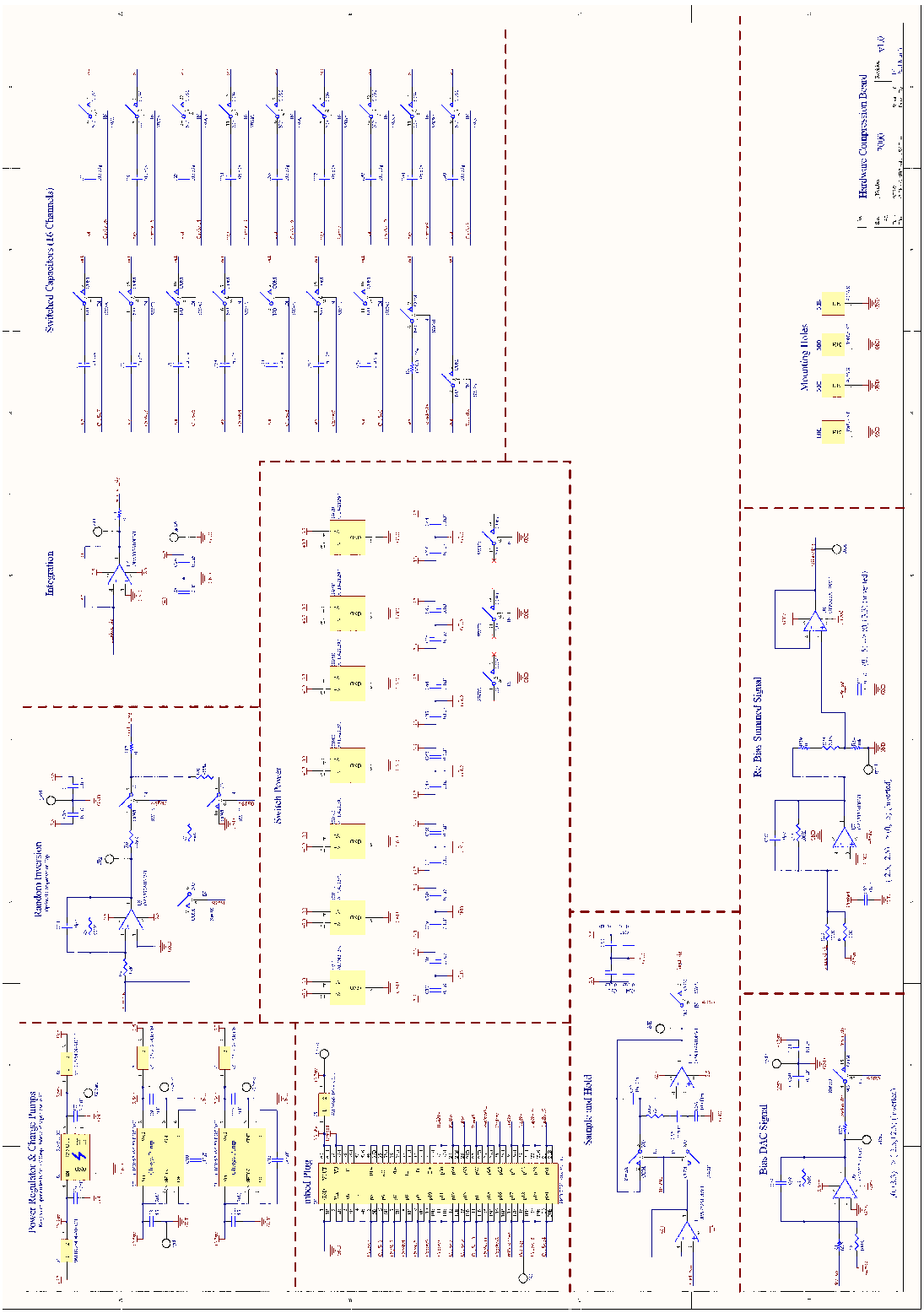

Figure A.1: Schematic of the CapMux Compressed Sensing Sampling Hardware

References

[ACR09] A. Abdulghani, A. Casson, and E. Rodriguez-Villegas. ''Quantifying the Feasibility of Compressive Sensing in Portable Electroencephalography Systems.'' *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, pp. 319--328, 2009.

