RICE UNIVERSITY

Distributed Redundant Representations in Man-made and Biological Sensing Systems

by

Christopher John Rozell

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Approved, Thesis Committee:

Don H. Johnson, Chair
J.S. Abercrombie Professor, Electrical and Computer Engineering and Statistics

---

Richard G. Baraniuk
Victor E. Cameron Professor,
Electrical and Computer Engineering

Mark P. Embree
Assistant Professor,
Computational and Applied Mathematics

Bruno A. Olshausen
Associate Professor, Helen Wills Neuroscience Institute and School of Optometry,
University of California, Berkeley

Houston, Texas
April 2007

ABSTRACT

Distributed Redundant Representations in
Man-made and Biological Sensing Systems
by
Christopher J. Rozell

The ability of a man-made or biological system to understand its environment is limited by the methods used to process sensory information. In particular, the data representation is often a critical component of such systems. Neural systems represent sensory information using distributed populations of neurons that are highly redundant. Understanding the role of redundancy in distributed systems is important both to understanding neural systems and to efficiently solving many modern signal processing problems.

This thesis makes contributions to understanding redundant representations in distributed processing systems in three specific areas. First, we explore the robustness of redundant representations by generalizing existing results regarding noise-reduction to Poisson process modulation. Additionally, we characterize how the noise-reduction ability of redundant representation is weakened when we enforce a distributed processing constraint on the system.

Second, we explore the task of managing redundancy in the context of distributed settings through the specific example of wireless sensor and actuator networks (WSANs). Using a crayfish reflex behavior as a guide, we develop an analytic WSAN model that implements control laws in a completely distributed manner. We also develop an algorithm to optimize the system resource allocation by adjusting the number of bits used to quantize messages on each sensor-actuator communication link. This optimal power scheduling yields several orders of magnitude in power savings over uniform allocation strategies that use a fixed number of bits on each communication link.

Finally, we explore the flexibility of redundant representations for sparse approximation. Neuroscience and signal processing both need a sparse approximation algorithm (i.e., representing a signal with few non-zero coefficients) that is physically implementable in a parallel system and produces smooth coefficient time-series for time-varying signals (e.g., video). We present a class of *locally competitive algorithms* (LCAs) that minimize a weighted combination of mean-squared error and a coefficient cost function. LCAs produce coefficients with sparsity levels comparable to centralized algorithms while being more realistic for physical implementation. The resultant LCA coefficients for video sequences are more regular (i.e., smoother and more predictable) than the coefficients produced by existing algorithms.

Acknowledgements

> "Religion, morality, and knowledge, being necessary to good government and the happiness of mankind, schools and the means of education shall forever be encouraged." –Northwest Ordinance (1787)

Only a fraction of the world's population has access to formal education beyond primary school. I find it difficult to express my gratitude for the opportunity I have had to explore the limits of studenthood at two world-class institutions: Rice University and the University of Michigan (where I received an "uncommon education for the common man"). Reading, thinking and writing are the only craft I know. I am fortunate to have had the chance to develop these skills in such fantastic environments.

The opportunities I have received did not occur by chance. Many people have given of their time, talents and resources on my behalf. In particular, my family has made sacrifices that I will likely never understand so that I could enjoy opportunities that were never available to them. Though my hands don't have the same calluses, I have learned more about integrity and hard work by watching my parents, grandparents and great-grandparents than I ever could have learned in a classroom. Whatever success I may enjoy in life is due in large part to lessons I learned from them.

My thesis committee has given generously (and patiently) of their time, energy and wisdom to further my professional development: Mark Embree provided thorough and valuable comments, in addition to setting an example as an accomplished researcher who is an artist in the classroom; Bruno Olshausen provided ideas and discussions that were insightful, while demonstrating an exciting passion for uncovering the mysteries of neuroscience; Rich Baraniuk taught me how to see my research (and myself) in the context of the big picture, all with a boundless energy that is contagious; and my advisor Don Johnson taught me to pursue hard problems by breaking them down into simple ideas that illustrate a fundamental truth. I am thankful for the freedom Don allowed me, the guidance he provided when I was stuck, and the countless hours he devoted to developing my research and communication skills.

I cannot imagine a more friendly and encouraging place to work than the ECE Department at Rice. Though many people have come in and out of the department during my time here, this group has consistently been kind, fun, interesting and helpful. Thank you for providing a great place to come to work every day. Similarly, I cannot imagine having a more supportive and inspiring group of friends (near and far). Thank you for the joy and meaning that you bring to my life.

Finally, I have been blessed beyond words by my wife Kara. Despite seeing me at my lowest points, she was an unrelenting source of encouragement and hope. The sacrifices she has made for me have taught me more about grace than what I could ever learn on my own. Thank you for walking with me down this path — I promise to never do it again.

Soli Deo gloria

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Sensing Systems

Sensing systems are the window that allows both man-made and biological systems to view the outside world. The ability of a machine or a biological organism to understand the local environment relies on the methods used to process this sensory information. In particular, a key aspect of processing this sensed data is choosing an appropriate way to represent the information of interest to the system. Much of the success of modern signal processing in tasks such as compression and denoising can be traced to finding a representation (e.g., Fourier, wavelets, etc.) that matches well to "interesting" signals (data) but not to "uninteresting" signals (noise).

Despite making significant progress in a number of important problems, general signal processing solutions to "scene analysis" tasks such as speech recognition and computer vision are still disappointing. Though their information processing strategies are not well understood, neural systems are very adept at understanding their environment even in very hostile conditions. For example, man-made solutions for recognizing speech have steadily improved over recent decades but they still cannot approach the perceptual abilities of human beings in general settings [116]. Understanding more about the information processing strategies used by neural systems would almost certainly improve machine performance in such tasks, even under different implementation constraints.

Beyond simply mimicking the tasks of biological systems, many modern signal processing problems share a common structure with the problems solved by neural systems. For instance, signal processing tasks often require extracting very structured information from high-dimensional, multi-modal data. Understanding how the information processing strategies used by neural system enable their perceptual abilities in one task can inspire solutions to many other problems that share a similar structure.

Unfortunately, we do not have a clear understanding of the information representations used by most sensory neural systems. Current technological limitations prevent us from acquiring the vast amounts of data required to decipher the tactics used by large and complex neural systems to process sensory information. Consequently, a mature understanding of these neural systems will likely only come through a combination of experimental data *and* theoretical models providing testable hypotheses.

Neural systems represent sensory information using neural populations that have two characteristics that are not well understood: redundancy and distributed processing. This thesis will focus on understanding the role of these particular characteristics

for two reasons. First, these characteristics appear to be ubiquitous in biological sensory neural systems. Second, the roles of both redundancy and distributed processing are increasingly important in modern signal processing problems.

1.2 Redundancy

Despite not knowing the inner workings of sensory neural systems, redundancy clearly plays a crucial role in representing information. Anatomical observations indicate that shortly after the transduction front-end, sensory systems process data in successive stages each composed of a neural population [87]. Examples from both the early visual pathway [112] and the early auditory pathway [121] indicate that these populations are redundant, using many more neurons than the previous stage. This observation raises the question "Why do neural systems use so much redundancy?" More specifically, what perceptual tasks does redundancy enable that would not be possible with a critically sampled representation?

Conventional signal processing wisdom associates redundancy with robustness. This viewpoint states that redundancy is useful to recover from corruption but should be removed in all other cases for maximum efficiency. This conflict appears in the classic information theory result known as the "separation theorem" [140], which divides the fundamental communication pathway into two stages. The first stage (source coding) removes as much redundancy as possible from a signal representation. The second stage (error control coding) introduces redundancy back into the signal representation to counter the inevitable communication errors.

While the communications paradigm describes efficient information transmission, it does not address the task of trying to efficiently *understand* the content of a sensed signal (i.e., trying to infer the environmental conditions responsible for the signal). When considering this broader task, a second advantage of redundancy comes to light: redundancy allows *flexibility* in the representation. Critically sampled representations (e.g., an orthonormal basis) contain enough information to reconstruct a signal but they must use a single fixed strategy for the encoding. In a redundant representation, the redundancy allows us to choose from among many possible encodings to find one that not only communicates the sensed signal but also makes the content of that signal easier to infer. Understanding the role that this flexibility can play is a critical step to understanding the nature of redundant representations in sensory neural systems.

1.3 Distributed Processing

Many modern signal processing problems rely on distributed processing, either because the data was originally gathered in that form (e.g., from a physically dispersed collection of sensors) or because the available processing resources are inadequate for centralized computation on a full dataset. Again, conventional wisdom says to remove the distributed processing constraints as quickly as possible in favor of centralized

computations. The most favorable situation distills the data down to a single sufficient statistic that can be evaluated in one central location.

This desire for data centralization is understandable. Managing the redundancy in a representation requires coordination that is most easily achieved with centralized oversight. In contrast, neural systems appear to take an opposite strategy. Rather than centralize information down to a single decision-making cell (known as a "grandmother cell"), neural systems appear to use the joint activity in successively larger neural populations to analyze a sensory stimulus. The example of neural computation indicates that there exist efficient strategies for managing redundant representations even without centralized control.

1.4 Contributions

This thesis takes the viewpoint that both our comprehension of neural systems and the quality of our signal processing solutions can be improved by better understanding the consequences of redundancy and distributed processing in the representation of sensed data. The approach we take is to abstract the sensing process using a simple linear vector space model to explore the fundamental role of these two characteristics. Specifically, this thesis explores three different aspects of redundant representations:

- the robustness of redundant representations;
- the coordination of redundant representations when turning sensing into actuation; and
- the flexibility of redundant representations for sparse approximation.

In each case, the system operates under distributed processing requirements.

Chapter 3 addresses the robustness of redundant representations in the context of noise reduction. This chapter generalizes known noise-reduction results from additive noise to Poisson process modulation. This generalization is important because nonlinear point process noise sources are a more appropriate model for neural communication than additive noise. The nonlinear nature of point process noise sources can cause significant difference from the additive noise case. Additionally, this chapter characterizes the amount of noise-reduction ability that is lost in a redundant representation when a distributed processing constraint is enforced. This assessment is a critical step to understanding how robustness is affected by the distributed processing constraints seen in both neural systems and many interesting signal processing problems.

Chapter 4 addresses the task of managing redundancy when turning sensed data into action. Wireless sensor and actuator networks (WSANs) use a distributed collection of resource constrained sensor nodes to collect data about the environment and a distributed collection of actuator nodes to affect the environment. We desire WSAN strategies similar to the behavior seen in neural systems: turning sensed data into

action without centralized control in a very redundant system. Using a crayfish reflex behavior as a guide, this chapter develops an analytic model for a class of WSANs that demonstrates many control laws can be implemented in a completely distributed manner.

In this WSAN model, a weight is associated with each sensor-actuator link denoting the importance of that communication link to the actuation fidelity. We show that these weights are useful in pruning away communication links to reduce the number of active channels. Inspired by recent work in power scheduling for decentralized estimation, this chapter also optimizes the allocation of system resources to achieve a desired actuation fidelity. In this scheme, each sensor acquires a noisy observation and sends a message to a subset of actuators. The message sent on each sensor-actuator communication link is quantized with a variable number of bits, with the number of bits optimized to minimize the total network power consumption subject to a constraint on the actuation distortion. We show analytically and verify through simulation that performing this optimal power scheduling can yield several orders of magnitude in power savings over communication strategies that use a uniform resource allocation on each communication link.

Chapter 5 addresses the flexibility of redundant representations in the context of sparse approximation. Several researchers hypothesize that neural systems may employ *sparse coding*, using the flexibility of a redundant representation to encode a stimulus through the activity of just a few neurons in a large population [114]. However, the mechanisms used by neural systems to produce these sparse codes are still unknown. Research over the last decade has also shown that this sparse approximation problem (i.e., finding a signal representation using as few non-zero coefficients as possible) is relevant to many signal processing tasks.

Sparse approximation methods currently employed in signal processing tasks suffer two significant drawbacks: they are difficult to implement in a physical system (hardware or neural), and they are inefficient for time-varying stimuli (e.g., video) because they produce erratic temporal coefficient sequences. We present a class of neurally plausible *locally competitive algorithms* (LCAs) that correspond to a collection of sparse coding principles minimizing a weighted combination of mean-squared error (MSE) and a coefficient cost function. LCAs use thresholding functions to induce local (usually one-way) inhibitory competitions. LCAs demonstrate sparsity levels comparable to existing sparse coding algorithms while being more realistic for neural and hardware implementation. Additionally, LCA coefficients for video sequences demonstrate inertial properties that are both qualitatively and quantitatively more regular (i.e., smoother and more predictable) than the coefficients produced by greedy algorithms. LCAs represent a first step toward producing testable predictions for the sparse coding hypothesis in neural systems as well as giving insight to new signal representation that could improve our processing of time-varying signals.

Chapter 2
General Background and Notation

2.1 Mathematical Preliminaries

2.1.1 General notation

We will be concerned with signals $\boldsymbol{x}$ considered to be elements from a finite dimensional vector space $\boldsymbol{x} \in \mathcal{H}$. We simply consider signals as N-dimensional vectors $\boldsymbol{x} \in \mathbb{R}^N$, where the meaning of the dimension (e.g., temporal samples, spatial samples, etc.) is inferred from the context. We will often use the terms signal, vector and stimulus interchangeably to refer to $\boldsymbol{x}$. The scalar components of the vector are notated with the appropriate non-boldface symbol and a subscript, $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^t$, where t indicates the transpose operator. Matrices will generally be notated with boldface capital letters such as $\mathbf{X}$. In particular, the identity matrix is denoted $\mathbf{I}$, with the exact dimensions given by the context. Scalars are always notated with non-boldface symbols and are either uppercase or lowercase depending on the accepted convention of the context.

We assume that a vector space $\mathcal{H}$ is endowed with the usual inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{n=1}^{N} x_n y_n$ (where $\boldsymbol{y} \in \mathcal{H}$) and induced norm $\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2}$. We will also employ more general ℓ^p norms,[1] where $\|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} |x_n|^p \right)^{1/p}$ for $p \in (0, \infty)$. The norm $\|\cdot\|$ (without a subscript) on $\mathcal{H}$ is therefore equivalent to the ℓ^2 norm $\|\cdot\|_2$. The distance between two signals is often quantified through the mean-squared error (MSE), given through the normalized ℓ^2 norm of the difference signal $\frac{1}{N} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.

We also adopt the standard convention of defining the ℓ^0 norm to be $\|\boldsymbol{x}\|_0 = \sum_{n=1}^{N} \mathbf{1}_{(x_n \neq 0)}$, where $\mathbf{1}_{(\cdot)}$ is the indicator function. While the notation $\|\cdot\|_0$ is simply a convenient way to denote a measure counting the non-zero elements of a vector, the origins of this notation can be seen by taking the limit of $\|\cdot\|_p^p$ as $p \to 0$.

2.1.2 Abstract sensing systems

Sensing systems consist of a collection of elements that each measure information about their local environment. This is most easily imagined with spatially distributed sensors (e.g., temperature sensors in a forest, or photoreceptors in the retina) that each

[1] The ℓ^p "norms" only satisfy the mathematical requirements for a norm when $p \geq 1$. However, we adopt the standard (though slightly misleading) terminology of referring unashamedly to these measures as norms regardless of the value of p.

measure an environmental field over their local spatial region. However, this same notion is also true in other domains. For example, the inner hair cells of the cochlea have a "local environment" defined as a specific frequency region. In a complementary example, each time sample of an analog-to-digital converter has a "local environment" defined as a specific time-window. Our focus will be on the collective representation of the information after acquisition. To this end, we need an abstract model of the sensing process that encompasses all of these scenarios.

Due to the dynamics of the local environment being sensed, sensor measurement models often consist of averaging the stimulus field over their local environment. We therefore represent each sensor by a "receptive field"[2] $\boldsymbol{\phi}_k$, where k is an index variable denoting the specific sensor element (e.g., spatial location, temporal sample number, center frequency, etc.). This receptive field incorporates both the physics of the device (e.g., preferential sensing directions) and the physics of the environment surrounding the sensor. For example, the receptive field of a sensor may incorporate the spatial averaging induced by the propagation properties of the surrounding medium, corresponding to the Green's function [26] for the process being sensed. As a specific example, a soil moisture sensor is represented by a spatial average of the true moisture defined by the diffusion behavior [101] of the soil.

Single ideal measurements of the signal $\boldsymbol{x}$ taken by the k^{th} sensor are therefore given by

$$m_k = \langle \boldsymbol{\phi}_k, \boldsymbol{x} \rangle,$$

where the inner product is taken over the appropriate domain for $\mathcal{H}$ (e.g., time, space, frequency, etc.). Note that this is a very general notion of sensing, encompassing the familiar Shannon-Nyquist sampling of a time series [99], the spatial averaging of an environmental sensor [109], and recent non-traditional notions of signal measurement using random projections [13, 50]. While this sensing model allows both positive and negative measurements, some systems are limited to only having positive coefficients (e.g., neural systems using point process communication can only employ positive intensity functions). This situation is easily addressed in practice by splitting our abstract sensing nodes into two physical nodes. These physical nodes have receptive fields that are opposite in sign and their measurements are rectified.

If the system must capture any signal $\boldsymbol{x} \in \mathcal{H}$, the sensor vectors must span the input signal space, $\mathcal{H} \subseteq \text{span}\left(\{\boldsymbol{\phi}_k\}\right)$. The space $\mathcal{H}$ represents a restricted class of fields that is consistent with the resolution of the sensors. For example, $\mathcal{H}$ may be a space of spatially or temporally bandlimited functions. The actual stimulus field in the environment may not be in $\mathcal{H}$, but the sensors have a limited resolution that precludes them from sensing an unrestricted class of signals. Therefore, $\mathcal{H}$ assumes the role of a signal model, quantifying the component of the true environmental signals most important to the specific application.

[2]The terminology "receptive field" is taken from the neuroscience community where neurons are often characterized by the stimulus causing the strongest response.

This linear model of sensor measurements obviously cannot capture the nonlinearities often seen in real sensing systems. However, this linear model suits the purposes of this study for two reasons. First, many observed nonlinearities in the transduction process are pointwise memoryless functions. In other words, the nonlinearity modifies the dynamic range of the sensor but does not change the fundamental information being captured. These nonlinearities can often be thought of as an invertible transformation through large portions of the dynamic range where the system usually operates. In contrast, one type of interactive (non-invertible) nonlinearity will be discussed in detail in Chapter 5. Second, understanding the fundamental consequences of redundancy and distributed processing requires an analytically tractable model of the sensing process. Including sensing nonlinearities would only complicate the analysis and may not shed additional insight at this point.