[ACT] A.M. Ali, T.C. Collier, C.E. Taylor, D.T. Blumstein, and L. Girod. ''An empirical study of collaborative acoustic source localization.'' In *IPSN'07*.

[AGN] Michael Allen, Lewis Girod, Ryan Newton, Samuel Madden, Daniel T. Blumstein, and Deborah Estrin. ''VoxNet: An Interactive, Rapidly-Deployable Acoustic Monitoring Platform.'' *IPSN08*.

[Ana10] Analog Devices. *ADXL335: Small, Low Power, 3-Axis 3g Accelerometer*, 2010.

[AND06] Kanak Agarwal, Kevin Nowka, Harmander Deogun, and Dennis Sylvester. ''Power Gating with Multiple Sleep Modes.'' *Quality Electronic Design, International Symposium on*, **0**:633--637, 2006.

[AO09] M. Aghagolzadeh and K. Oweiss. ''Compressed and distributed sensing of neuronal activity for real time spike train decoding.'' *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **17**(2):116--127, 2009.

[APM05] I.F. Akyildiz, D. Pompili, and T. Melodia. ''Underwater acoustic sensor networks: research challenges.'' *Ad Hoc Networks*, 2005.

[AR08] M.S. Asif and J. Romberg. ''Streaming Measurements in Compressive Sensing: l Filtering.'' In *42nd Asilomar conference on Signals, Systems and Computers*, 2008.

[AT08] A. Anta and P. Tabuada. ''To sample or not to sample: Self-triggered control for nonlinear systems.'' *Arxiv preprint arXiv:0806.0709*, 2008.

[Avi07] S. Aviyente. ''Compressed sensing framework for EEG Compression.'' In *Statistical Signal Processing, 2007. SSP'07. IEEE/SP 14th Workshop on*, pp. 181--184. IEEE, 2007.

[BBS08] S. Bamberg, A.Y. Benbasat, D.M. Scarborough, D.E. Krebs, and J.A. Paradiso. ''Gait analysis using a shoe-integrated wireless sensor system.'' *Information Technology in Biomedicine, IEEE Transactions on*, **12**(4), 2008.

[BCJ98] AS Bhushan, F. Coppinger, and B. Jalali. ''Time-stretched analogue-to-digital conversion.'' *Electronics Letters*, **34**(9):839--841, 1998.

[BDD08] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. ''A simple proof of the restricted isometry property for random matrices.'' *Constructive Approximation*, 2008.

[BG09] P.K. Baheti and H. Garudadri. ''An ultra low power pulse oximeter sensor based on compressed sensing.'' In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pp. 144--148. IEEE, 2009.

[BHF07] F.A. Boyle, J. Haupt, G.L. Fudge, and C.C.A. Yeh. ''Detecting Signal Structure from Randomly-Sampled Data.'' In *Statistical Signal Processing*, 2007.

[BISa] Guillermo Barrenetxea, Francois Ingelrest, Gunnar Schaefer, and Martin Vetterli. ''The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments.'' In *SenSys'08*.

[Bisb] C. Bisdikian. ''On sensor sampling and quality of information: A starting point.'' In *Percom'07*.

[BM92] I. Bilinskis and AK Mikelson. *Randomized Signal Processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[BP72] LT Bruton and RT Pederson. ''Time-multiplexed active filters.'' *Solid-State Circuits, IEEE Journal of*, **7**(3):259--265, 1972.

[Bra00] P.C. Brath. ''Atlas of Cardiovascular Monitoring.'' *Anesthesiology*, **93**(1):312, 2000.

[Can08] E.J. Candès. ''The restricted isometry property and its implications for compressed sensing.'' *Comptes rendus-Mathématique*, 2008.

[CCC08] Haksoo Choi, Sukwon Choi, and Hojung Cha. ''Structural Health Monitoring system based on strain gauge enabled wireless sensor nodes.'' *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, 2008.

[CCG09] S. Craciun, D. Cheney, K. Gugel, J.C. Sanchez, and J.C. Principe. ''Compression of neural signals using discriminative coding for wireless applications.'' In *Neural Engineering, 2009. NER'09. 4th International IEEE/EMBS Conference on*, pp. 629--632. IEEE, 2009.