2.1.3 Frame theory

Section 2.1.2 describes an abstraction of the sensing process involving a collection of linear projections onto vectors spanning a vector space. This notion is very familiar: projecting a signal onto an orthonormal basis (ONB) is a fundamental tool in signal processing (e.g., wavelet and Fourier transforms). However, general sensing systems deviate from an ONB because they are *redundant*.

In words, this notion of redundancy means that there are more measurements acquired than are necessary to capture the signal class of interest. Mathematically, this situation corresponds to having more vectors $\{\boldsymbol{\phi}_k\}_{k=1}^K$ than the dimension of the signal space $N = \dim(\mathcal{H})$, $K > N$. While we are normally interested in substantially redundant systems where $K \gg N$, any amount of overcompleteness means that the vectors no longer constitute a basis. The generalization of a basis to an overcomplete set of representation elements is known as a *frame* [15, 20, 55, 76, 93].

Frames were originally introduced to the signal processing community in 1952 in the context of nonharmonic Fourier series [55]. However, overcomplete representations would not gain significant popularity until much later when they played a key role in wavelet theory [33, 34, 36]. The appeal of frames is rooted in the flexibility they provide. In one viewpoint (driven by research in wavelet analysis), the redundancy introduces extra degrees of freedom that provide an avenue to get properties such as compactly supported filters or increased shift-invariance in a wavelet representation. In another viewpoint, the redundancy allows one to build a representation that is well-matched to signals that do not match well to a single ONB (e.g., a signal consisting of a combination of sine waves and impulses). Frames have recently been used in many other areas to model redundant systems, including filterbanks [10], quantization [6, 11, 35, 74], image processing [12, 23, 53, 65, 94], communications [144, 149, 152], coding [14, 73, 75, 138] and machine learning [68].

Specifically, a collection of K vectors $\{\boldsymbol{\phi}_k\}$ forms a *frame* for $\mathcal{H}$ if there exist

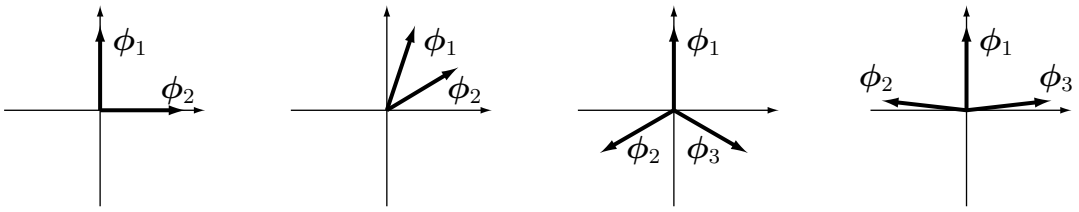


Figure 2.1: Example basis and frame representations in $\mathbb{R}^2$. (a) An orthonormal basis (ONB). (b) A non-orthogonal basis. (c) A tight frame with three vectors ($A = B = \frac{3}{2}$). (d) A general frame with three vectors ($A < \frac{3}{2} < B$).

constants $0 < A \leq B < \infty$ so that Parseval's relation is bounded for any $\boldsymbol{x} \in \mathcal{H}$,

$$A \|\boldsymbol{x}\|^2 \leq \sum_{k=1}^{K} |\langle \boldsymbol{\phi}_k, \boldsymbol{x} \rangle|^2 \leq B \|\boldsymbol{x}\|^2 .$$

The constants A and B are called the upper and lower *frame bounds*. Without losing generality for our applications, we assume the frame vectors are all unit-norm, $\|\boldsymbol{\phi}_k\| = 1, \forall k$ (sometimes called a *uniform frame*). Under this condition, the frame bounds satisfy $A \leq \frac{K}{N} \leq B$, and are viewed as measuring the minimum and maximum redundancy of the system. Figure 2.1 illustrates some simple example bases and frames in $\mathbb{R}^2$. While frame theory also includes generalizations to infinite dimensions, we will only discuss frames having a finite number of elements ($K < \infty$).

The frame condition given above guarantees that the collection of measurements $\{m_k\}$ (also called analysis coefficients) obtained from projecting a signal onto the frame vectors contains all of the information necessary to synthesize (or reconstruct) the signal. Mathematically, the analysis coefficients are generated through the frame analysis operator $\boldsymbol{\Phi} : \mathcal{H} \to \mathbb{R}^K$, given by $\boldsymbol{\Phi}\boldsymbol{x} = [m_1, m_2, \ldots, m_K]^t$. In the finite dimensional cases we consider, $\boldsymbol{\Phi}$ is simply a $(K \times N)$ matrix containing the vector $\boldsymbol{\phi}_k$ on the k^{th} row.

In finite dimensions, the adjoint of the frame analysis operator is simply the matrix transpose, $\boldsymbol{\Phi}^t : \mathbb{R}^K \to \mathcal{H}$, known as the frame synthesis operator. The composition is given by

$$\boldsymbol{\Phi}^t \boldsymbol{\Phi} \boldsymbol{x} = \sum_{k=1}^{K} \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle \boldsymbol{\phi}_k .$$

Though the analysis and synthesis operators are inverse operations in an ONB, the dependency between frame vectors means that they are *not* inverse operations in a frame, $\boldsymbol{\Phi}^t \boldsymbol{\Phi} \neq \mathbf{I}$. In other words, the same set of vectors cannot be used for both analysis and synthesis.

In general, $\boldsymbol{\Phi}$ represents an underdetermined system and therefore does not have

a unique inverse operator. The pseudoinverse operator $\boldsymbol{\Phi}^\dagger = \left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^t$ is most commonly used for reconstruction,

$$\boldsymbol{x} = \boldsymbol{\Phi}^\dagger\boldsymbol{\Phi}\boldsymbol{x} = \left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\right)^{-1}\sum_{k=1}^{K}\langle\boldsymbol{x},\boldsymbol{\phi}_k\rangle\boldsymbol{\phi}_k.$$

Equivalently, we can view the reconstruction as using a different set of vectors $\{\widetilde{\boldsymbol{\phi}}_k\}$ called the dual set, $\boldsymbol{x} = \sum_{k=1}^{K}\langle\boldsymbol{x},\boldsymbol{\phi}_k\rangle\widetilde{\boldsymbol{\phi}}_k$. While there are an infinite number of sets of dual vectors that will work for reconstruction, the canonical dual vectors are given by $\widetilde{\boldsymbol{\phi}}_k = \left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\phi}_k$, corresponding to choosing the pseudoinverse to invert the analysis operator $\boldsymbol{\Phi}$. The canonical dual vectors are also a frame for $\mathcal{H}$, with lower and upper frame bounds $\left(\frac{1}{B},\frac{1}{A}\right)$, respectively. Also note that the frame vectors and the dual set are interchangeable in the reconstruction equation,

$$\boldsymbol{x} = \sum_{k=1}^{K}\langle\boldsymbol{x},\boldsymbol{\phi}_k\rangle\widetilde{\boldsymbol{\phi}}_k = \sum_{k=1}^{K}\langle\boldsymbol{x},\widetilde{\boldsymbol{\phi}}_k\rangle\boldsymbol{\phi}_k.$$

The frame bounds are related directly to the eigenstructure induced by the frame vectors: A and B are the minimum and maximum eigenvalues (respectively) of $(\boldsymbol{\Phi}^t\boldsymbol{\Phi})$. When a collection of vectors has frame bounds that are equal, $A = B = \frac{K}{N}$, it is called a *tight frame*. When a frame is tight, the dual vectors are simply re-scaled versions of the frame vectors, $\widetilde{\boldsymbol{\phi}}_k = \frac{1}{A}\boldsymbol{\phi}_k$. A collection of vectors is an ONB if and only if it is a tight frame with $A = B = 1$.

This frame theory viewpoint describes signal representation as an analysis operation (sensing) paired with a synthesis operation (reconstruction). However, we can take a more general approach to representing a signal. Consider the frame vectors as constituting a dictionary, $\mathcal{D} = \{\boldsymbol{\phi}_k\}$. We can then represent a signal vector $\boldsymbol{x}$ by drawing selected elements (sometimes called atoms) from this dictionary and forming a linear combination with a set of coefficients $\{a_k\}$

$$\boldsymbol{x} = \sum_{k=1}^{K} a_k\boldsymbol{\phi}_k.$$

Though the coefficients are unique when $\mathcal{D}$ is an ONB, there are an infinite number of ways to choose the coefficients when $\mathcal{D}$ is overcomplete. This flexibility can be used in nonlinear encodings to find coefficients that represent the signal *and* have other properties. One example of this is the sparse approximation problem, where coefficients are chosen to have as few non-zero elements as possible (e.g., see Chapter 5).

2.1.4 Perturbations in a frame

Despite the pace of modern technological innovation, we do not expect to ever be able to store and transmit signals with infinite precision. Because we rarely have the luxury of a perfect signal representation, one of the most obvious benefits of a redundant representation is the ability to reduce corruption in the coefficients representing a signal. In an ONB, perturbing a measurement coefficient (including removing it entirely) has a proportional impact on the reconstruction — the energy in the reconstruction error is the same as the energy in the perturbation. While the redundancy of a frame can provide a measure of robustness to these perturbations, it also makes the effect of such perturbations harder to analyze. In order to design and analyze systems utilizing this benefit, we must quantify the effect of coefficient perturbations on a signal representation.

Stated generally, we need to calculate a bound on the maximum error when a perturbation p_k is added to each frame coefficient m_k in the reconstruction, $\widehat{\boldsymbol{x}} = \sum_{k=1}^{K} (m_k + p_k)\, \widetilde{\boldsymbol{\phi}}_k$. Perturbations may include removing the coefficient from the reconstruction, $p_k = -(m_k)$. The error resulting from these perturbations is

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2 = \left\| \sum_{k=1}^{K} p_k \widetilde{\boldsymbol{\phi}}_k \right\|^2 . \tag{2.1}$$

We recall that the dual set $\{\widetilde{\boldsymbol{\phi}}_k\}$ is also a frame for $\mathcal{H}_{\boldsymbol{x}}$, and we denote the analysis operator for the dual frame to be $\widetilde{\boldsymbol{\Phi}}$. Note that the error signal recast in matrix notation is $(\boldsymbol{x} - \widehat{\boldsymbol{x}}) = \widetilde{\boldsymbol{\Phi}}^t \boldsymbol{p}$, where $\boldsymbol{p}$ is the perturbation vector $\boldsymbol{p} = [p_1, p_2, \ldots, p_K]^t$. Linear algebra can yield a bound on the error,

$$\left\| \widetilde{\boldsymbol{\Phi}}^t \boldsymbol{p} \right\|^2 = \left| \langle \boldsymbol{p}, \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^t \boldsymbol{p} \rangle \right| \leq \left\| \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^t \right\| \cdot \|\boldsymbol{p}\|^2 .$$

Note that because the singular values of $\widetilde{\boldsymbol{\Phi}}$ are the square roots of the eigenvalues of both $\left(\widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^t\right)$ and $\left(\widetilde{\boldsymbol{\Phi}}^t \widetilde{\boldsymbol{\Phi}}\right)$, it follows that $\left\| \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^t \right\| = \left\| \widetilde{\boldsymbol{\Phi}}^t \widetilde{\boldsymbol{\Phi}} \right\|$. Because the dual set is a frame for $\mathcal{H}$ with upper frame bound $\left(\frac{1}{A}\right)$ and because of the relationship between the eigenvalues of $\left(\widetilde{\boldsymbol{\Phi}}^t \widetilde{\boldsymbol{\Phi}}\right)$ and the frame bounds, we can finally write a useful bound (alluded to in [5]) on the reconstruction error

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2 \leq \frac{\|\boldsymbol{p}\|^2}{A}. \tag{2.2}$$

In words, the perturbation energy is reduced in the reconstruction by at least the minimum redundancy in the set of frame analysis vectors $\{\boldsymbol{\phi}_k\}$. The upper bound in (2.2) is consistent with probabilistic robustness results when stochastic noise is added to frame coefficients [73].

2.1.5 Signal dictionaries

Many different signal dictionaries have significance in signal processing. Perhaps the most fundamental dictionary is the canonical basis, $\{\boldsymbol{e}_k\}$. Each element in this ONB has a single non-zero entry, $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^t$, $\boldsymbol{e}_2 = [0, 1, 0, \ldots, 0]^t$, $\ldots$ $\boldsymbol{e}_K = [0, \ldots, 0, 1]^t$. Though not an overcomplete system, the canonical basis arises as a useful analytic tool in a number of settings.

A number of more specialized dictionaries are also commonly used in signal processing applications. The family of Fourier transforms (including continuous and discrete domains of time and frequency) can be thought of as a set of dictionaries consisting of complex exponentials. Similarly, the family of wavelet transforms are a set of dictionaries consisting of localized oscillating functions with various locations and scales. Each of these dictionaries are well-matched to specific classes of signals that can be represented with relatively few dictionary elements. For example, Fourier dictionaries are well-matched to "harmonic" signals consisting of strong periodic components. In contrast, wavelet dictionaries are well-matched to signals consisting of smooth regions connected by abrupt transitions (i.e., piecewise polynomial signals).

A number of other dictionaries are commonly used in signal and image processing applications. Various representation systems have been developed by altering the space-frequency (or time-frequency) partitions underlying the dictionary atoms, including wavelet packets [24], complex wavelets [89, 139], windowed Gaussian (i.e., Gabor) atoms [123], steerable pyramids [141, 142], curvelets [12] and contourlets [46]. Several other dictionaries have been designed by simply "tiling" a single atom over various parts of the domain (generally 2-D dyadic squares over the spatial domain). Examples of such tilings include wedgelets [48] (piecewise constant atoms), platelets [154] (piecewise planar) and surflets [18] (piecewise constant with polynomial discontinuities). Finally, other researchers have taken to building complex dictionaries by simply combining dictionaries that individually have very different characteristics [18, 151].

2.2 Neuroscience Preliminaries

2.2.1 The neurobiology of single neurons

Sensory neural systems are built using connected networks of single neurons. While the structure of a neuron varies throughout a single system, typical neurons have three distinct components: an input region, an integrative region, and an output path, seen in Figure 2.2. The input region is largely made up of a *dendritic tree*. The cell body, or *soma* contains the biological components common to all cells, housing the machinery necessary to maintain cellular function. The soma also typically serves as the neuron's integrating region, combining inputs initiated in distinct portions of the dendritic tree. The *axon* is the major extension projecting away from the soma, carrying output signals to other neurons. The connection from the axon to another

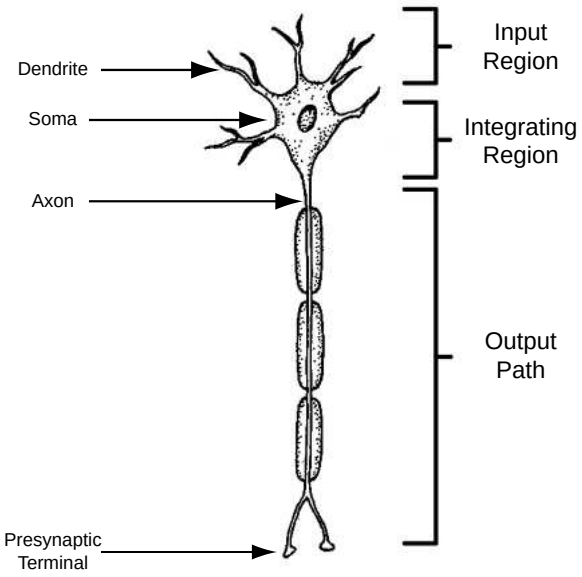


Figure 2.2: Basic anatomy of a general neuron (adapted from [87]). Typical neurons have three distinct regions: an input region consisting of a dendritic tree, an integration region in the cell body (soma) and the output along the axon. The axon terminates in a presynaptic terminal that forms a synapse with an adjacent neuron.

neuron's dendritic tree is called a *synapse*. The synapse consists of the *presynaptic terminal* (a specialized area on the sending neuron) and a *postsynaptic terminal* (a specialized area on the receiving neuron) [87, 106]. A typical neuron has many dendrites but only one axon.

Neurons represent signals by an electrochemical potential across the cell membrane. The intracellular and extracellular solutions are electrically neutral, so the electrical potential is determined by the presence (or absence) of charged ions. The four most significant ions in neurophysiology are sodium (Na^+), potassium (K^+), calcium (Ca^+) and chloride (Cl^+). The cell membrane is a two-layer lipid structure that is virtually impermeable to these ions [106]. When there is a potential difference between the intracellular and extracellular ionic concentrations, the cell membrane separates and stores ionic charge the same way capacitors function with free electrons.

The cell membrane is embedded with ion channels that are often selective for a certain ion species. These channels can vary their transmission rate depending on several stimulating factors, including the magnitude of the membrane potential or the presence of a specific chemical agent [87,106]. In electrical terms, the ion channels are modeled as resistors with a specific conductance (the reciprocal of resistance) being driven by an electrochemical gradient.

In a typical model of a neuron, presynaptic activity induces a membrane potential change in a dendrite (called an excitatory postsynaptic potential, or EPSP). The resulting current travels down the electrical gradient from the dendrite to the soma and is integrated with inputs from other dendrites [87, 106].

In most neurons, the analog potentials in the soma act as an input for the stochastic production of *action potentials*, or spikes in the membrane potential. These spike waveforms actively propagate down the axon to act as an output signal for the neuron. Spike waveforms are generally stereotyped, leaving only event timing as the significant feature. Spike trains are frequently modeled mathematically as *point processes* [84], stochastic processes completely defined by event timing [28, 143]. When action potentials reach the presynaptic terminal, a chemical neurotransmitter is released into the extracellular space to generate an EPSP on a postsynaptic neuron.

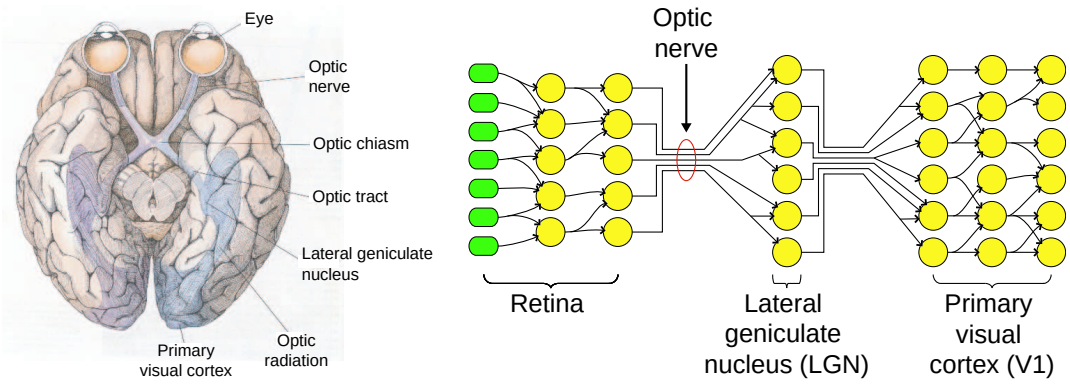
2.2.2 The early visual pathway

While our goal is to understand general information representation principles used by sensory neural systems, we will phrase most of our discussion in terms of the human visual system. Though the operation of the systems is very different, many of the same general principles can be seen (using appropriate abstractions) in the auditory system. Our overview of the early visual pathway will be exceedingly brief and providing only a very high level description of the system. A general introductory text such as [81] provides a more detailed exposition.

Anatomically, sensory systems appear to be organized using stages (or layers) of neural populations. The collective activity of the entire population of neurons in a single stage represents the sensory information about the external stimulus. In moving from one stage to the next, a single neuron generally projects its output to the dendritic tree of many neurons in the succeeding stage. While there are typically many different types of cells at each stage (e.g., rod and cone type photoreceptors in the retina), we will ignore these distinctions to illustrate the broad function of the stages.

The first stage of the human visual pathway is a collection of photoreceptors that transduce the retinal image into electrochemical signals. Each photoreceptor essentially integrates the light activity over a spatial region. We can view the activity in the photoreceptor population as a (non-uniformly) sampled representation of the retinal image. The photoreceptor outputs excite a layer of bi-polar cells, which in turn project to a layer of retinal ganglion cells. The axons of the retinal ganglion cells form the optic nerve, principally projecting to a region of the thalamus called the lateral geniculate nucleus (LGN). The output of the LGN cells serve as the principal input to the primary visual cortex (V1). The outputs of V1 appear to project to several different areas of the visual cortex, forming parallel streams that have distinct functions in higher-level processing (e.g., the ventral stream and the dorsal stream).

One way to understand the function of cells in a particular layer is to record the cell activity while presenting the system with simple stimuli (e.g., small points of light). By making systematic stimulus variations and recording the cell activity, one can map out the form of the stimulus that causes the cell to respond. This type of experiment leads to the notion of a *receptive field*, a description of the canonical stimulus that makes the cell respond most strongly. In the LGN, these receptive fields



(a) (b)

Figure 2.3: Anatomic and functional diagrams illustrating the early visual pathway in humans. Both figures are adapted from [81]. (a) The major components of the early visual pathway as arranged anatomically in the human. (b) The functional connections of the early visual pathway. Populations of neurons are grouped together in stages. Each neuron in one stage typically projects to many nerves in the succeeding stage.

have a center-surround structure, responding most strongly to a bright spot of light surrounded by a darker contrasting ring (or conversely, a dark center followed by a bright contrasting ring). In V1, one type of cell (called a *simple cell*) has a receptive field resembling a Gabor atom [86]. These receptive fields match well with edges in the image, having a preferential spatial location, spatial bandwidth and orientation. Though we will not discuss them in any detail, the other primary type of cell in V1 are called *complex cells*. These cells also responds to oriented edges, but may respond regardless of the exact location of the edge within the receptive field or may only respond to moving stimuli.

The notion of a receptive field essentially implies a linear-filtering operation in space and time to represent the stimulus.[3] In the terminology of the abstract sensing system described in Section 2.1.2, a commonly accepted model of the early visual pathway views each layer as applying a set of linear filters to the outputs of the previous stage. Much like the overcomplete dictionaries described by frame theory, each stage past the retina appears to be significantly more overcomplete than the previous stage. For example, one estimate of the ratio of V1 outputs to V1 inputs (in cat) yields a 25:1 ratio [112].

[3]Common models of these cells also include a nonlinearity. However, the "standard model" that has emerged to describe V1 function typically views this nonlinearity as a normalization by the responses of neighboring units and a pointwise rectification and compression of the dynamic range [115]. This type of nonlinearity would not affect the information content of the signals being transmitted and so will not be considered here.

Chapter 3
Relying on Redundancy for Robust Representations: Noise Reduction in Frames and Fusion Frames

In many signal processing applications, robustness is achieved through introducing redundancy into the signal representation. For example, it is well-known that when frame coefficients are corrupted with additive noise, reconstruction using the canonical dual vectors reduces the MSE in proportion to the minimum frame bound [99]. However, it is not always the case that redundancy is a good use of system resources, even when there are signal imperfections. In order to optimize resource allocation, it is important to understand the various robustness gains achieved by increasing redundancy in centralized and distributed systems.

3.1 Corruption in Distributed Sensing Systems

Assessing the robustness properties of redundant representations can be complicated for several reasons. First, additive noise is not always a good model for processes that induce coefficient corruption. In particular, sensory neural systems communicate through action potentials that are often modeled as point process representations [84]. Extrapolating results from additive noise sources to point processes can lead to incorrect results [132]. In order to understand the potential noise reduction ability available to a neural system employing redundant population codes, we must quantify and extend the known results from additive noise sources to include point process modulation.

Second, the known noise reduction abilities of a redundant representation assume a centralized reconstruction scheme where all of the coefficients are collected at a single location. There are important and interesting examples of systems that have redundant total representations but are limited to distributed processing environments where information must be processed locally before it is globally aggregated. One example of such a system is a sensor network, where sensor proximity can produce redundant observations but constraints prohibit centralized processing. Another example is sensory neural systems, where sensory information is represented by collections of neural populations represented different features of the stimulus [133]. It is important to understand how this distributed processing requirement affects the noise-reduction ability of the system.

Using frame theory to model an abstract sensing system, we extend the known noise reduction properties of redundant representations to Poisson process noise model commonly used for neural systems. We also determine the potential impact one type of distributed processing has on the noise reduction properties of a redundant representation. We employ an extension of frame theory called "fusion frames" [16,17] that links the properties of a global frame (centralized structures) and local collections of frame elements (distributed structures). Using this tool, we find conditions under which there is no loss in noise reduction ability, and we bound the performance loss for more general cases.

3.2 Background and Related Work

3.2.1 Poisson counting processes

We consider the encoding of a signal vector $\boldsymbol{x} \in \mathbb{R}^N$ in a frame through the collection of measurements $m_k = \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle, k = 1, \ldots, K$. We assume that these projections have bounded magnitudes, $|m_k| \leq \Delta, \forall k$. These analog coefficients would require infinite precision to store and transmit in their current state. The most common model for the inevitable corruption caused by finite precision storage or transmission is additive noise. However, in some systems (including neural systems) the uncertainty comes from modulating the signal onto a point process.

The simplest starting place for modeling point process communication is to use a Poisson counting process. To ensure that we have non-negative rates for the Poisson encoding, we first define a simple invertible transformation to make the coefficients non-negative, $h(m_k) = m_k + \Delta$. To simplify notation, we denote the coefficients that will be transmitted (sent) as $m_k^s = h(m_k)$ over a common time interval.[1] These coefficients will be transmitted over a noisy medium to generate received coefficients $\{m_k^r\}$. In a Poisson counting process, the received coefficients are independent and identically distributed according to the Poisson distribution,

$$P(m_k^r | m_k^s) = \frac{e^{-m_k^s} \left(m_k^s\right)^{m_k^r}}{m_k^r!}.$$

The received coefficients are non-negative integers, with an expected value and variance equal to the sent coefficient

$$\mathcal{E}\left[m_k^r\right] = \left(\mathcal{E}\left[(m_k^r)^2\right] - \mathcal{E}\left[m_k^r\right]^2\right) = m_k^s.$$

3.2.2 Fusion frames

Though a frame certainly provides some robustness to reconstruction error (see Section 3.2.3), the underlying assumption when reconstructing from frame coefficients

[1] The length of the time interval is not critical for our purposes, so we implicity assumed it is 1.

is that those coefficients can be centrally collected in one place. In many settings with distributed processing constraints, centrally collecting the frame coefficients is either impossible or very undesirable. Instead, it is likely that subcollections of coefficients will have to be aggregated locally before centralization. In other words, signal reconstruction will be first performed in a subspace of the signal space before those subspace reconstructions are aggregated for the final (full) reconstruction. To begin analyzing a reconstruction from redundant elements in this distributed way, we turn to a new theory of "fusion frames" [16,17].[2] Here, signals are decomposed in terms of overlapping subspaces that are each locally spanned by a collection of frame elements.

Specifically, a family of closed subspaces $\{W_i\}_{i=1}^{L}$ is a *fusion frame* for the vector space $\mathcal{H}$ if for every signal $\boldsymbol{x} \in \mathcal{H}$,

$$A^s \|\boldsymbol{x}\|^2 \leq \sum\nolimits_i \|\pi_i(\boldsymbol{x})\|^2 \leq B^s \|\boldsymbol{x}\|^2, \tag{3.1}$$

where $0 < A^s \leq B^s < \infty$ are the fusion frame bounds and $\pi_i(\cdot)$ is the orthogonal projection onto W_i. From (3.1) it is clear that the fusion frame bounds are themselves bounded by the number of subspaces, $L \geq B^s$. A fusion frame is in many ways analogous to a frame. In frame theory, an input signal is represented by a collection of scalar coefficients that measure the projection of that signal onto each frame vector. In a fusion frame, an input signal is represented by a collection of *vector* "coefficients" that represent the projection (not just the projection energy) onto the each subspace. Formally, for a fusion frame, the representation space is defined as $\mathcal{V} = \{\{\boldsymbol{x}_i\} | \boldsymbol{x}_i \in W_i\}$, the space of all collections of vectors containing one representative from each subspace.[3]

Analogous to frame theory, a fusion frame has an analysis operator, $\mathbf{W} : \mathcal{H} \to \mathcal{V}$, $\mathbf{W}\boldsymbol{x} = \{\pi_i(\boldsymbol{x})\} = \{\boldsymbol{x}_i\}$. The adjoint of the analysis operator is the synthesis operator, $\mathbf{W}^* : \mathcal{V} \to \mathcal{H}$, $\mathbf{W}^*\{\boldsymbol{x}_i\} = \sum_i \boldsymbol{x}_i$. The composition of the analysis and synthesis operators simply project the signal onto the overlapping subspaces and then take the linear combination of these projections, $(\mathbf{W}^*\mathbf{W}) : \mathcal{H} \to \mathcal{H}$, $(\mathbf{W}^*\mathbf{W})\,\boldsymbol{x} = \sum_i \pi_i(\boldsymbol{x})$. This composite operator is bounded and invertible, with bounds $A^s \leq \|(\mathbf{W}^*\mathbf{W})\,\boldsymbol{x}\| \leq B^s$ and $\frac{1}{B^s} \leq \left\|(\mathbf{W}^*\mathbf{W})^{-1}\,\boldsymbol{x}\right\| \leq \frac{1}{A^s}$ for all $\|\boldsymbol{x}\| = 1$. As in section 2.1.3, the unique pseudoinverse for $\mathbf{W}$ is given by $\mathbf{W}^\dagger = (\mathbf{W}^*\mathbf{W})^{-1}\,\mathbf{W}^*$.

Fusion frames are a powerful tool because one of the fundamental results of [16] shows that there is a link between the properties of a global frame and local collections of those frame elements. If each subspace W_i has an associated family of vectors that locally form a frame for W_i with bounds (A_i, B_i), then the total collection of vectors globally form a frame for $\mathcal{H}$. It can be shown that the frame bounds (A, B) for this

[2] Fusion frames also alternately go by the name "frames of subspaces".

[3] The fusion frames theory also extends to infinite dimensional spaces by incorporating the usual finite-energy restriction on the vectors $\{\boldsymbol{x}_i\}$. This explicit restriction is not necessary in our finite-dimensional setting where all vectors have finite energy.

global frame are bounded by $A^s A_{\min} \leq A$ and $B \leq B^s B_{\max}$, with equality when the same frame bounds apply to each local frame, $A_i = A_{\min}$ and $B_i = B_{\max}, \forall i$.

Because of the link between frames for a local subspace and the global frame of elements taken together, we can use fusion frames to represent distributed processing [130, 131]. In this model, the frame vectors local to one subspace are used to reconstruct the orthogonal projection of the signal into that subspace while maximizing the noise reduction properties of the local frame. The collection of reconstructed signals within each subspace are then used to reconstruct the original input signal while maximizing the noise reduction properties of the fusion frame. This subspace-based reconstruction distributes the processing by only requiring knowledge of the frame vectors within a subspace to do a local reconstruction. Such a scheme adds a level of robustness to the system because only vectors in one local subspace are affected if a frame vector is added or removed.

3.2.3 Additive noise reduction in a frame

In a frame decomposition (tight or non-tight), it is well known that the pseudoinverse operator $\mathbf{\Phi}^\dagger$ is optimal among all linear operators at reducing the reconstruction MSE when the coefficients are corrupted with additive noise [99]. The amount of noise power eliminated is related to the amount of redundancy in the elements, which is reflected in the frame bounds [74]. Section 2.1.4 quantified the effect of a general perturbation in a frame. Here we consider the specific case of additive stochastic noise. While these results could follow immediately from the results of Section 2.1.4, the literature on this topic normally takes a slightly different approach than the techniques used in Section 2.1.4.

Let $\{\boldsymbol{\phi}_k\}_{k=1}^K$ be a frame with bounds (A, B) for an N-dimensional input space $\mathcal{H}$. Reconstruction coefficients are corrupted by uncorrelated noise n_k with variance σ^2,

$$\widehat{\boldsymbol{x}} = \sum_{k=1}^{K} \left[\langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle + n_k\right] \widetilde{\boldsymbol{\phi}}_k.$$

The total MSE of the reconstructed vector $\widehat{\boldsymbol{x}}$ is given by

$$\mathcal{E}\left[\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|^2\right] = \sigma^2 \sum_{k=1}^{K} \left\|\widetilde{\boldsymbol{\phi}}_k\right\|^2 = \sigma^2 \sum_{n=1}^{N} \frac{1}{\eta_n}, \tag{3.2}$$

where $\{\eta_n\}_{n=1}^N$ are the eigenvalues of the composite operator $\left(\mathbf{\Phi}^t \mathbf{\Phi}\right)$ [20, 34, 74, 99]. The MSE per signal dimension is bounded by

$$\frac{\sigma^2}{B} \leq \frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|^2\right]}{N} \leq \frac{\sigma^2}{A}. \tag{3.3}$$

Note that in reconstructing with the redundant expansion, the noise is reduced by

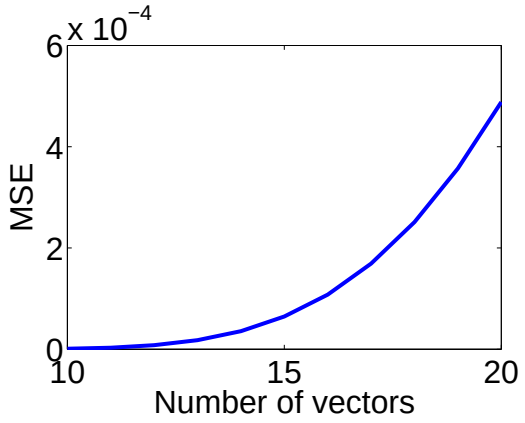


Figure 3.1: Reconstruction MSE increases with the number of vectors in a resource constrained system. The input space is taken to be $\mathbb{R}^{10}$ and we vary the number of vectors (K) in the tight frame from 10 to 20. When there are 100 total bits that used to transmit the coefficients, the total MSE (accounting for the quantization noise and the frame reconstruction) is $\frac{K}{N}2^{-\frac{100}{K}}$. The increased quantization noise overwhelms the increased noise-reduction ability of the frame as the number of vectors is increased. In this resource constrained system, an ONB ($K = 10$) is optimal.

an amount at least proportional to the minimum redundancy. It is clear from the derivation of equation (3.3) that the bounds are not tight. The MSE will only achieve the bounds in (3.3) in the case of a tight frame when the bounds are equal.

The noise reduction result in (3.3) appears to indicate that increasing the redundancy in the frame (i.e., increasing A) will improve system performance by lowering the reconstruction MSE. However, this result can be deceptive because it does not include the resource cost of increasing the redundancy. For example, consider a tight frame with K vectors ($A = \frac{K}{N}$) that is limited in the total number of bits β it can use to communicate all of the coefficients. Assume that each of the K coefficients is communicated with $\frac{\beta}{K}$ bits. Assuming the common additive noise approximation for the effects of uniform scalar quantization, the variance of the received coefficients is $2^{-\frac{2\beta}{K}}$. Increasing the number of vectors (K) in the system would increase the redundancy (and therefore have stronger noise reduction abilities), but would also increase the noise power on each coefficient by decreasing the bit rate assigned to each frame vector. This effect is illustrated in a simple example shown in Figure 3.1. In this case, the increased quantization noise from adding more vectors (and thereby spreading the available bits across more coefficients) overwhelms the increased noise reduction from increasing the frame bounds. Consequently, the optimal representation in this resource constrained setting would be an ONB.

3.3 Point Process Noise Reduction in a Frame

To understand the potential robustness benefits available to neural systems using redundant representations, we must consider the impact of point process noise sources on frame reconstruction. While we may expect similar results to the standard additive-noise analysis presented in Section 2.1.4, point process encodings can exhibit non-intuitive behavior. Therefore, we must extend the know results to include this class of corruption that more closely models neural encodings.

As in Section 3.2.1, we assume a signal vector $\boldsymbol{x} \in \mathbb{R}^N$ is encoded through bounded projections onto the frame vectors, $m_k = \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle$, $|m_k| \leq \Delta, \forall k$. To ensure positive firing rates, the coefficients are shifted via the function $m_k^s = h(m_k) = m_k + \Delta$. These coefficients represent the intensity function of a Poisson counting process, with the received coefficients denoted m_k^r.