[CCG11] S. Craciun, D. Cheney, K. Gugel, J.C. Sanchez, and J.C. Principe. ''Wireless Transmission of Neural Signals Using Entropy and Mutual Information Compression.'' *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **19**(1):35 --44, 2011.

[CCS10] F. Chen, A.P. Chandrakasan, and V. Stojanovic. ''A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors.'' In

Custom Integrated Circuits Conference (CICC), 2010 IEEE, pp. 1--4. IEEE, 2010.

[CCZ10] Z. Charbiwala, S. Chakraborty, S. Zahedi, Y. Kim, M.B. Srivastava, T. He, and C. Bisdikian. ''Compressive oversampling for robust data transmission in sensor networks.'' In *INFOCOM, 2010 Proceedings IEEE*, pp. 1--9. IEEE, 2010.

[CDS98] S.S. Chen, D.L. Donoho, and M.A. Saunders. ''Atomic Decomposition by Basis Pursuit.'' *SIAM Journal on Scientific Computing*, 1998.

[CFS11] Z. Charbiwala, J. Friedman, M.B. Srivastava, and B. Kuris. ''Filters That Remember: Duty Cycling Analog Circuits for Long Term Medical Monitoring.'' 2011.

[CFX07] S. Chen, Y. Fang, and Y. Xia. ''Lexicographic Maxmin Fairness for Data Collection in Wireless Sensor Networks.'' *IEEE Transactions on Mobile Computing*, 2007.

[CH] Antonio Carzaniga and Cyrus P. Hall. ''Content-Based Communication: a Research Agenda.'' In *SEM '06*.

[Cha11] Zainul M Charbiwala. ''Nordic Radio nRF24L01+ Power Characterization.'' *NESL Tech Report (TR-UCLA-NESL-201107-02)*, 2011.

[CKG11] Z. Charbiwala, V. Karkare, S. Gibson, D. Markovic, and M.B. Srivastava. ''Compressive Sensing of Neural Action Potentials Using a Learned Union of Supports.'' 2011.

[CKZ09a] Zainul Charbiwala, Younghun Kim, Sadaf Zahedi, Rahul Balani, and Mani B. Srivastava. ''Weighted ℓ_1 Minimization for Event Detection in Sensor Networks.'' *NESL Tech Report, http://nesl.ee.ucla.edu/document/show/299*, 2009.

[CKZ09b] Zainul M Charbiwala, Younghun Kim, Sadaf Zahedi, Jonathan Friedman, and Mani B Srivastava. ''Energy Efficient Sampling for Event Detection in Wireless Sensor Networks.'' *International Symposium on Low Power Electronics and Design (ISLPED)*, 2009.

[CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[CMB08] W.L. Chan, M.L. Moravec, R.G. Baraniuk, and D.M. Mittleman. ''Terahertz imaging with compressed sensing and phase retrieval.'' *Optics Letters*, 2008.

[CMS12] Zainul M Charbiwala, Paul Martin, and Mani B Srivastava. ''CapMux: A Scalable Analog Front End for Low Power Compressed Sensing.'' *Internation Green Computing Conference, in review*, 2012.

[CPG] K. Chintalapudi, J. Paek, O. Gnawali, T.S. Fu, K. Dantu, J. Caffrey, R. Govindan, E. Johnson, and S. Masri. ''Structural damage detection and localization using NETSHM.'' In *IPSN'06*.

[CR07] E. Candes and J. Romberg. ''Sparsity and incoherence in compressive sampling.'' *Inverse Problems*, 2007.

[CRT06a] E.J. Candes, J. Romberg, and T. Tao. ''Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information.'' *IEEE Transactions on Information Theory*, 2006.

[CRT06b] E.J. Candes, J.K. Romberg, and T. Tao. ''Stable signal recovery from incomplete and inaccurate measurements.'' *Communications on Pure and Applied Mathematics*, 2006.

[CT05] E.J. Candes and T. Tao. ''Decoding by linear programming.'' *IEEE Transactions on Information Theory*, 2005.

[CT06] EJ Candes and T. Tao. ''Near-optimal signal recovery from random projections: Universal encoding strategies?'' *Trans on Info Theory*, 2006.

[CW06] E.J. Candes and M.B. Wakin. ''People hearing without listening: An introduction to compressive sampling.'' *IEEE Signal Processing Magazine*, 2006.

[CWB08] E.J. Candes, M.B. Wakin, and S.P. Boyd. ''Enhancing Sparsity by Reweighted L1 Minimization.'' *Journal of Fourier Analysis and Applications*, 2008.

[CZK09a] Zainul M Charbiwala, Sadaf Zahedi, Younghun Kim, Supriyo Chakraborty, Ting He, Chatschik Bisdikian, and Mani B Srivastava. ''Improving Data Integrity with Randomness - A Compressive Sensing Approach.'' *Annual Conference of ITA (ACITA), in submission*, 2009.

[CZK09b] Zainul M Charbiwala, Sadaf Zahedi, Younghun Kim, Young H Cho, and Mani B Srivastava. ''Toward Quality of Information Aware Rate Control for Sensor Networks.'' *International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBID)*, 2009.

[DBH08] T. Dang, N. Bulusu, and W. Hu. ''Lightweight Acoustic Classification for Cane-Toad Monitoring.'' *Asilomar Conference on Signals, Systems and Computers*, 2008.

[DDT08] M.F. Duarte, M.A. Davenport, D. Takhar, J.N. Laska, T. Sun, K.F. Kelly, and R.G. Baraniuk. ''Single-pixel imaging via compressive sampling.'' *IEEE Signal Processing Magazine*, 2008.

[DDW06] MF Duarte, MA Davenport, MB Wakin, and RG Baraniuk. ''Sparse Signal Detection from Incoherent Projections.'' In *ICASSP*, 2006.

[DDW07] M.A. Davenport, M.F. Duarte, M.B. Wakin, J.N. Laska, D. Takhar, K.F. Kelly, and R.G. Baraniuk. ''The smashed filter for compressive classification and target recognition.'' In *SPIE*, 2007.

[Dec11] Decawave. *ScenSor*, 2011. `http://decawave.com/scensor.html`.

[Dep05] Department of Health. *Self-care - a real choice. Self-care support - a practical option.* London: The Stationery Office, 2005.

[dG08] Alexandre d'Aspremont and Laurent El Ghaoui. ''Testing the Nullspace Property using Semidefinite Programming.'', 2008.

[DGA05] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. ''Design of a wireless sensor network platform for detecting rare, random, and ephemeral events.'' In *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press Piscataway, NJ, USA, 2005.

[DH01] D.L. Donoho and X. Huo. ''Uncertainty principles and ideal atomic decomposition.'' *Information Theory, IEEE Transactions on*, **47**(7):2845--2862, 2001.

[Ell63] EO Elliott. ''Estimates of error rates for codes on burst-noise channels.'' *Bell Syst. Tech. J*, 1963.

[ES10] D. Estrin and I. Sim. ''Open mHealth Architecture: An Engine for Health Care Innovation.'' *Science*, **330**(6005):759, 2010.

[FGJ06] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo. ''The Collection Tree Protocol (CTP).'' *TinyOS Extension Proposal 123. V*, **8**, 2006.

[Fri06] S. Frintrop. *Vocus: A Visual Attention System for Object Detection And Goal-directed Search*. Springer-Verlag, 2006.

[FZS] Kai-Wei Fan, Zizhan Zheng, and Prasun Sinha. ''Steady and Fair Rate Allocation for Rechargeable Sensors in Perpetual Sensor Networks.'' In *SenSys'08*.

[GAG00] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. ''PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals.'' *Circulation*, 2000.

[GMP] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin. ''Capturing high-freq phenomena using a bandwidth-limited sensor network.'' In *IPSN'06*.

[GN] E. Gelenbe and E. Ngai. ''Adaptive QoS Routing for Significant Events in Wireless Sensor Networks.'' In *MobiSys'08*.

[GT] A. Griffin and P. Tsakalides. ''Compressed Sensing of Audio Signals Using Multiple Sensors.'' *EUSIPCO08*.

[HCB00] WR Heinzelman, A. Chandrakasan, H. Balakrishnan, and C. MIT. ''Energy-efficient communication protocol for wireless microsensor networks.'' In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000.