The received coefficients are then used to estimate the original coefficients by inverting the shifting function, $\widehat{m}_k = h^{-1}(m_k^r) = m_k^r - \Delta$. The estimated signal using the received coefficients is given by

$$\widehat{\boldsymbol{x}} = \sum_{k=1}^{K} \widehat{m}_k \widetilde{\boldsymbol{\phi}}_k.$$

Interpreting the coefficient perturbation as $(\widehat{m}_k - m_k)$, the results of Section 2.1.4 quantify the reconstruction MSE for this system,

$$\mathcal{E}\left[\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2\right] \leq \frac{\mathcal{E}\left[\sum_{k=1}^{K} \left(\widehat{m}_k - m_k\right)^2\right]}{A}.$$

Because each coefficient is assumed to be encoded independently, the problem reduces to calculating the second moment of the reconstructed coefficients $\mathcal{E}\left[\left(\widehat{m}_k - m_k\right)^2\right]$. Remembering from Section 3.2.1 that the variance of a Poisson random variable is equal to the mean, we calculate the expected value of one coefficient perturbation to be

$$\begin{aligned}
\mathcal{E}\left[\left(\widehat{m}_k - m_k\right)^2\right] &= m_k^2 - 2m_k^2 + \mathcal{E}\left[\widehat{m}_k^2\right] \\
&= \mathcal{E}\left[\left(m_k^r - \Delta\right)^2\right] - m_k^2 \\
&= \mathcal{E}\left[\left(m_k^r\right)^2\right] - 2\left(m_k + \Delta\right)\Delta + \Delta^2 - m_k^2 \\
&= \left(m_k + \Delta\right) + \left(m_k + \Delta\right)^2 - 2\left(m_k + \Delta\right)\Delta + \Delta^2 - m_k^2 \\
&= \left(m_k + \Delta\right).
\end{aligned}$$

Therefore, the reconstruction MSE under Poisson counting process modulation is

$$\mathcal{E}\left[\|\boldsymbol{x}-\widehat{\boldsymbol{x}}\|^2\right] \leq \frac{\sum_{k=1}^{K}(m_k+\Delta)}{A}. \tag{3.4}$$

As expected, the result in (3.4) is reminiscent of the noise reduction formula for additive noise given in Section 2.1.4. The primary difference is that the total noise variance is now signal dependent, scaling with the size of the projections onto the frame vectors. This property follows because in a Poisson distribution the noise variance scales with the mean. This result may indicate that higher firing rates may hamper the reconstruction fidelity for a system communicating via point processes.

The result in (3.4) also clearly depends on the specific form of the shifting function $h(\cdot)$ to achieve positive firing rates. This function was chosen for its simplicity, and it is not clear at this point how other choices of $h(\cdot)$ would affect the results.

3.4 Noise Reduction in a Fusion Frame

In a distributed processing setting, local groups of frame vectors (i.e., groups of vectors not spanning the whole signal space) will be used to partially reconstruct the signal before these local representations are combined at a centralized location. Each local collection of vectors forms a frame for a subspace of the signal space. The local signal reconstruction using just those vectors is equivalent to the projection of the signal into this subspace. These subspace projections are then combined at a global location to estimate the original signal. There are two redundancies that must be accounted for in this setting: the redundancy of the frame vectors spanning each individual subspace, and the redundancy between the subspaces.

Section 3.2.3 investigates the noise reduction properties inherent in the redundancy between frame vectors. These results quantify the noise reduction *within* a subspace reconstruction. To completely quantify the noise reduction possible in this distributed processing setting, we must also determine the noise reduction properties due to the overlap *between* the subspaces. In other words, we need to extend the noise reduction results known in frame theory to the theory of fusion frames. Section 3.4.1 calculates the noise reduction properties of overlapping subspaces, and Section 3.4.2 combines these results with the results of Section 3.2.3 to characterize the distributed processing setting.

3.4.1 Subspace noise reduction

Let $\{W_i\}_{i=1}^{L}$ be a fusion frame for the vector space $\mathbb{R}^N$ with fusion frame bounds (A^s, B^s). The "coefficients" in this decomposition are the collection of vectors $\{\boldsymbol{x}_i\} \in \mathcal{V}$, where each vector is the projection of the input signal $\boldsymbol{x}$ onto a subspace, $\boldsymbol{x}_i = \pi_i(\boldsymbol{x})$. The total collection of subspace projections is given by the fusion frame operator, $\{\boldsymbol{x}_i\} = \mathbf{W}\boldsymbol{x}$. Consider the case when the vector coefficients are corrupted

independently with a noise vector,

$$\widehat{\boldsymbol{x}} = \mathbf{W}^{\dagger}\{(\boldsymbol{x}_i + \boldsymbol{n}_i)\} = (\mathbf{W}^{*}\mathbf{W})^{-1} \sum_i (\boldsymbol{x}_i + \boldsymbol{n}_i)\,,$$

where the vector $\boldsymbol{n}_i \in W_i$ has covariance matrix $\boldsymbol{\Gamma}_i$. The linearity of $\mathbf{W}^{\dagger}$ implies that $(\widehat{\boldsymbol{x}} - \boldsymbol{x}) = (\mathbf{W}^{*}\mathbf{W})^{-1} \sum_i \boldsymbol{n}_i$. Define $\widetilde{\boldsymbol{n}} = \sum_i \boldsymbol{n}_i$, so that $\widetilde{\boldsymbol{n}}$ has covariance matrix $\widetilde{\boldsymbol{\Gamma}} = \sum_i \boldsymbol{\Gamma}_i$. Because multiplication affects the covariance in a quadratic way, the covariance of $(\mathbf{W}^{*}\mathbf{W})^{-1}\,\widetilde{\boldsymbol{n}}$ is given by $(\mathbf{W}^{*}\mathbf{W})^{-1}\,\widetilde{\boldsymbol{\Gamma}}\,(\mathbf{W}^{*}\mathbf{W})^{-1}$. The total MSE of the reconstructed signal therefore equals the trace of the covariance,

$$\mathcal{E}\left[\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|^2\right] = \mathrm{Tr}\left[(\mathbf{W}^{*}\mathbf{W})^{-1}\,\widetilde{\boldsymbol{\Gamma}}\,(\mathbf{W}^{*}\mathbf{W})^{-1}\right]. \tag{3.5}$$

As mentioned earlier, $(\mathbf{W}^{*}\mathbf{W})^{-1}$ is a bounded operator with bounds $\left(\frac{1}{B^s}, \frac{1}{A^s}\right)$.[4] Appendix A calculates upper and lower bounds on the trace of a quadratic form that directly yields bounds on the MSE per signal dimension,

$$\frac{\mathrm{Tr}\left[\widetilde{\boldsymbol{\Gamma}}\right]}{N\,(B^s)^2} \leq \frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|^2\right]}{N} \leq \frac{\mathrm{Tr}\left[\widetilde{\boldsymbol{\Gamma}}\right]}{N\,(A^s)^2}. \tag{3.6}$$

Notice that the natural reconstruction for a fusion frame also reduces the noise in the reconstructed signal by an amount that depends on the minimum redundancy. As before, the bounds in (3.6) are not tight bounds. The MSE will only achieve the extreme bounds in (3.6) when the frame of subspaces is tight ($A^s = B^s$).

3.4.2 Noise reduction under distributed processing

We now have the tools available to consider the noise reduction capability of a redundant expansion under distributed processing requirements. Let $\{W_i\}_{i=1}^{L}$ be a fusion frame for the vector space $\mathcal{H}$, with frame bounds (A^s, B^s). Let each subspace be spanned by a collection of K_i vectors that locally form a frame for W_i with frame bounds (A_i, B_i). When taken together, these vectors form a frame for $\mathcal{H}$ with bounds (A, B). A signal $\boldsymbol{x}$ is represented by all of the vectors in the global frame, and coefficients are corrupted by additive noise with subspace-dependent variance σ_i^2.

For distributed reconstruction, local frame vectors are first used to reconstruct the projection of the signal onto each subspace, $\widehat{\boldsymbol{x}}_i$. From equation (3.3), we have a bound on the total MSE when reconstructing each $\boldsymbol{x}_i = \pi_i(\boldsymbol{x})$,

$$\frac{\sigma_i^2 N}{B_i} \leq \mathcal{E}\left[\|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2\right] \leq \frac{\sigma_i^2 N}{A_i}.$$

[4]Note that the "bounds" of a linear operator are equivalent to the minimum and maximum eigenvalues of that operator.

Distributed reconstruction $\widehat{\boldsymbol{x}}^d$ is completed by using each $\widehat{\boldsymbol{x}}_i$ as a corrupted coefficient in the fusion frame, $\widehat{\boldsymbol{x}}^d = \mathbf{W}^\dagger\{\widehat{\boldsymbol{x}}_i\}$. The previous bound tells us that we can view the subspace projections $\boldsymbol{x}_i$ as being corrupted independently by an additive noise vector $\boldsymbol{n}_i \in W_i$, with covariance matrix $\boldsymbol{\Gamma}_i$ that has bounded total variance $\frac{\sigma_i^2 N}{B_i} \leq \operatorname{Tr}[\boldsymbol{\Gamma}_i] \leq \frac{\sigma_i^2 N}{A_i}$. If we define $\widetilde{\boldsymbol{\Gamma}} = \sum_i \boldsymbol{\Gamma}_i$, applying equation (3.6) upper bounds the MSE per signal dimension,

$$\begin{aligned}\frac{\mathcal{E}\left[\left\|\widehat{\boldsymbol{x}}^d - \boldsymbol{x}\right\|^2\right]}{N} &\leq \frac{\operatorname{Tr}\left[\widetilde{\boldsymbol{\Gamma}}\right]}{N\left(A^s\right)^2} \\ &\leq \frac{1}{\left(A^s\right)^2}\sum_{i=1}^{L}\frac{\sigma_i^2}{A_i} \\ &\leq \frac{L\sigma_{\max}^2}{A_{\min}\left(A^s\right)^2}. \end{aligned} \tag{3.7}$$

Using the same technique in the opposite direction we can also write a lower bound on the MSE per signal dimension

$$\frac{\mathcal{E}\left[\left\|\widehat{\boldsymbol{x}}^d - \boldsymbol{x}\right\|^2\right]}{N} \geq \frac{L\sigma_{\min}^2}{B_{\max}\left(B^s\right)^2}.$$

Frame reconstruction with our distributed processing constraint has the power to reduce noise in the reconstructed signal by an amount that depends on the minimum redundancy of both the fusion frame, and the individual local frames that span the subspaces. From the derivation of this bound, it is clear that these bounds are also not tight and are only achieved in the special case when the fusion frame is tight ($A^s = B^s$) and the local frames are all tight with the same bounds ($A_i = A_{\min} = B_i = B_{\max}, \forall i$).

3.4.3 Comparison of centralized and distributed processing bounds

Section 3.2.3 mentions that the pseudoinverse operator for a frame is the unique linear operator that achieves maximum noise reduction. Consequently, we know that no other linear reconstruction scheme (including the distributed processing of section 3.4.2) can perform better than the centralized processing scheme that uses the pseudoinverse operator for the global frame (i.e., all frame vectors considered together at once). In fact, the distributed reconstruction operator is optimal *only* over linear operators that factor (using local frame analysis and synthesis operators $\boldsymbol{\Phi}_i$ and $\boldsymbol{\Phi}_i^t$) into

$$\mathbf{W}^\dagger = \left[\begin{array}{ccc} \left(\mathbf{W}^*\mathbf{W}\right)^{-1} & \ldots & \left(\mathbf{W}^*\mathbf{W}\right)^{-1} \end{array}\right] \left[\begin{array}{c} \left(\boldsymbol{\Phi}_1^t\boldsymbol{\Phi}_1\right)^{-1}\boldsymbol{\Phi}_1^t \\ \vdots \\ \left(\boldsymbol{\Phi}_L^t\boldsymbol{\Phi}_L\right)^{-1}\boldsymbol{\Phi}_L^t \end{array}\right].$$

We must quantify the penalty in noise reduction a distributed processing scheme incurs compared to centralized processing. From section 3.2.3, considering the collection of local frames together yields a global frame for $\mathcal{H}$ with lower and upper bounds $A \geq A^s A_{\min}$ and $B \leq B^s B_{\max}$. Using a simple extension of equation (3.3) (with unequal noise power) tells us that a centralized reconstruction of $\boldsymbol{x}$ using the global frame directly, $\widehat{\boldsymbol{x}}^c$, would yield an upper bound on the MSE per signal dimension of

$$\frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}}^c - \boldsymbol{x}\|^2\right]}{N} \leq \frac{\sigma^2_{\max}}{A} \leq \frac{\sigma^2_{\max}}{A^s A_{\min}}.$$

Comparing this bound to the MSE upper bound in (3.7) when using the distributed scheme, we see that (because $L \geq A^s$) the upper bound using the centralized approach is better than the distributed reconstruction by a factor of $\frac{L}{A^s}$. While these are not tight bounds on the error, they hint at the potential for the distributed reconstruction to perform worse than the centralized scheme and give a bound on the performance reduction.

It is also interesting to consider conditions under which the noise reduction ability is the same for distributed and centralized processing. Consider the case when all of the local frames have the same number of vectors $\left(K_i = \frac{K}{L}\right)$ and are tight frames with the same frame bounds, $A_i = A_{\min} = B_i = B_{\max}, \forall i$. In this case, the global frame has frame bounds $A = A^s A_i$ and $B = B^s B_i$. If the fusion frame is also tight ($A^s = B^s$), the global frame will additionally be tight with bounds $A = B = \frac{K}{N}$. If the noise has equal power in each subspace ($\sigma_i^2 = \sigma^2$), then we show in Appendix B that the MSE per signal dimension under both processing schemes is equal,

$$\frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}}^c - \boldsymbol{x}\|^2\right]}{N} = \frac{\mathcal{E}\left[\left\|\widehat{\boldsymbol{x}}^d - \boldsymbol{x}\right\|^2\right]}{N} = \frac{\sigma^2 N}{K}.$$

Though the conditions proposed here for equal noise reduction may seem restrictive, a result from [73] regarding random frames indicates that frames will become tight asymptotically as more random vectors are added. Therefore, systems where random vectors are randomly assigned to a local subspace will asymptotically meet the conditions for achieving the optimal (centralized) noise reduction.

3.4.4 Numerical example

Figure 3.2 compares the noise reduction properties of a centralized and distributed reconstruction in $\mathcal{H} = \mathbb{R}^3$. Frame vectors were generated at random (uniformly distributed in $\mathbb{R}^3$) and assigned to one of five possible subspaces. Centralized and distributed reconstructions using noisy coefficients ($\sigma_i^2 = 1$) are performed and the average reconstruction error is plotted. The centralized scheme always outperforms the decentralized scheme, though the performance tends to become similar as more vectors are added and the frames become tighter.

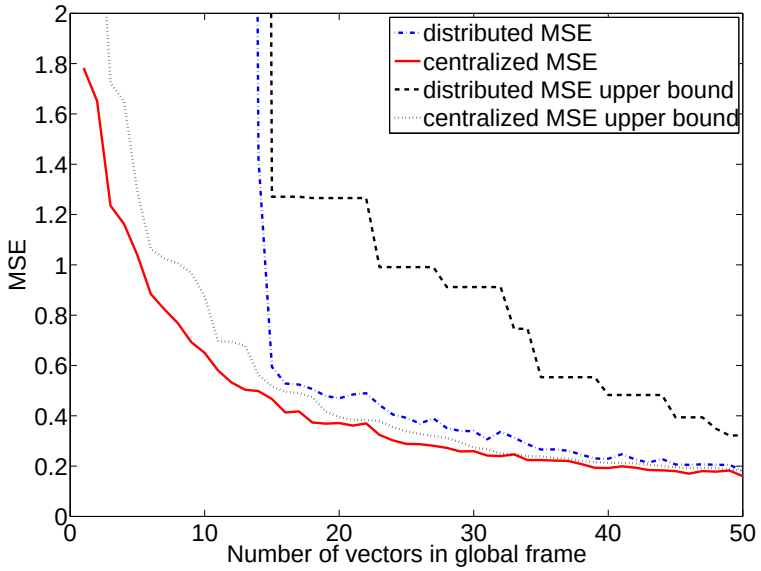


Figure 3.2: An example of the noise reduction properties for in centralized and distributed reconstruction. Vectors are randomly placed in $\mathbb{R}^3$ and separated into five subspaces. Noisy coefficients are used for both centralized reconstruction (using the global frame) and distributed reconstruction (using local reconstruction followed by fusion frame reconstruction).

3.5 Summary and Future Work

3.5.1 Summary of contributions

In redundant expansions, assessing the noise reduction capability of the representation is an important consideration. In this chapter, we have extended the known noise reduction results for additive noise to modulation onto a Poisson counting process. This model is more realistic than additive noise models for neural communication. We have shown that the Poisson process noise reduction in a frame parallels the known results for additive noise, with the complication that the noise variance is now signal dependent.

When distributed reconstruction is required by the system, assessing the noise reduction capability is further complicated. It would be intuitive to think that distributed reconstruction would always incur a penalty over the optimal centralized case. However, in this chapter we have shown at least one condition (there may be more) where the distributed system can be constructed to suffer no penalty compared to centralized reconstruction. Additionally, we provide bounds on the performance penalty in the more general case when distributed reconstruction is suboptimal.

3.5.2 Future directions

The results of this chapter could characterize the fundamental noise reduction potential of many distributed systems, including neural populations and distributed sensing systems. However, in many of these settings "noise" is only one type of failure typically encountered. Many systems composed of collections of simple nodes often encounter individual node failures. These failures can be thought of as removing a coefficient from the frame reconstruction (known as an "erasure"). The effect of erasures on a system is much more difficult to analyze. Previous work has started to investigate the effect of erasures, but only in the case of noiseless coefficients [79] and centralized reconstruction schemes [73]. We will pursue a characterization of the effects of erasures in noisy systems under distributed processing constraints.

Chapter 4
Managing Redundancy for Decentralized Action: Distributed Strategies for Wireless Sensor and Actuator Networks

One of the primary difficulties present when employing overcomplete representations is that they require coordination in either the encoding or decoding phase to account for the redundancy. This is illustrated in Section 2.1.3 by the fact that the typical reconstruction equation for an ONB does not work for a frame; reconstructing a signal from the canonical frame coefficients requires the use of a dual frame.

In a redundant sensing system, the overlap between adjacent sensor receptive fields must be accounted for before the data is used. When the goal is to induce action in the environment via a collection of redundant actuators, coordination is complicated even further. The simplest approach to achieving the necessary coordination is to have a centralized coordinator that collects all of the sensed data and makes the actuation decisions. However, in a distributed system data centralization is antithetical to the goal of using distributed data representations. In this chapter, we explore strategies for efficient distributed decision-making within the context of wireless sensor and actuator networks.

4.1 Wireless Sensor and Actuator Networks

Recent interest in wireless sensor networks (WSN) has led to increased research in many areas central to distributed data processing, including novel information processing, communications, and networking strategies [25, 57, 159]. Energy conservation to increase the functional life of the WSN is very important because the battery-powered sensors may be difficult (or impossible) to access on a large scale for maintenance. This energy conservation principle generally translates into minimizing the communication among sensors to preserve both individual node power and total network throughput. Basic research in communications and networking can improve the efficiency in transporting data from one location to another in the network. However, these aspects are limited by the underlying information processing strategies that determine the type and amount of data that needs to be transported within the network to achieve the application goal. Consequently, much of the recent sensor network research has focused on adapting well-known signal processing algorithms to distributed settings where individual sensor nodes perform local computations to minimize the

information that needs to be passed to more distant nodes (e.g., [9, 42, 108, 124]).