[HKC09] R.R. Harrison, R.J. Kier, C.A. Chestek, V. Gilja, P. Nuyujukian, S. Ryu, B. Greger, F. Solzbacher, and K.V. Shenoy. ''Wireless neural recording with single low-power integrated circuit.'' *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **17**(4):322--329, 2009.

[HLR] Nan Hua, Ashwin Lall, Justin Romberg, Jun (Jim) Xu, Mustafa al'Absi, Emre Ertin, Santosh Kumar, and Shikhar Suri. ''Just-in-time sampling and prefiltering for wearable physiological sensors: going from days to weeks of operation on a single charge.'' In *Wireless Health 2010*.

[HN07] J. Haupt and R. Nowak. ''Compressive Sampling for Signal Detection.'' In *ICASSP*, 2007.

[HYL09] L. Hoang, Z. Yang, and W. Liu. ''VLSI architecture of NEO spike detection with noise shaping filter and feature extraction using informative samples.'' In *Engineering in Medicine and Biology Society, 2009. EMBC 2009.*, pp. 978--981. IEEE, 2009.

[ISK08] Adnan Iqbal, Khurram Shahzad, Syed Ali Khayam, and Yongju Cho. ''CRAWDAD trace niit/bit_errors/802.15.4/Traces_802.15.4.'' Downloaded from http://crawdad.cs.dartmouth.edu/niit/bit_errors/802.15.4/Traces_802.15.4, July 2008.

[JE07] J. Jeong and C.T. Ee. ''Forward error correction in sensor networks.'' *International Workshop on Wireless Sensor Networks (WWSN)*, 2007.

[JRP07] M. Juusola, H.P.C. Robinson, and G.G. de Polavieja. ''Coding with spike shapes and graded potentials in cortical networks.'' *Bioessays*, **29**(2):178--187, 2007.

[Kay98] S.M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall PTR, 1998.

[KFD] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. ''Flush: a reliable bulk transport protocol for multihop wireless networks.'' In *SenSys'07*.

[KGM09] V. Karkare, S. Gibson, and D. Markovic. ''A 130-μW, 64-channel spike-sorting DSP chip.'' In *Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian*, pp. 289--292. IEEE, 2009.

[KLW06] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk. ''Analog-to-information conversion via random demodulation.'' In *DCAS*, 2006.

[KLX] A. Kashyap, LA Lastras-Montano, and C. Xia. ''Distributed source coding in dense sensor networks.'' In *DCC'05*.

[KMK11] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza. ''A real-time compressed sensing-based personal electrocardiogram monitoring system.'' In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011.

[KOM09] A.M. Kamboh, K.G. Oweiss, and A.J. Mason. ''Resource constrained VLSI architecture for implantable neural data compression systems.'' In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 1481 --1484, May 2009.

[KRS] A. Kansal, A. Ramamoorthy, MB Srivastava, and GJ Pottie. ''On sensor network lifetime and data distortion.'' In *ISIT'05*.

[KSG] A. Krause, A. Singh, and C. Guestrin. ''Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies.'' *JMLR'08*.

[KSS98] T. Klingenheben, C. Sticherling, M. Skupin, and S.H. Hohnloser. ''Intracardiac QRS electrogram width - an arrhythmia detection feature for implantable cardioverter defibrillators: exercise induced variation as a base for device programming.'' *Pacing and clinical electrophysiology*, 1998.

[KXA09] M. Amin Khajehnejad, Weiyu Xu, Amir Salman Avestimehr, and Babak Hassibi. ''Weighted ℓ_1 Minimization for Sparse Recovery with Prior Information.'' 2009.

[LD] J.C.F. Li and S. Dey. ''Lifetime Optimization for Multi-hop Wireless Sensor Networks with Rate Distortion Constraints.'' In *SPAWC'06*.

[LDP07] M. Lustig, D. Donoho, and J.M. Pauly. ''Sparse MRI: The application of compressed sensing for rapid MR imaging.'' *Magnetic Resonance in Medicine*, **58**(6):1182--1195, 2007.

[Le88] P. L'ecuyer. ''Efficient and portable combined random number generators.'' 1988.

[LKD07] 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 *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 1959--1962. Ieee, 2007.

[LKR12] W. Lu, K. Kpalma, J. Ronsin, et al. ''Sparse Binary Matrices of LDPC codes for Compressed Sensing.'' 2012.