Many WSN algorithms start with the assumption that information contained in sensor measurements must be communicated to a destination (called a sink, or a fusion center) accessible by the WSN operator. In these applications, the sink represents the notion that the data collected in the WSN is only valuable to the extent that it can be removed from the network. However, in many applications the implicit assumption is that the information coming out of the network will be used to monitor the environment and *take action* (i.e., affecting the sensed environment) when necessary. One example of this is in agricultural irrigation. Soil moisture measurements are taken precisely so irrigation levels can be adjusted to improve growing conditions while conserving as much water as possible [102].

A significant and natural extension to the sensor network paradigm is a wireless sensor and actuator network (WSAN). A WSAN consists of a network of sensor nodes that can measure stimuli in the environment and a network of actuator nodes capable of affecting their local environment [1]. With direct communication from sensors to actuators, WSANs can achieve the application goals without aggregating the information in a single location or removing it from the network. In this way, WSANs can approach the robust and efficient ideal embodied in the mantra "the network is the processor". Using only local communications may become even more important in applications where the sensor nodes are deployed in conditions very hostile for wireless communications (e.g., under a layer of soil). Such hostile environments deteriorate the wireless channel and significantly reduce the realistic communication distances [2]. While WSAN applications open the door for in-network processing, optimal WSAN information processing strategies may be very different from the strategies developed for WSN applications. Sensor processing and communication strategies that blindly optimize sensor data fidelity may not yield the best results when actuation is involved. Information strategies in the WSAN must be designed with the final actuation performance fidelity in mind. While WSANs are often discussed as extending the WSN paradigm, quantitative analysis of their performance has received little attention.

The task of merging sensed information directly into actions efficiently without centralizing the information and decision making is difficult primarily because of the redundancy among the sensors and actuators. The coordination necessary to resolve this overlap must be built into the behavior of each individual node if no centralized controller will be used. Fortunately, distributed sensing and actuation occurs in biology where neural systems perform a chain of tasks very similar to the needs of WSANs: sensing, analysis, and response. Furthermore, evidence indicates that neural systems represent and process information in a distributed way (using groups of neurons) rather than centralizing the information and decision making in one single location. This shrewd strategy avoids creating a single point of vulnerability, so the system can function in the presence of isolated failures.

In neural systems, two types of behaviors exist, depending on whether there is "thinking" involved, which we call *conscious* and *reflex* behaviors. In conscious be-

havior biological systems gather sensory information, make inferences from that information about the structure of their environment, and generate actions based on that inferred structure. In reflex behavior, a sensed stimulus directly generates an involuntary and stereotyped action in the peripheral nervous system before the brain is even aware of the stimulus [87]. An obvious example of a reflex behavior is the knee-jerk reaction achieved by a doctor's well-placed tap below the kneecap. A more subtle example is the eye position correction that allows our vision to stay focused on an object even when our head is moving.

WSAN applications have an analogous division, which we call *object-based* and *measurement-based* network tasks. For example, the canonical target tracking scenario is an object-based task because it involves using sensory measurements to infer information about objects in the environment. On the other hand, an application such as agricultural irrigation is a measurement-based task because sensor measurements directly contain all the information necessary to take action — there is no underlying environmental object to try to infer. In this thesis we consider models of measurement-based WSAN applications. While measurement-based systems are simpler and possibly more limited than object-based systems, they provide an entry point for analyzing and designing WSAN algorithms.

This chapter presents a WSAN model based on the principles observed in a crayfish ocular reflex behavior. We show that linear control laws are exactly implementable using completely decentralized strategies (i.e., no centralized controller) even when the sensing and actuation systems are highly redundant. Extending the vector space models of the sensing process described in Section 2.1.2 to include the actuation process, we demonstrate that each sensor-actuator link has an associated importance value/ This metric can be used to efficiently determine which communication links are least important to the final actuation fidelity and can be eliminated to conserve power.

Finally, we consider the optimal power scheduling problem for this WSAN model in an inhomogeneous sensing environment. Optimal WSAN resource allocation requires us to consider not only the importance of a single sensor measurement value to the final actuation fidelity, but also the reliability of that measurement and the energy required to communicate it to its destination. These factors represent competing interests and must all be considered jointly when determining the optimal power allocation for each communication link in the WSAN application. Inspired by recent work for power scheduling in decentralized estimation [157], we propose and solve the optimal power scheduling problem for this WSAN model.

4.2 Background and Related Work

As an example reflex behavior that will shape our thinking about WSANs, we consider the crayfish visual system. The crayfish has a dorsal light reflex [105] where light movement in the visual field elicits a predictable reflex movement in the eyestalk

that attempts to keep a constant orientation of the visual field. The main visual representation is comprised of sensory neurons (called "sustaining fibers") that sum light activity in overlapping spatial regions. All of the information available to the creature about the light stimulus is contained in this collection of sustaining fiber responses [91, 92, 137, 153].

The crayfish eyestalk movement is controlled by a set of motorneurons that send signals to several small muscles. Each muscle generates movement in one specific direction. As with the sensory units, the muscle movement directions overlap in the movement space (i.e., muscle movements are not "orthogonal"). Most importantly, motorneuron activity is determined directly from a weighted combination of sustaining fiber inputs. Though all of the motorneurons have to be coordinated to produce the desired total action, their distributed individual responses are generated directly from the distributed sustaining fiber representation without a centralized decision-making structure. Previous research has shown that even in this critical behavior the contributions of each sensory unit to the total action are essentially linear [72].

Our WSAN model will follow the principles seen in this example from the crayfish. Though the constraints facing biological systems are different from the constraints imposed by wireless networking, neural systems must also be very resource efficient and try to minimize communication to conserve metabolic energy. Biological systems must have solutions that do a good job (some would even argue optimal) at trading-off performance and efficiency, and we use them as a rough guide.

In our model, a collection of sensors measuring overlapping spatial regions gather information about a stimulus field. A collection of actuators have individual environmental effects that overlap and must be coordinated. Each actuator determines its individual contribution to a behavioral goal through a weighted combination of the sensor measurements. The actuators do not communicate directly with one another but their behavioral decisions take into account what the other actuators in the network will be doing. In this simplest communication scenario with no inter-sensor and inter-actuator communication, we eliminate the communication overhead necessary for explicit cooperation. It may be possible to improve system performance by allowing additional communication and cooperation, depending on the specific networking model and communication costs involved.

A major goal in any WSAN information processing strategy is retaining good actuation performance while reducing the communication burden from the sensors to the actuators. To analyze the performance of a WSAN under different design decisions, we require a model of the physical sensing and actuation process. We extend the tools of frame theory to include the actuation process, giving us a set of mathematical tools to quantify the interaction of the WSAN with the environment.

4.2.1 Vector space sensor and actuator models

Sensor network models often begin with a collection of sensors distributed over a 2-D spatial field limited to the spatial domain Ω (e.g., $\Omega = [0, 1]^2$). Sensors are

indexed by $k \in \mathcal{K}$, and are located either irregularly or on a regular grid. The spatial region being sensed contains a stimulus field, denoted by $\boldsymbol{x}(\omega)$, where $\omega \in \Omega$ indicates location in the field. Sensor measurement models often consist of averaging the stimulus field over non-overlapping spatial regions surrounding each sensor (e.g. [109]). Using the abstracted sensing system described in Section 2.1.2, we generalize this typical model by representing each sensor with a receptive field $\boldsymbol{\phi}_k(\omega)$ over Ω that performs a weighted average over a spatial region.

Specifically, we assume that the collection of sensors represented by $\{\boldsymbol{\phi}_k\}$ form a frame for a (finite dimensional) vector space of environmental signals $\mathcal{H}_{\boldsymbol{x}} \subseteq \text{span}\,(\{\boldsymbol{\phi}_k\})$, with $\boldsymbol{x} \in \mathcal{H}_{\boldsymbol{x}}$. We suppress the explicit notation of spatial location ω for brevity. This sensor frame has frame bounds (A_s, B_s) and dual functions given by $\{\widetilde{\boldsymbol{\phi}}_k\}$. As in Section 2.1.2, the sensor receptive field would be defined by the physics of the device and could indicate sensors that are directional or have varying sensitivity over a region. As before, ideal sensor measurements of the field are given by the inner product

$$m_k = \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle. \tag{4.1}$$

We will not assume any particular arrangement or shape of the sensor fields; in general we expect sensors to be irregularly spaced and have highly overlapping receptive fields. The measurements described above could also be offset by an additive factor to form relative measurements from a nominal or desired value. These additive terms do not affect our analysis and will not be explicitly notated.

Just as individual sensors have local but overlapping regions of sensitivity, actuator networks are composed of individual actuators that each can affect the environment through (possibly overlapping) local regions of influence. Actuators are indexed by $l \in \mathcal{L}$, and again are located either irregularly or on a regular grid. Whereas each sensor is represented by a receptive field, each actuator is represented by an influence field over Ω, denoted by a function $\boldsymbol{\psi}_l(\omega)$. As with the sensor receptive fields, an actuator's influence field depends on the physics of the specific device and the surrounding medium. In the agricultural irrigation example, the actuator influence field may represent the water delivery pattern of the sprinkler elements. Additionally, the influence field function $\boldsymbol{\psi}_l(\omega)$ may also incorporate the physics of the environment, such as the water absorption or runoff rates of the soil.

Each actuator responds with an intensity that indicates how strongly it acts on the environment. We will model an actuator's intensity d_l as weighting its influence function. The resulting total actuation field $\boldsymbol{y}$ over Ω is

$$\boldsymbol{y} = \sum_{l \in \mathcal{L}} d_l \boldsymbol{\psi}_l,$$

where we again suppress the explicit notation of the spatial domain ω. The collection of actuators can therefore cause any actuation field $\boldsymbol{y}$ in a (finite dimensional) signal space $\mathcal{H}_{\boldsymbol{y}} \subseteq \text{span}\,(\{\boldsymbol{\psi}_l\})$. Just as with the sensors, we assume that the collection of

actuators represented by $\{\boldsymbol{\psi}_l\}$ form a frame for $\mathcal{H}_{\boldsymbol{y}}$ with frame bounds (A_a, B_a) and with dual functions given by $\{\widetilde{\boldsymbol{\psi}}_l\}$.

It is critical to note that the collection of sensors $\{\boldsymbol{\phi}_k\}$ and actuators $\{\boldsymbol{\psi}_l\}$ do not share many characteristics; they can have different numbers of elements at different locations over Ω. Most importantly, individual sensor and actuator functions can have different shapes and even involve different modalities (e.g., temperature sensors and water delivery actuators). Consequently, $\mathcal{H}_{\boldsymbol{x}}$ and $\mathcal{H}_{\boldsymbol{y}}$ can be *very* different spaces of functions, and defining them in terms of general vector spaces allows us to make connections between the sensed inputs and the resulting actuated outputs. Also note that the dual sets $\{\widetilde{\boldsymbol{\phi}}_k\}$ and $\{\widetilde{\boldsymbol{\psi}}_l\}$ are not realized directly in physical systems. For example, the sensor receptive field dual functions $\{\widetilde{\boldsymbol{\phi}}_k\}$ may have spatial characteristics that would be impossible to build into any type of real-world sensor.

In order to design effective communication strategies between sensors and actuators (each represented by a frame), we need to understand the consequences of inducing imperfections in a frame coefficient. Section 2.1.4 analyzed the reconstruction error in a general frame when the coefficients are perturbed. We will draw on these results to understand the impact of reducing communication costs in a wireless sensor and actuator network by approximating the ideal communications required to implement the desired control law.

4.2.2 Other related work

While WSANs are often discussed, there has not been much work quantitatively analyzing their performance. Existing work can be found in areas such as software development models for WSANs [98] and heuristic algorithms for resource competition based on market models [69]. Other recent work [21] uses techniques from causal inference to evaluate specific actuation strategies. None of the current approaches address quantitative methods for decentralized WSAN implementations.

4.3 Connecting Sensors to Actuators

4.3.1 WSAN actuation strategies

A specific WSAN application is defined by its goal. Given an environmental signal, the goal defines the optimal behavior of the system in terms of the actuated response. For example, in the agricultural irrigation scenario the goal would be to keep the measured soil moisture values close to a predetermined set-point [102]. In an insect repellent application, the goal may be to deliver insecticides based on the measurements of temperature and humidity sensors [111]. For any measured stimulus field $\boldsymbol{x}$, we assume that there is a linear behavior mapping $\mathbf{B} : \mathcal{H}_{\boldsymbol{x}} \to \mathcal{H}_{\boldsymbol{y}}$ that defines the ideal actuation field response, $\boldsymbol{y} = \mathbf{B}\boldsymbol{x}$. The mapping $\mathbf{B}$ would be determined as a design specification for the WSAN in advance, and we assume that it remains fixed over a duration between WSAN calibrations. Such a control law based strictly

on the current measurements obviously does not encompass every interesting WSAN application (e.g., target tracking and pursuit). However, such a control strategy does cover a number of significant settings, and our analysis will be limited to applications that fall into this class.

The mapping $\mathbf{B}$ may reflect either an open- or closed-loop control scheme, depending on the nature of the application and whether the actuator activity is reflected directly in the sensors. In a closed-loop scenario, the linear mapping $\mathbf{B}$ is analogous to a proportional controller [66]. As one specific closed-loop example where the input and output spaces are equal, assume that the user has specified a desired environmental field $\boldsymbol{x}_0$ as a set point for the system. For example, a farmer may specify the desired moisture levels of a field containing several different types of crops. In this case, we may want the current environmental field with the addition of the actuated field to be equal to the set point, $\boldsymbol{x}_0 = \boldsymbol{x} + \boldsymbol{y}$. One simple actuation strategy would to apply the identity mapping ($\mathbf{B} = \mathbf{I}$) to the difference of the desired and the actual environmental field, $\boldsymbol{y} = \boldsymbol{x}_0 - \boldsymbol{x}$.

4.3.2 Decentralized strategy for optimal actuation

Following our example of reflex behavior, actuators must calculate their activity levels using weighted combinations of sensor measurements without communicating with the other actuators. The overlapping influence fields of the actuators prevent a purely greedy approach where each actuator generates the locally optimal activity. Nearby actuators could be nearly identical and wildly overcompensate their actions in a greedy approach. Sensors and actuators must together have coordinated behavior that accounts for the components of the action field that other nodes must be covering.

Given a specified actuation function $\mathbf{B}$ and a current environmental field $\boldsymbol{x}$, an ideal actuator network would have each node determine action coefficients $\{d_l\}$ to generate the optimal response $\boldsymbol{y} = \mathbf{B}\boldsymbol{x} = \sum_{l \in \mathcal{L}} d_l \boldsymbol{\psi}_l$. Drawing on the frame theory results from Section 2.1.3, the coefficients weighting the action influence field vectors are given by the inner products between the actuation *dual* vectors and the optimal action signal that we are trying to generate,

$$d_l = \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\boldsymbol{x} \rangle. \tag{4.2}$$

To determine the optimal action coefficients, consider first the reconstruction equation for the stimulus field based on the sensor measurements,

$$\boldsymbol{x} = \sum_{k \in \mathcal{K}} m_k \widetilde{\boldsymbol{\phi}}_k. \tag{4.3}$$

Substituting (4.3) into (4.2), the optimal action coefficients are

$$d_l = \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B} \sum_{k \in \mathcal{K}} m_k \widetilde{\boldsymbol{\phi}}_k \rangle = \sum_{k \in \mathcal{K}} m_k \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle = \sum_{k \in \mathcal{K}} w_{k,l}\, m_k, \tag{4.4}$$

where $w_{k,l} = \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle$ is a weighting factor involving the dual frame vectors. The conversion from sensor measurements $\boldsymbol{m} = \left[m_1, m_2, \ldots, m_{|\mathcal{K}|}\right]^t$ to actuator intensity coefficients $\boldsymbol{d} = \left[d_1, d_2, \ldots, d_{|\mathcal{L}|}\right]^t$ in matrix form is $\boldsymbol{d} = \mathbf{V}\boldsymbol{m}$, where

$$\mathbf{V} = \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_1^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_1 & \widetilde{\boldsymbol{\psi}}_1^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_2 & \cdots & \widetilde{\boldsymbol{\psi}}_1^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_{|\mathcal{K}|} \\ \widetilde{\boldsymbol{\psi}}_2^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_1 & \ddots & & \vdots \\ \vdots & & & \\ \widetilde{\boldsymbol{\psi}}_{|\mathcal{L}|}^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_1 & \cdots & & \widetilde{\boldsymbol{\psi}}_{|\mathcal{L}|}^t \mathbf{B}\widetilde{\boldsymbol{\phi}}_{|\mathcal{K}|} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & \cdots & w_{|\mathcal{K}|,1} \\ w_{1,2} & \ddots & & \vdots \\ \vdots & & & \\ w_{1,|\mathcal{L}|} & \cdots & & w_{|\mathcal{K}|,|\mathcal{L}|} \end{bmatrix}.$$

The expression in (4.4) (or equivalently the entries of $\mathbf{V}$) illuminate the form of the actuator intensity coefficients necessary to generate the optimal total action $\mathbf{B}\boldsymbol{x}$. Unfortunately, each coefficient d_l is a sum including sensor measurements $\boldsymbol{\phi}_k$ over all $k \in \mathcal{K}$; each individual actuator would require knowledge of *every* sensor measurement in order to generate an optimal actuation intensity.

A scenario where every sensor in the network communicates its measurement to every actuator would present an unreasonable communication burden on the network (approximately $|\mathcal{K}| \cdot |\mathcal{L}|$ communication links would be necessary). While a portion of this burden could be reduced through broadcast communication, some sensor-to-actuator links may involve several communications in a multi-hop routing scheme. Any realistic networking scheme will have to eliminate some of these communication links based on their communication cost and their contribution to the total actuation performance.

Intuitively, some sensor measurements will be more important than others in determining an actuator's behavior. For example, a moisture sensor spatially located a long distance away from the influence field of a specific irrigation actuator will likely have very little relevance on that actuator's optimal behavior coefficient. Especially if the sensor measurement was insignificant, this particular communication link could likely be eliminated with no substantial effects to the actuation fidelity [134]. Likewise, a nearby sensor-actuator pair with a poor communications channel between them would not benefit from a high-fidelity transmission. The bitrate on this channel could be dramatically reduced because the quantization errors would still be insignificant compared to the communication errors [135]. These two types of strategies for reducing the communications power will be considered in the following sections.

4.4 Pruning Communication Links

Each entry of the matrix $\mathbf{V}$ indicates a communication link from a sensor to an actuator. Before blindly reducing communications, a networking scheme must know the importance of each possible communication. In a WSN, performance is often judged by assessing the fidelity of the information at the sink, using the distortion of either the original sensor measurements or the underlying stimulus field as the metric.