[LLK10] Seung Bae Lee, Hyung-Min Lee, M. Kiani, Uei-Ming Jow, and M. Ghovanloo. ''An Inductively Powered Scalable 32-Channel Wireless Neural Recording System-on-a-Chip for Neuroscience Applications.'' *Biomedical Circuits and Systems, IEEE Transactions on*, **4**(6):360 --371, 2010.

[LRM] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli. ''A comparison of casting and spiraling algorithms for odor source localization in laminar flow.'' In *ICRA'08*.

[Lub02] M. Luby. ''LT Codes.'' In *Proceedings of the 43rd Symposium on Foundations of Computer Science*. IEEE Computer Society Washington, DC, USA, 2002.

[LV01] R.M.T. Laukkanen and P.K. Virtanen. ''Heart rate monitors: state of the art.'' *Journal of Sports Sciences*, **16**:3--7, 2001.

[LV10a] W. Lu and N. Vaswani. ''Modified Basis Pursuit Denoising (modified-BPDN) for noisy compressive sensing with partially known support.'' In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 3926--3929. IEEE, 2010.

[LV10b] W. Lu and N. Vaswani. ''Modified compressive sensing for real-time dynamic MR imaging.'' In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 3045--3048. IEEE, 2010.

[MB89] M. Mushkin and I. Bar-David. ''Capacity and coding for the Gilbert-Elliot channels.'' *IEEE Transactions on Information Theory*, 1989.

[MCL05] R. Madan, S. Cui, S. Lail, and A. Goldsmith. ''Cross-layer design for lifetime maximization in interference-limited wireless sensor networks.'' In *INFOCOM 2005*, volume 3, 2005.

[MDL] D. Marco, E.J. Duarte-Melo, M. Liu, and D.L. Neuhoff. ''On the Many-to-One Transport Capacity of a Dense Wireless Sensor Network and the Compressibility of Its Data.'' *LNCS*.

[ME10] M. Mishali and Y.C. Eldar. ''Xampling: Analog data compression.'' In *2010 Data Compression Conference*, pp. 366--375. IEEE, 2010.

[MED09] M. Mishali, Y.C. Eldar, O. Dounaevsky, and E. Shoshan. ''Xampling: Analog to digital at sub-Nyquist rates.'' *Arxiv preprint arXiv:0912.2495*, 2009.

[MLT08] Razvan Musaloiu-E., Chieh-Jan Mike Liang, and Andreas Terzis. ''Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks.'' In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, Washington, DC, USA, 2008. IEEE Computer Society.

[Mur06a] B. Murmann. ‘‘Digitally assisted analog circuits.’’ *Micro, IEEE*, **26**(2):38--47, 2006.

[Mur06b] Boris Murmann. ‘‘LIMITS ON ADC POWER DISSIPATION.’’ In Michiel Steyaert, Johan H. Huijsing, and Arthur H.M. van Roermund, editors, *Analog Circuit Design*. Springer Netherlands, 2006.

[MW95] R. N. McDonough and A.D. Whalen. *Detection of Signals in Noise, 2nd edition, Academic Press*. 1995.

[NB04] Z. Nenadic and J.W. Burdick. ‘‘Spike detection using the continuous wavelet transform.’’ *Biomedical Engineering, IEEE Transactions on*, **52**(1):74--87, 2004.

[NTC07] S. Narasimhan, M. Tabib-Azar, H.J. Chiel, and S. Bhunia. ‘‘Neural Data Compression with Wavelet Transform: A Vocabulary Based Approach.’’ In *Neural Engineering, 2007. CNE ’07. 3rd International IEEE/EMBS Conference on*, pp. 666 --669, May 2007.

[NTG] R. Newton, S. Toledo, L. Girod, H. Balakrishnan, and S. Madden. ‘‘Wishbone: Profile-based Partitioning for Sensornet Applications.’’.

[Nyq28] H. Nyquist. ‘‘Certain Factors Affecting Telegraph Speed. AT&T, 1924 and Certain topics in telegraph transmission theory.’’ *Trans. American Institute of Elect. Eng*, 1928.

[OL08] A. Olson and D. Langlois. ‘‘Solid State Drives Data Reliability and Lifetime.’’ In *Imation White Paper*, 2008.

[Ooh98] Y. Oohama. ‘‘The rate-distortion function for the quadratic Gaussian CEO problem.’’ *Information Theory, IEEE Transactions on*, 1998.

[OOM] E. O’Connell, S. O’Connell, R.P. McEvoy, and W.P. Marnane. ‘‘A low-power wireless ECG processing node and remote monitoring system.’’ In *Signals and Systems Conference (ISSC 2010), IET Irish*, pp. 48--53. IET.

[Owe06] K.G. Oweiss. ‘‘A systems approach for data compression and latency reduction in cortically controlled brain machine interfaces.’’ *Biomedical Engineering, IEEE Transactions on*, **53**(7):1364--1377, 2006.

[PG] J. Paek and R. Govindan. ‘‘RCRT: rate-controlled reliable transport for wireless sensor networks.’’ In *SenSys’07*.

[Pol73] G. Polya. ‘‘How to Solve It: A New Aspect of Mathematical Method.’’ *New York*, 1973.

[PR03] SS Pradhan and K. Ramchandran. ‘‘Distributed source coding using syndromes (DISCUS): design and construction.’’ *Information Theory, IEEE Transactions on*, 2003.

[PRH11] Kurt Plarre, Andrew Raij, Syed Monowar Hossain, Amin A Ali, Motohiro Nakajima, Mustafa al'Absi, Emre Ertin, Thomas Kamarck, Santosh Kumar, Marcia Scott, Daniel Siewiorek, Asim Smailagic, and Larry Wittmers. ''Continuous Inference of Psychological Stress from Sensory Measurements Collected in the Natural Environment.'' In *IPSN*, 2011.

[PTR] V. Prabhakaran, D. Tse, and K. Ramachandran. ''Rate region of the quadratic Gaussian CEO problem.'' In *ISIT'04*.

[Qua11] Qualcomm Inc. *Peanut: Low Power Short Range Radio*, 2011.

[RGF01] P. Rubel, F. Gouaux, J. Fayn, D. Assanelli, A. Cuce, L. Edenbrandt, and C. Malossi. ''Towards intelligent and mobile systems for early detection and interpretation of cardiological syndromes.'' In *Computers in Cardiology 2001*, pp. 193--196. IEEE, 2001.

[RGG] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. ''Interference-aware fair rate control in wireless sensor networks.'' In *SIGCOMM'06*.

[RM06] U. Rutishauser, , A.N. Mamelak, and E.M. Schuman. ''Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo.'' *Journal of Neuroscience Methods*, **154**:204--224, 2006.

[Roo10] M.J. Rooijakkers. ''Design space exploration for scalable R-peak detection.'' 2010.

[RS60] IS Reed and G. Solomon. ''Polynomial codes over certain finite fields.'' *Journal of the Society for Industrial and Applied Mathematics*, pp. 300--304, 1960.

[RV06] M. Rudelson and R. Vershynin. ''Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements.'' In *Information Sciences and Systems,*, 2006.

[SBB06] S. Sarvotham, D. Baron, and R.G. Baraniuk. ''Measurements vs. bits: Compressed sensing meets information theory.'' In *Proceedings of 44th Allerton Conf. Comm., Ctrl., Computing*, 2006.

[SBW09] D. Schmidt, M. Berning, and N. Wehn. ''Error Correction in Single-Hop Wireless Sensor Networks-A Case Study.'' *Design, Automation, and Test in Europe (DATE) conference*, 2009.

[Sha49] CE Shannon. ''Communication in the presence of noise.'' *Proceedings of the IRE*, 1949.

[SHE03] M.O. Sweeney, A.S. Hellkamp, K.A. Ellenbogen, A.J. Greenspon, R.A. Freedman, K.L. Lee, and G.A. Lamas. ''Adverse effect of ventricular pacing on

heart failure and atrial fibrillation among patients with normal baseline QRS duration in a clinical trial of pacemaker therapy for sinus node dysfunction.'' *Circulation*, 2003.

[Shi10a] E.I. Shih. *Reducing the computational demands of medical monitoring classifiers by examining less data*. PhD thesis, MIT, 2010.

[Shi10b] Shimmer Research. ''Shimmer Biophysical Expansion User Guide I, Revision 1a.'', 2010.