However, the only performance metric of any consequence in a WSAN is the fidelity of the resulting total action.

4.4.1 Quantifying the cost of pruning

To quantify the importance of individual communications, we must determine how the total actuation performance is affected when a communication is not executed. Consider the case where for actuator l, a subset of sensor nodes $\mathcal{R}_l \subset \mathcal{K}$ do not transmit their measurement coefficient to this actuator (called an "erasure" in the frame literature [73]). Instead of optimal actuator intensity coefficients given in (4.4), actuators form approximate intensity coefficients using the received sensor measurements

$$\widehat{d}_l = \sum_{k \in (\mathcal{K} \setminus \mathcal{R}_l)} m_k \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle. \tag{4.5}$$

The approximate actuator intensities generate a total action field approximating the desired optimal action $\mathbf{B}\boldsymbol{x}$,

$$\widehat{\boldsymbol{y}} = \sum_{l \in \mathcal{L}} \widehat{d}_l \boldsymbol{\psi}_l.$$

Generating a total action field with the approximate coefficients $\{\widehat{d}_l\}$ is equivalent to performing a frame reconstruction with perturbed coefficients, as described in Section 2.1.4. Subtly, the actuator frame vectors are performing synthesis, meaning that dual vectors (with lower frame bound $\frac{1}{B_a}$) are now the analysis set. Therefore, (2.2) relates the fidelity of the approximate actuator intensity coefficients to the fidelity of the resulting total action field,

$$\|\mathbf{B}\boldsymbol{x} - \widehat{\boldsymbol{y}}\|^2 \leq B_a \sum_{l \in \mathcal{L}} |d_l - \widehat{d}_l|^2.$$

Using equations (4.5) and (4.4), we can write the total action field error in terms of individual sensor coefficients *not* communicated to actuator nodes

$$\begin{aligned} \|\mathbf{B}\boldsymbol{x} - \widehat{\boldsymbol{y}}\|^2 &\leq B_a \sum_{l \in \mathcal{L}} \left| \sum_{k \in \mathcal{R}_l} m_k \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle \right|^2 && (4.6) \\ &\leq B_a \sum_{l \in \mathcal{L}} \sum_{k \in \mathcal{R}_l} \left| m_k \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle \right|^2. && (4.7) \end{aligned}$$

As we see in (4.7), the networking strategy for sensor node k can use the value of $\left| m_k \langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k \rangle \right|^2$ to quantify the maximum contribution it would make to the total action error by *not* communicating its measurement to actuator l. The bound in (4.7) can be used to set a threshold λ guaranteeing an absolute upper limit on the actuation error.

Importantly, the form of the error bound in (4.7) isolates each communication link as an independent term so that no communication overhead is required to determine the absolute worst actuation error that can be incurred by eliminating a communication link[1]. In applications where a WSAN must respond quickly to critical but rare events (e.g., a fire suppression system), an absolute bound on the actuation error computed locally is probably appropriate. To ensure that the actuation error is within an absolute tolerance, the active communication links between sensors and actuators will necessarily change depending on the input signal. While this dynamic decision making does not impose a large computational burden on the sensor nodes, the underlying communications network must be able to handle large fluctuations in demand for resources.

The overlap in sensor and actuator fields mean that contributions from two different sensor measurements to an actuator coefficient could, in effect, "cancel" each other. Because the error bound provided in (4.7) is expressly written in terms of local sensor node measurements, this bound favors a conservative interpretation rather than accounting for these interactions. Given a specific communication and networking scenario, it may or may not be advantageous to allow sensors to explicitly communicate to calculate a tighter error estimate (based on the original error expression in (2.1)) and coordinate their communication accordingly. While the frame theoretic analysis paradigm introduced here would allow such an analysis, it would necessarily be specific to the application details (particularly the communication and networking scenario).

4.4.2 Pruning for expected behavior

In many settings, designing around an absolute error constraint results in a system that is too conservative in its average behavior. To analyze the average actuation error one must assume a stochastic model for the measurements, such as assuming that the sensor measurements have zero mean ($\mathcal{E}[\boldsymbol{m}] = 0$) and covariance matrix $\boldsymbol{\Gamma}_{\boldsymbol{m}}$. The covariance matrix $\boldsymbol{\Gamma}_{\boldsymbol{m}}$ will be determined by a combination of the sensor receptive field properties and the distribution assumed on $\boldsymbol{x}$ within the signal space $\mathcal{H}_{\boldsymbol{x}}$. Only the first two moments of the distribution on $\boldsymbol{m}$ are relevant, so we need not assume Gaussian distributions.

Average WSAN performance is much easier to calculate if we recast (4.6) using matrix notation. We first need to write approximate actuator coefficients in (4.5) in terms of a perturbation of $\mathbf{V}$, which captures the ideal transformation from sensor measurements to actuator coefficients. Let the approximate actuator coefficient vector be given by $\widehat{\boldsymbol{d}} = \left(\mathbf{V} + \widetilde{\mathbf{V}}\right)\boldsymbol{m}$, where the matrix $\widetilde{\mathbf{V}}$ is defined to remove inactive

[1]We are assuming that the setup phase of the WSAN has given nodes information about the relative locations of their neighboring nodes that can be used to calculate the necessary inner product. Efficient and accurate node localization is an ongoing branch of WSN research (e.g., [8, 83, 95, 117]).

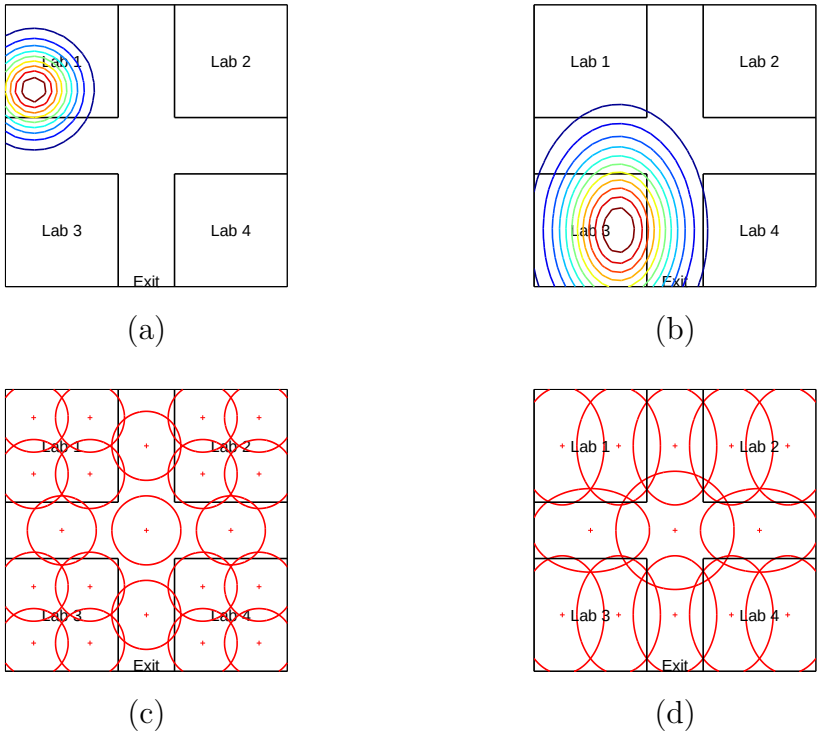


Figure 4.1: Example sensor/actuator locations and shapes for a simple fire suppression example. (a) Contour plot of an example sensor receptive field. (b) Contour plot of an example actuator influence field. (c) Position and single iso-contour of the sensor node receptive fields. (d) Position and single iso-contour of the actuator node influence fields.

communication links:

$$\widetilde{\mathbf{V}}_{k,l} = \begin{cases} -\left(\widetilde{\boldsymbol{\psi}}_l^t \mathbf{B} \widetilde{\boldsymbol{\phi}}_k\right) & \text{if } k \in \mathcal{R}_l \\ 0 & \text{if } k \in (\mathcal{K} \setminus \mathcal{R}_l). \end{cases}$$

Incorporating this definition into (4.6) and taking the expectation of both sides lets us bound the average error

$$\mathcal{E}\left[\|\mathbf{B}\boldsymbol{x} - \widehat{\boldsymbol{y}}\|^2\right] \leq B_a \mathrm{Tr}\left[\widetilde{\mathbf{V}} \boldsymbol{\Gamma}_{\boldsymbol{m}} \widetilde{\mathbf{V}}^t\right], \tag{4.8}$$

where $\mathrm{Tr}\left[\cdot\right]$ is the trace operator.

A system designer could use (4.8) to characterize (on average) how important a communication link between a specific sensor and actuator pair is to generating the total actuation field. Using this information, a WSAN design could specify which communication links between sensors and actuators are active. Such a scheme has the disadvantage that it may not react well to events that are large deviations from the usual behavior. However, non-adaptive WSAN communication schemes can use communication resources more efficiently most of the time, limit the network com-

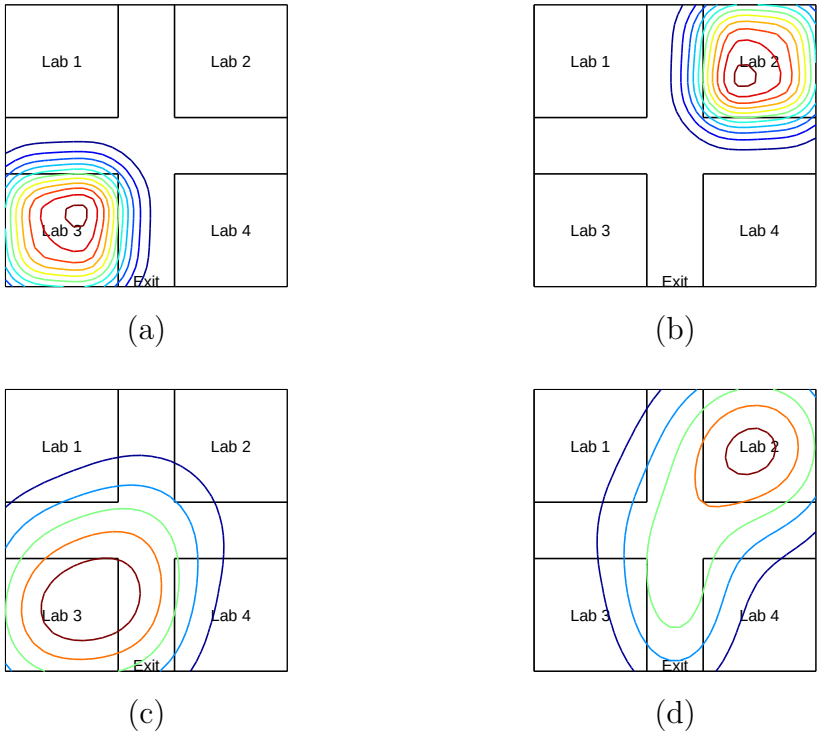


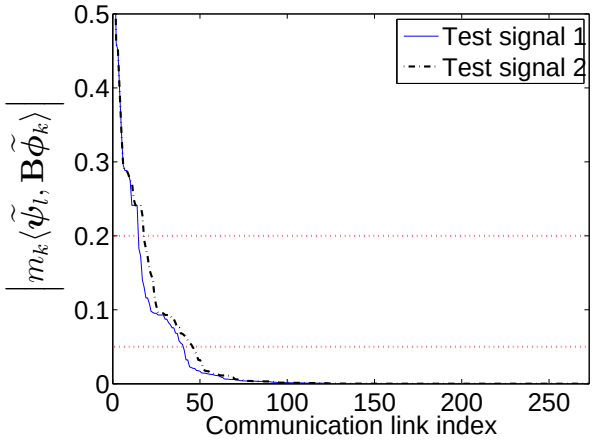
Figure 4.2: Contour plots of sample temperature fields and optimal actuation fields. (a) A sample temperature field for test signal 1 indicating a fire in lab 3. (b) A sample temperature field for test signal 2 indicating a fire in lab 2. (c) Optimal actuation response for test signal 1. (d) Optimal actuation response for test signal 2. Different spatial response characteristics in the actuation response keep the main exits clear.

munication burden for any stimulus field, and more easily integrate the real cost of executing individual communication links (through a possibly multi-hop network) into generating an optimal strategy. Also, it is worth noting that the bound in (4.8) is tighter than the bound in (4.7) (because it is based directly on (4.6)), reflecting the fact that all of the communication links can be considered jointly when designing the system for average error performance.

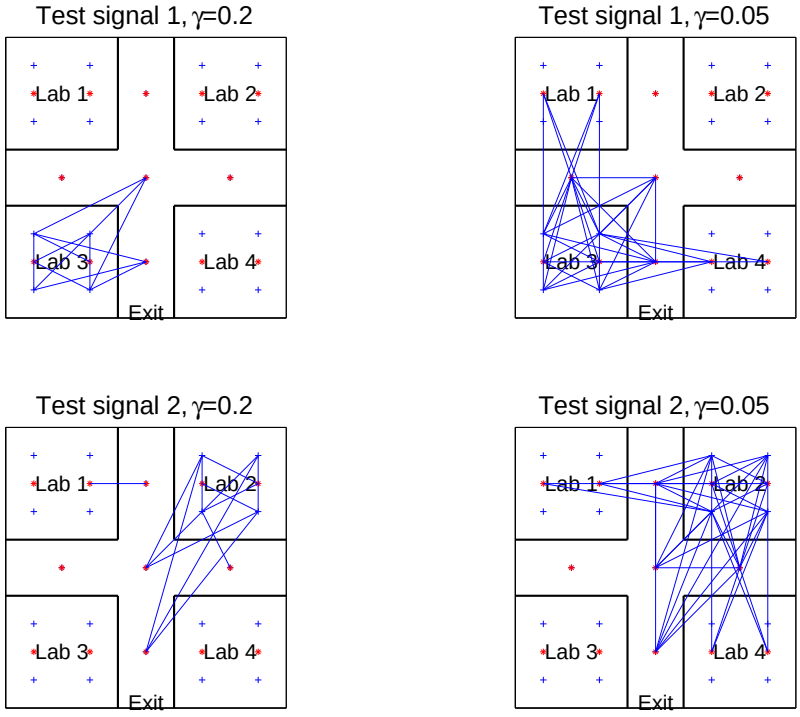
4.4.3 An example WSAN system

As an illustrative example, consider a WSAN operating a stylized fire suppression system in an office building with four research labs. The building space is covered with a network of 21 temperature sensors (modeled with radially symmetric, exponentially-decaying receptive fields) and 13 actuators (modeled with an oriented and exponentially decaying influence field), all illustrated in Figure 4.1. This WSAN has 273 possible communication links from the sensor nodes to actuator nodes.

The actuation response is generated for both test signals using threshold values of $\lambda = 0.2$ and $\lambda = 0.05$, and the resulting total actuation fields are plotted in



(a)



(b)

Figure 4.3: Importance measurements and connection diagrams for each communication link. (a) The importance measurements of each communication link $\left(\left|m_k\langle\widetilde{\psi}_l, \mathbf{B}\widetilde{\phi}_k\rangle\right|\right)$ are sorted and plotted for the two test signals. (b) Connection diagrams for the two test signals under the two thresholds in the example system. Sensor nodes are marked with a blue (+) and actuator nodes are marked with a red (*). Active connections from a sensor to an actuator are denoted by a blue line.

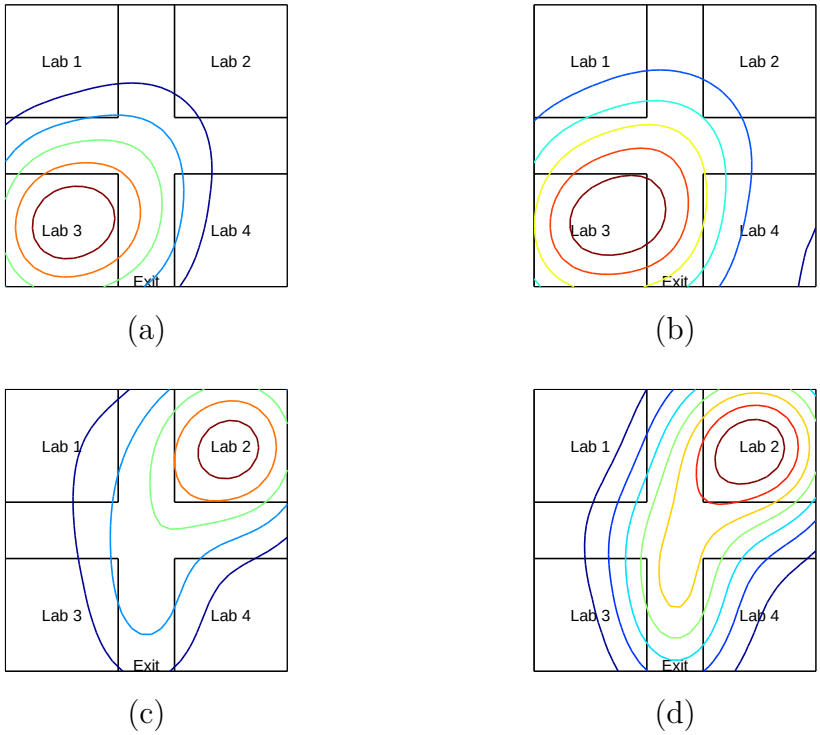


Figure 4.4: Contour plots of actuation responses when using only a subset of possible communication links. Links are pruned by thresholding each link's importance to the total actuation. (a) Approximate responses to test signal 1 when using 14 communication links. (b) Approximate responses to test signal 1 when using 40 communication links. (c) Approximate responses to test signal 2 when using 17 communication links. (d) Approximate responses to test signal 2 when using 45 communication links.

Figure 4.4. The reduced communication scheme based on the thresholds resulted in the number of active communication channels and associated percentage errors given in Table 4.1. The principles discussed in Section 4.4.1 allow the WSAN to generate excellent approximations to the optimal actuation field by using local rules to activate only a fraction of the communication links. Interestingly, if we activate the same number of links using the more intuitive measure $|m_k\langle \boldsymbol{\psi}_l, \mathbf{B}\boldsymbol{\phi}_k\rangle|$, the resulting actuation error increases by roughly an order of magnitude.

We specified a function $\mathbf{B}$ mapping the temperature inputs to an imaginary desired fire suppression output. To illustrate that this mapping may be spatially varying, we note that fire activity in all labs will induce fire suppression activity along a path to the main exit. We used two sample temperature fields indicating a fire in different labs areas (shown in Figure 4.2, along with optimal responses). As discussed in Section 4.4.1, the quantity $\left|m_k\langle \widetilde{\boldsymbol{\psi}}_l, \mathbf{B}\widetilde{\boldsymbol{\phi}}_k\rangle\right|$ determines the importance of each communication link (sorted and plotted in Figure 4.3 for these test signals). When a measurement is taken, we allow communication on a link only when the importance value is greater than the threshold λ. In these signals, a threshold of $\lambda = 0.2$ allows

Table 4.1: Results from the example WSAN fire suppression system.