[SHS98] T. M. Seese, H. Harasaki, G. M. Saidel, and C. R. Davies. ''Characterization of Tissue Morphology, Angiogenesis, and Temparature in Adaptive Response of Muscele Tissue to Chronic Heating.'' *Lab Investigaion, vol. 78(12)*, 1998.

[Sic] M.L. Sichitiu. ''Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks.'' In *INFOCOM'04*.

[SKA08] K. Srinivasan, M.A. Kazandjieva, S. Agarwal, and P. Levis. ''The β-factor: measuring wireless link burstiness.'' In *SenSys*, 2008.

[SKG] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin. ''Efficient planning of informative paths for multiple robots.'' In *IJCAI'07*.

[SLD11] JP Slavinsky, J. Laska, M. Davenport, and R. Baraniuk. ''The compressive mutliplexer for multichannel compressive sensing.'' In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP), Prague, Czech Republic*, 2011.

[SM05] V. Srivastava and M. Motani. ''Cross-layer design: a survey and the road ahead.'' *IEEE CommMag*, 2005.

[SO11] K.G. Stanley and N.D. Osgood. ''The potential of sensor-based monitoring as a tool for health care, health promotion, and research.'' *The Annals of Family Medicine*, **9**(4):296, 2011.

[SRK03] S. Shakkottai, TS Rappaport, and PC Karlsson. ''Cross-layer design for wireless networks.'' *IEEE CommMag*, 2003.

[Tex] Texas Instruments. *TS3A4751: 0.9 Ohm Low-Voltage Single-Supply Quad SPST Analog Switch*.

[Tex09] Texas Instruments. *Quad Current Input 20-Bit Analog-To-Digital Converter*, 2009.

[Tex11] Texas Instruments. *MSP430L092: 0.9V 16-bit Mixed Signal Microcontroller*, 2011.

[TG07] JA Tropp and AC Gilbert. ''Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit.'' *Information Theory, IEEE Transactions on*, 2007.

[TLD10] J.A. Tropp, J.N. Laska, M.F. Duarte, J.K. Romberg, and R.G. Baraniuk. ''Beyond Nyquist: Efficient sampling of sparse bandlimited signals.'' *Information Theory, IEEE Transactions on*, **56**(1):520--544, 2010.

[TLP05] B.L. Titzer, D.K. Lee, and J. Palsberg. ''Avrora: Scalable sensor network simulation with precise timing.'' In *IPSN*, 2005.

[Van68] H.L. Van Trees. *Detection, estimation, and modulation theory.. part 1,. detection, estimation, and linear modulation theory*. Wiley New York, 1968.

[VB97] H. Viswanathan and T. Berger. ''The quadratic Gaussian CEO problem.'' *Information Theory, IEEE Transactions on*, 1997.

[VBN] P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, and A. Ledeczi. ''Shooter localization and weapon classification with soldier-wearable networked sensors.'' In *MobiSys'07*.

[Web98] J.G. Webster et al. *Medical instrumentation: application and design*. John Wiley, New York, 1998.

[Whi15] E.T. Whittaker. *On the functions which are represented by the expansions of the interpolation-theory*. Edinburgh University, 1915.

[WM95] HS Wang and N. Moayeri. ''Finite state Markov channel-a useful model for radio communications systems.'' *IEEE Transactions on Vehicular Technology*, 1995.

[WS09] Anthony D. Wood and John A. Stankovic. ''Online Coding for Reliable Data Transfer in Lossy Wireless Sensor Networks.'' In *Distributed Computing in Sensor Systems: 5th IEEE International Conference, DCOSS 2009, Marina Del Rey, CA, USA, June 8-10, 2009, Proceedings*, 2009.

[WVR87] G. Wegmann, E.A. Vittoz, and F. Rahali. ''Charge injection in analog MOS switches.'' *Solid-State Circuits, IEEE Journal of*, 1987.

[YHS08] Z. Yu, S. Hoyos, and B.M. Sadler. ''Mixed-signal parallel compressed sensing and reception for cognitive radio.'' In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 3861--3864. IEEE, 2008.

[YSH] W. Ye, F. Silva, and J. Heidemann. ''Ultra-low duty cycle MAC with scheduled channel polling.'' In *SenSys'06*.