	$\lambda = 0.2$	$\lambda = 0.05$
Active links	14	40
Relative error	2.22%	0.04%

Test signal 1

	$\lambda = 0.2$	$\lambda = 0.05$
Active links	17	45
Relative error	2.46%	0.15%

Test signal 2

approximately 15 of the 273 possible communication links to be active, and $\lambda = 0.05$ allows approximately 40 active communication links. The resulting active communication links are shown in Figure 4.3. Close examination of the connection diagrams shows that some communication choices are non-obvious; the most important sensor to a particular actuator is not always the one with heavily overlapping influence functions.

4.5 Optimal Power Scheduling

Each communication in the WSAN setting contributes to the actuation fidelity based on the interplay of several factors, including the measurement reliability, the transmission fidelity, the communication channel quality, the measurement value, and the specific sensor-actuator pair in the WSAN topology. Section 4.4 details an analysis of an adaptive situation where sensors prune communication links based on a combination of the measurement value and the overall importance of that sensor-actuator pair in the actuation fidelity. This pruning strategy has the advantage of being adaptive (i.e., depending on the sensor's measurement), but does not consider the fidelity of a link or the cost of performing that communication. In essence, all communications are considered equally costly and equally precise. This section will consider the WSAN design problem of optimally allocating communication resources to each link depending on the reliability of that measurement and the energy required to communicate it to its destination. We propose and solve the optimal power scheduling problem for this WSAN model by determining the optimal bitrates that each communication link should use when jointly considering the sensor noise, quantization noise and channel errors.

Specifically, we consider a set of k sensors, each recording the ideal measurement corrupted by additive noise due to sensor imperfections:

$$m_k^s = m_k + n_k^s = \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle + n_k^s.$$

We assume that the sensor noise is zero mean with variance $\mathcal{E}\left[(n_k^s)^2\right] = \sigma_k^2$, but the distribution is otherwise unknown. Note that in general, the sensors can be inhomogeneous with different local signal-to-noise ratios. We further assume that the measurements lie in the range $m_k^s \in [-\Delta, \Delta]$. Note also that depending on the form of the actuation law, the sensor may record the difference from a nominal value (or

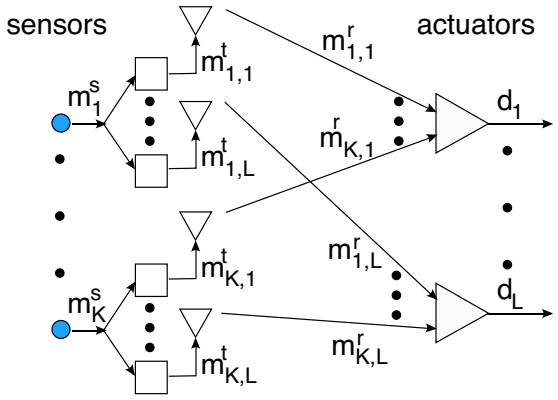


Figure 4.5: A schematic of the communications pathways in the WSAN model. The k^{th} sensor makes a noisy reading m^s_k. The quantized measurement $m^t_{k,l}$ with $\beta_{k,l}$ bits of precision is sent to the l^{th} actuator. After incurring bit errors in the wireless channel, the measurements $m^r_{k,l}$ are received by the actuator and used to form the approximate actuation coefficient $\widehat{d}_l$.

a set point) instead of an absolute measurement. We will not notate this explicitly because a mean shift is irrelevant to the subsequent noise analysis.

The ideal actuation coefficient for the l^{th} actuator is a weighted combination of the ideal sensor measurements (shown in (4.4)), where the weights $\{w_{k,l}\}$ are determined by the actuation strategy as described in Section 4.3.2. However, each actuator must quantize its measurement and send it with some finite precision along a noisy communications channel to each actuator. An inherent rate-distortion tradeoff exists for a single actuator coefficient: higher fidelity in the estimated actuator coefficient requires higher precision in the transmitted measurements, but sending more bits expends more energy. The optimal power scheduling problem results from realizing that not all sensor coefficients are equally reliable or important to a given actuator, and the variable distances among the nodes creates an inhomogeneity in the cost of each communication link. The fidelity of each communicated message can be chosen through using a variable number of bits (including zero) on each communication path to optimally use the system resources. To set up the optimal power scheduling problem, we must have both a model for the communications scheme that calculates the relative transmission power needed to communicate each bit and a model of the distortion that calculates the relative error reduction for each bit of precision.

4.5.1 Communication energy model

We first consider the transmission energy required to communicate the measurement m^s_k from sensor k to actuator l. Following previous work in decentralized estimation in the WSN context [30, 31, 157], we assume that sensors communicate to

an actuator over a wireless link using a M-order quadrature amplitude modulation (MQAM) strategy and following a common multiple access scheme (e.g., time-division multiple access) [122]. If sensor k sends $\beta_{k,l}$ bits of precision about its measurement to actuator l, the MQAM scheme will have a constellation with $2^{\beta_{k,l}}$ symbols. We also assume that the transmission scheme is designed with a fixed bit error probability of p over an additive white Gaussian noise (AWGN) channel.

We expect that to achieve a fixed bit error rate, nodes will require higher transmission energy when we increase the number of bits being communicated. In previous work [30, 31, 157], an upper bound on the transmission energy under this MQAM transmission scheme is estimated to be

$$P_{k,l} \leq \kappa d_{k,l}^{\alpha} \left(2^{\beta_{k,l}} - 1\right) \ln\left(\frac{2}{p}\right),$$

where $d_{k,l}$ is the distance between the sensor and actuator, α is the fading coefficient of the transmission medium, and κ is a constant depending on several aspects of the communication environment (e.g., channel signal-to-noise ratio, antenna gain, coding gain, etc.). This expression applies for coded or uncoded MQAM, though κ will change when error correcting codes are used. We take κ to be a constant for all communication links in the system and we estimate the relevant portion of the transmission power for a measurement by the upper bound

$$P_{k,l} \propto d_{k,l}^{\alpha} \left(2^{\beta_{k,l}} - 1\right). \tag{4.9}$$

If the communications scenario is inhomogeneous, additional link-dependent constants can be introduced without significant difficulty. While the constants being left out of this expression are critical for determining the exact amount of transmission energy required, they are not necessary for comparing the relative effectiveness of two competing resource allocation strategies. For our purposes, we will therefore interpret the transmission power for a node as being equality in (4.9).

4.5.2 Communication distortion model

Quantization noise

Computation and communication systems must represent measurements in terms of a finite precision bitstream. Ideally, a real-valued measurement in the range $m_k^s \in [-\Delta, \Delta]$ is represented by an infinitely long bitstream $\{b_j\}, b_j \in \{0, 1\}$,

$$m_k^s = 2\Delta \left(\sum_{j=1}^{\infty} b_j 2^{-j}\right) - \Delta.$$

The k^{th} sensor quantizes its transmitted measurement to the l^{th} actuator ($m^t_{k,l}$) by truncating this series at $\beta_{k,l}$ bits

$$m^t_{k,l} = 2\Delta \left(\sum_{j=1}^{\beta_{k,l}} b_j 2^{-j} \right) - \Delta.$$

The resulting quantization error thus creates an additive noise term

$$m^t_{k,l} = m^s_k + n^q_{k,l}.$$

Assuming m^s_k is uniformly distributed over $[-\Delta, \Delta]$ and uniform quantization is employed, it is straightforward to calculate the first two moments of the quantization noise

$$\mathcal{E}\left[n^q_{k,l}\right] = 0, \qquad \mathcal{E}\left[\left(n^q_{k,l}\right)^2\right] = \frac{\Delta^2}{3} 2^{-2\beta_{k,l}}.$$

The uniform distribution of the measurements and the specific form of the quantizer are not critical for the setup of the optimization problem established in Section 4.5.3. If a given application has a more appropriate model, the variance of that quantization noise can be used instead. Note also that although the same measurement is being communicated from sensor k to every actuator, the quantization noise term is different for each of those transmissions. In fact, some of those transmission links may use zero bits (i.e., no transmission actually occurs) when the transmission would be particularly costly relative to the benefit of transmitting the measurement.

Transmission noise

Ideally the quantized measurement would reach the destination exactly as it was transmitted. However, communication with finite power on a wireless channel will inevitably suffer errors. When sensor k transmits its quantized measurement $m^t_{k,l}$ to actuator l, the received measurement is denoted $m^r_{k,l}$. We can similarly represent the inaccuracy caused by transmission errors as an additive noise term

$$m^r_{k,l} = m^t_{k,l} + n^t_{k,l}.$$

To send its quantized measurement to actuator l, sensor k transmits the bit sequence $\{b_j\}_{j=1}^{\beta_{k,l}}$ along a noisy channel. The received bit sequence $\{b^r_j\}_{j=1}^{\beta_{k,l}}, b^r_j \in \{0,1\}$ is a Bernoulli random variable with the conditional distribution based on the bit error rate p

$$\begin{aligned} P\left(b^r_j = b_j | b_j\right) &= 1 - p \\ P\left(b^r_j = (b_j \oplus 1) | b_j\right) &= p, \end{aligned}$$

where $\oplus$ denotes modulo-2 addition. The resulting received measurement is

$$m_{k,l}^r = 2\Delta \left(\sum_{j=1}^{\beta_{k,l}} b_j^r 2^{-j} + 2^{-(\beta_{k,l}+1)} \right) + \Delta.$$

We can then write the transmission noise term

$$n_{k,l}^t = 2\Delta \sum_{j=1}^{\beta_{k,l}} 2^{-j} \left(b_j^r - b_j \right).$$

A uniform distribution of m_k^s over the dynamic range translates to an assumption that b_j is a Bernoulli random variable with probability 1/2. Consequently, the expectation of the transmission noise is also zero, $\mathcal{E}\left[n_{k,l}^t\right] = 0$. The variance of the transmission noise can be calculated as:

$$\mathcal{E}\left[\left(n_{k,l}^t\right)^2\right] = 4\Delta^2 \sum_{j=1}^{\beta_{k,l}} 2^{-2j} \mathcal{E}\left[\left(b_j^r - b_j\right)^2\right] = \frac{4p\Delta^2}{3}\left(1 - 2^{-2\beta_{k,l}}\right).$$

4.5.3 Optimal bit allocation

As mentioned earlier, the power scheduling problem is fundamentally a rate-distortion tradeoff. In one form of the resource allocation problem, we desire to meet a specified distortion criteria while expending the minimal amount of transmission energy across the WSAN. The vehicle for making this tradeoff is to assign the bit allocations on each sensor-actuator link $\{\beta_{k,l}\}$ to make optimal use of the power resources. As in [157], we will also minimize the ℓ^2 norm of the total power as a compromise measure between minimizing the total power (the ℓ^1 norm) and the maximum power (the ℓ^∞ norm).

We consider for now a single actuator coefficient d_l, which is approximated at the actuator using the received measurements

$$\widehat{d_l} = \sum_{k \in \mathcal{K}} w_{k,l}\, m_{k,l}^r = \sum_{k \in \mathcal{K}} w_{k,l} \left(m_k + n_{k,l}\right),$$

where $n_{k,l} = n_k^s + n_{k,l}^q + n_{k,l}^t$ represents the total distortion from the ideal sensor measurement. As stated earlier, the results from Section 4.4.1 indicate that reducing the error in the actuator coefficients causes a proportional decrease in the upper bound of the error on the actuation field. Therefore, we will consider only the fidelity of a single actuation coefficient here, which can be used to bound the fidelity of the total actuation field. If desired, the resource allocation problem outlined here can be applied jointly to all of the actuator coefficients.

Making the typical assumption that the noise sources due to the sensor, quan-

tization and bit errors are independent, we calculate the variance of the total noise source as the sum of the component variances

$$\mathcal{E}\left[(n_{k,l})^2\right] = \sigma_{k,l}^2 = \sigma_k^2 + \frac{\Delta^2}{3}2^{-2\beta_{k,l}} + \frac{4p\Delta^2}{3}\left(1 - 2^{-2\beta_{k,l}}\right).$$

Accounting for the multiplication of the weights necessary to calculate the actuation coefficient, the MSE of $\widehat{d}_l$ is given by

$$\mathcal{E}\left[\left(d_l - \widehat{d}_l\right)^2\right] = \sum_{k\in\mathcal{K}} w_{k,l}^2 \sigma_{k,l}^2.$$

This distortion model clearly has some limitations. It is likely that the noise terms will not be completely independent, especially at very low bit rates. We will see in a simulation described later that the independence assumption does not have a large effect on the results for distortion ranges of interest. Also, the sensor noise is included in the distortion calculation even when no bits are used on the communication link and the measurement is not communicated at all. We will also see that this inaccuracy in the model does not have a large effect as long as the sensor noise is small compared to the dynamic range of the sensor so it is overwhelmed by the quantization error in the low bit-rate regime.

For individual actuator coefficients, the optimal resource allocation problem is expressed as an optimization of the bitrates $\{\beta_{k,l}\}$ over the field $\mathcal{X} = \mathbb{Z}_+$ subject to a distortion constraint of D_0,

$$\min_{\beta_{k,l}\in\mathcal{X}} \quad \sum_{k\in\mathcal{K}} P_{k,l}^2$$

$$\text{s.t.} \quad \mathcal{E}\left[\left(d_l - \widehat{d}_l\right)^2\right] \leq D_0.$$

Writing this optimization problem in terms of the energy and variance terms calculated from the communication and distortion models described above, the optimal resources allocation problem is fully stated as

$$\min_{\beta_{k,l}\in\mathcal{X}} \sum_{k\in\mathcal{K}} d_{k,l}^{2\alpha}\left(2^{\beta_{k,l}} - 1\right)^2$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} w_{k,l}^2\left(\sigma_k^2 + \frac{\Delta^2}{3}\left(4p + (1-4p)\,2^{-2\beta_{k,l}}\right)\right) \leq D_0.$$

This optimization is an integer program, whose solutions are generally NP-hard. As in [157], we relax the space of the optimization problem to be $\mathcal{X} = \mathbb{R}_+$, where we obtain a convex program with an explicit solution.

Relaxing the set of admissible solutions to the real numbers means that the solution will no longer fit our original constraint set, but the real valued solutions are

likely to be very close to the optimal integer solutions we desire [58,70]. By using the typical procedure of converting the optimal real-valued answers into integer values by rounding, we will see that even introducing this suboptimality still allows for significant relative gains over uniform bit allocation methods. Using a ceiling function to convert the resulting values to integers ensures that the targeted distortion criteria D_0 is still met. However, using a less stringent rounding function may produce a better rate-distortion tradeoff curve even though the individual distortion criteria are not guaranteed to be met. In particular, using a rounding instead of a ceiling function generates many more links using zero bits, thereby possibly saving the overhead energy needed to power up the communications circuits on the sensor.

We use Lagrangian methods to solve this optimization for $\beta_{k,l} \in \mathbb{R}_+$. Making the substitution $z_{k,l} = 2^{\beta_{k,l}}$ highlights the convex nature of the underlying optimization problem.[2] The resulting Lagrangian function is

$$\Lambda = \sum_{k \in \mathcal{K}} d_{k,l}^{2\alpha} \left(z_{k,l} - 1\right)^2 + \left(\theta \sum_{k \in \mathcal{K}} w_{k,l}^2 \left(\sigma_k^2 + \frac{\Delta^2}{3} \left(4p + (1-4p)\, z_{k,l}^{-2}\right) \right) - D_0 \right),$$

where θ is the Lagrange multiplier constant. Differentiating Λ with respect to $z_{k,l}$ and setting it equal to zero we see that the optimal values of $z_{k,l}$ are those that meet the constraints and are a solution of the quartic equation

$$z_{k,l}^3 \left(z_{k,l} - 1\right) - \theta \left(\frac{w_{k,l}^2 \Delta^2 \left(1 - 4p\right)}{3 d_{k,l}^{2\alpha}} \right) = 0.$$

While analytic solutions to the quartic equation are cumbersome, numerical methods can easily solve this equation for specific numeric values. There are four roots to this quartic equation, only one of which is real and positive so as to be a viable solution to our optimization problem. Moreover, we find that the real and positive solutions for $z_{k,l}$ are strictly increasing in θ, which suggests an optimization strategy similar to the well-known water-filling solution problem for multi-channel communications [27]. We can increase θ to increase the number of bits allocated to each channel until the distortion criteria D_0 is met with equality. We see that communication links with measurements more critical to the actuation coefficient (measured by $w_{k,l}^2$) will be assigned bits more quickly, and communication links expending more transmission energy (measured by $d_{k,l}^{2\alpha}$) will be penalized and assigned bits more slowly.

4.5.4 Numerical results

To demonstrate the utility of the power scheduling problem described in Section 4.5.3, we examine a stylized WSAN setup for agricultural irrigation. Figure 4.6

[2]This type of substitution is standard in the field of geometric programming [54] to transform an objective function containing products of exponentials into a convex program.

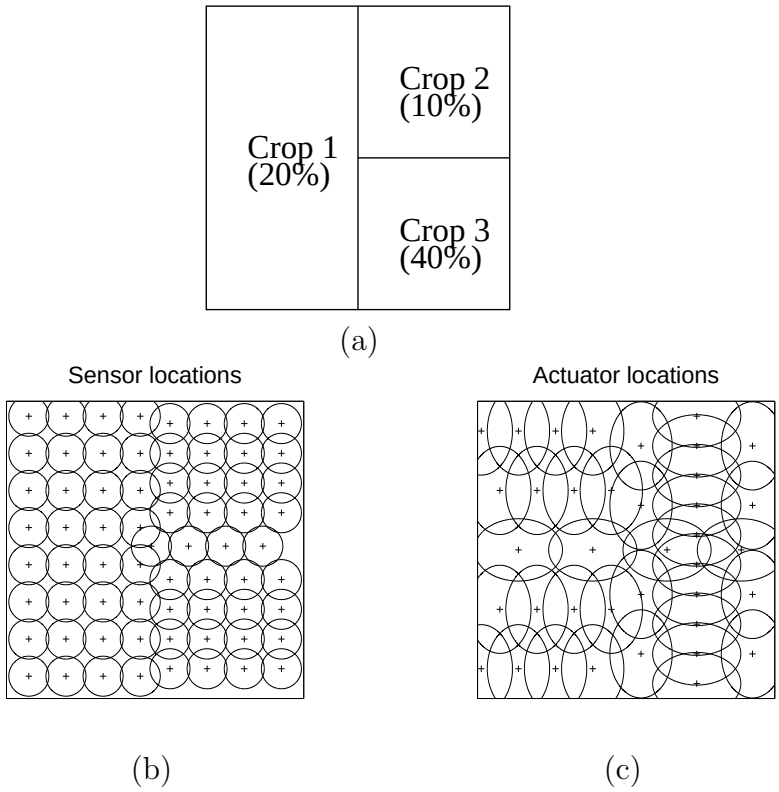
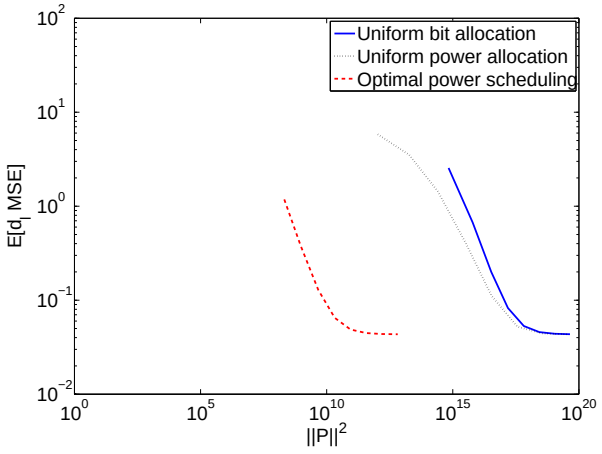


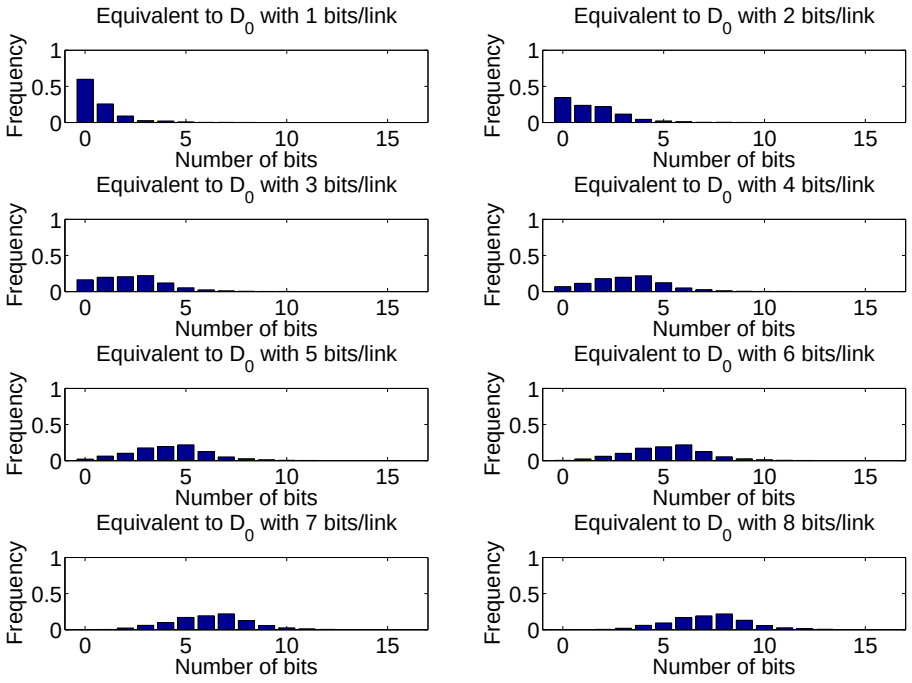
Figure 4.6: An example WSAN for a stylized agricultural irrigation application. (a) A field 100m on each edge has three crops with varying desired soil moisture percentages. A collection of moisture sensors (b) and irrigation actuators (c) are deployed, with a single contour denoting where the receptive field or influence field has mostly vanished.

denotes the location of different crops with varying optimal soil moisture needs, as well as the location of 68 moisture sensors and 38 irrigation actuators within a 100m square field. The sensor receptive fields are modeled by a circular Gaussian function, corresponding to the Green's function for a homogeneous diffusion medium with infinite boundaries [26]. The actuator influence fields are modeled by an elliptic Gaussian function, representing a shaped water delivery pattern. Denoting the ideal moisture levels depicted in Figure 4.5 as $\boldsymbol{x}_0$, the ideal sensor measurements are relative to the nominal measurement, $m_k = \langle \boldsymbol{x}_0, \boldsymbol{\phi}_k \rangle - \langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle$. One choice of actuation function to achieve the set point is $\mathbf{B} = \mathbf{I}$, making the weights $w_{k,l} = \langle \widetilde{\boldsymbol{\psi}}_l, \widetilde{\boldsymbol{\phi}}_k \rangle$. We use simulation parameters $\alpha = 3.5$ and $p = 10^{-3}$. To focus on the noise aspect related to bit allocation, we also assume a common sensor noise variance of $\sigma_k^2 = 10^{-4}$ for all k.

Focusing on a specific actuator coefficient d_l, we first examine the theoretical improvement of the power scheduling algorithm described in Section 4.5.3 over a uniform bit allocation scheme. Figure 4.7 shows the rate distortion curve produced for the expected MSE $\mathcal{E}\left[\left(d_l - \widehat{d}_l\right)^2\right]$ when each link from a sensor to the l^{th} actuator used



(a)



(b)

Figure 4.7: The rate-distortion curve and bit allocation rates after optimal power scheduling. (a) The average rate-distortion curve comparing MSE estimation error for a single actuation coefficient d_l and the total network power needed to achieve that distortion. (b) The distribution of bits on the sensor-actuator communication links after optimal power allocation. Each pane corresponds to a distortion equivalent to a uniform bit allocation scheme. The bars indicate the fraction of links that communicate with a specific number of bits.

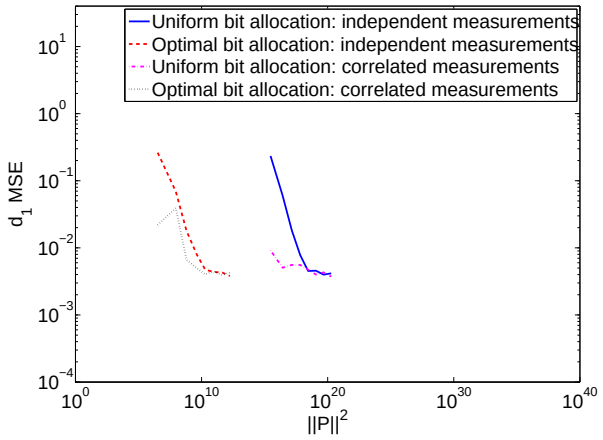


Figure 4.8: The rate-distortion curves calculated through simulated measurements. Measurements were simulated directly through independent random generation. Correlated measurements were also simulated by generating random environmental fields from a piecewise linear class and taking the resulting sensor measurements.

either the same number of bits or the same amount of power. For comparison, the optimal power scheduling algorithm of Section 4.5.3 was run to find optimal bit allocations and the resulting calculated rate-distortion curve is also shown in Figure 4.7. We note that the bit allocation algorithm is able to reduce the network power consumption by several orders of magnitude for the same distortion over both allocation strategies allocating a uniform number of bits or a uniform power level to each link. The rate-distortion curves shown are averages over all actuator coefficients in the example setup. Figure 4.7 also indicates the distribution of bit assignments in the optimal power allocation scheme. Each pane corresponds to a distortion equivalent to a uniform bit allocation scheme, with the bars denoting the fraction of sensor-actuator communication links using that number of bits. The distribution of bit allocation levels across a range of values both above and below the corresponding uniform allocation indicates that power savings are being achieved not only through a reduction of bits but also through allocating those bits to the communication links that are the best combination of important and reliable.

To verify our independent additive noise distortion model and the calculated rate-distortion curves shown above, we simulated the calculation of actuation coefficients from correlated measurements in the WSAN example application. Figure 4.8 shows the rate distortion curves calculated when independent uniform measurements are used to calculate actuation coefficients. The sensor noise, quantization and transmission error simulation is the same as the previous simulation. We see that the rate-distortion curves are essentially the same as those theoretically calculated and shown in Figure 4.7. To explore the effect of sensor correlation on our distortion

estimate, we performed the same experiment when the measurements are correlated. In this experiment we generated random environmental fields from a class of piecewise linear functions. The resulting correlated measurements were used to calculate the actuator coefficients in the same manner and the rate-distortion results are also shown. We see that these correlations do not play a significant role in the distortions calculated from the theoretical model.

4.6 Summary and Future Work

4.6.1 Summary of contributions

WSANs are often discussed as a logical extension to sensor networks, but there is little research investigating sensor and actuator systems working in concert together. While algorithms that reduce communications and ensure data fidelity for sensor measurements are important for many applications, they are not the ultimate arbiter for obtaining good actuation performance. The total system must be designed and managed with the final actuation goal in mind. However, the coordination among overlapping sensor and actuator nodes without a centralized controller presents, however, a daunting challenge to this goal.

Our frame-theoretic WSAN model illustrates one strategy for taking such a holistic information management view with actuation fidelity as the relevant metric. We have demonstrated that using frames as an abstract model to include information about the ways the sensors and actuators interact with the environment allows us to implement linear control laws with no explicit centralized coordination. The necessary coordination is achieved through the use of dual frame vectors, which allow each sensor-actuator pair to account for the behavior of their neighboring nodes without direct communication.

The analytic tools we present characterize the effect of eliminating an individual communication link between a sensor and an actuator, both in terms of absolute (for specific sensor measurements) and average actuation error. Choosing a networking strategy for eliminating communication links is both difficult and non-intuitive. While intuition would indicate that the relationship between the activation fields of a sensor and an actuator are the relevant quantity characterizing the importance of the communication between those two nodes, our work shows that it is the relationship between the mathematical *duals* of the activation fields that captures this inherent importance. It is through these dual functions that the relationship of the whole sensor network to the whole actuator network can be accounted for in local communications between pairs of nodes. Characterizing the importance of individual communication links to the overall goal points directly to how a networking strategy could weigh the costs and benefits of each communication link to achieve the desired balance between performance and energy efficiency. The value of our analysis is highlighted in an example WSAN system where link activations based on the sensor and actuator duals performed an order of magnitude better than activations based on the simple overlap

of the sensor and actuator receptive field functions.

We have also addressed the variable communication costs and fidelities of each communicated measurement by solving the optimal power scheduling algorithm problem for our WSAN system. The solution of this convex optimization problem yields a resource allocation strategy that greatly reduces the network's power demands compared to a uniform resource allocation. While this optimization involves centralized information and processing, the nodes operate independently once they are given their optimal bit allocations for each communications link. The optimization strategy can therefore be done off-line and updated with a simple transmission to each sensor whenever the nodes are recalibrated. In this way, the necessary coordination is built into the pre-calculation and no centralized controller is needed to implement the control strategy.

4.6.2 Future directions in distributed WSAN strategies

There are many open areas for future work in WSAN information management. Our WSAN model assumes a simple network topology where every node can communicate to every actuator. This is clearly an unrealistic assumption when the spatial extent of the WSAN is large and multiple hops would be needed to connect two nodes. Future work will be needed to determine when a WSAN environment requires nodes connected over an area wider than their transmission range, and to extend our communication reduction strategies to these more complicated network topologies.

We envision several direct extensions to the WSAN strategies addressed in this chapter. Recent research shows that altering the bit-interval durations to create lower bit error rates for the most significant bits can result in reduced error of the received measurement [85]. It is possible that even more significant power savings gains could be produced by incorporating the bit-interval duration as a design parameter in the optimal power scheduling problem. Similarly, WSAN information processing strategies may also benefit from recent work and analysis in cooperative MIMO communications [29, 110] and distributed estimation using analog communication [32].

The two different communication reduction strategies described in this chapter (adaptive link pruning and optimal power scheduling) each have their own advantages. One can imagine a hybrid strategy between these two approaches where the resource allocation is a combination of off-line computations and adaptive decision making. It is likely that such a system, though possibly difficult to analyze, could produce significant gains over each strategy in isolation.

Finally, our WSAN analysis has assumed that a control law specifying the system task is specified in the design stage and our goal is to simply implement this desired behavior using distributed strategies. However, the design of good actuation laws is non-trivial, especially in the presence of uncertainty in the physical models of the environment. It is also possible that significant gains can be produced if the optimal control laws are found that account for the communication costs needed to implement them in WSANs.

Chapter 5
Exploiting Redundancy for Sparse Approximation: Sparse Coding via Locally Competitive Algorithms

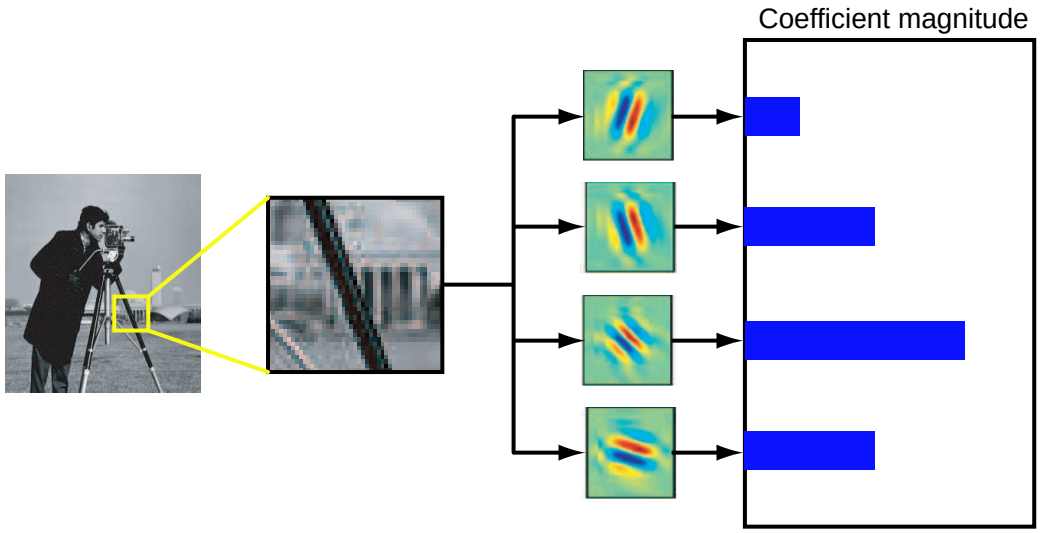
Perhaps the most powerful benefit of redundant representations is their flexibility. Section 2.1.3 mentions that a signal can be represented by an infinite number of coefficient choices when the dictionary is overcomplete. This flexibility allows the system to go beyond a simple linear transform of the data (e.g., a Fourier transform) to a data encoding that may help the system understand the content of the signal. When the task is scene *understanding* rather than just scene capture, this flexibility allows the systems to represent data in a format that makes that task easier.

For example, the flexibility of an overcomplete dictionary often allows an image to be well-approximated by a small subset of dictionary elements [12] even though natural images are composed of several contrasting components (e.g., edges, textures, and smooth regions). Consider an image patch containing an edge, illustrated heuristically in Figure 5.1. Calculating a set of coefficients by simply projecting the image onto the dictionary elements makes it difficult to determine the presence of the edge. In contrast, an encoding that uses only a few non-zero coefficients (thereby choosing only the best-matching dictionary elements) would make the signal structure more explicit and easier to determine. This process of using the flexibility in an overcomplete dictionary to choose a small subset of coefficients to represent a signal is known as *sparse approximation*. Though the flexibility of an overcomplete dictionary could be used to find coefficients with many various properties, we focus on sparsity as a compelling principle.

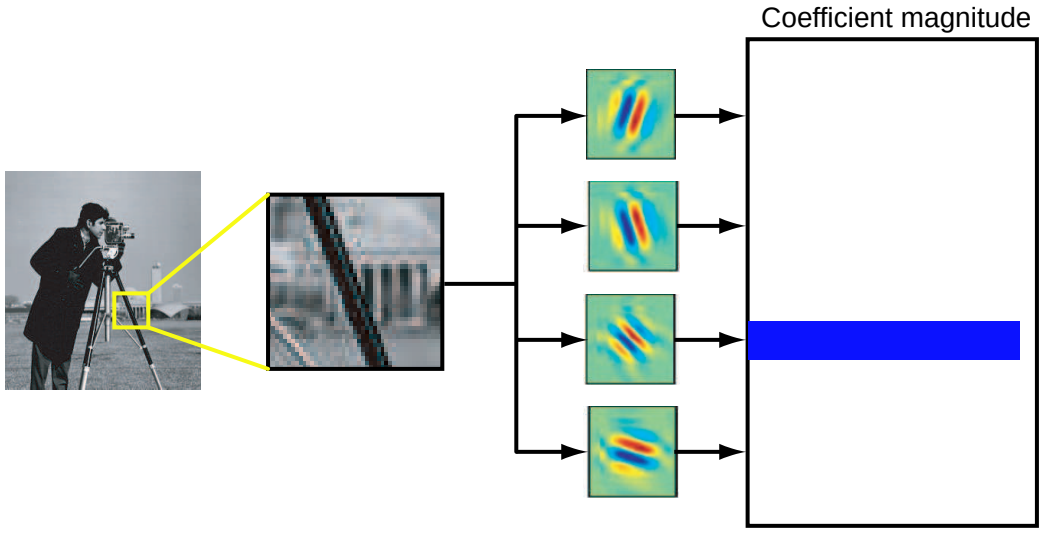
5.1 Sparse Signal Representations

Recent theoretical and experimental evidence indicates that many sensory neural systems appear to employ sparse representations with their population codes [41, 96, 113, 114, 150], encoding a stimulus in the activity of just a few neurons. While sparse coding in neural populations is an intriguing hypothesis, neurally plausible mechanisms capable of efficiently finding sparse approximations are currently unknown. The challenge of collecting simultaneous data from large neural populations makes it difficult to evaluate this hypothesis without testing predictions from a specific proposed mechanism.

Sparse approximation is a challenging problem that is the center of much current research in mathematics and signal processing. Inferring sparse representations of a



(a)



(b)

Figure 5.1: Projection and sparse coefficient characteristics for an image patch containing an edge. (a) Coefficients calculated through projecting the edge in the image patch onto the dictionary elements. Projection coefficients represent the image exactly, but make it difficult to determine the presence of the edge. (b) A sparse set of coefficients representing the edge in the image patch. In addition to representing the image exactly, sparse coefficients also make the content more explicit. In this case, the edge in the image patch is much more evident from the coefficients.

signal is a key ingredient for solutions to many signal and image processing tasks, including compression [44, 67, 104], denoising [19, 47], super-resolution (deconvolution) [61, 62, 145], in-painting [129], machine learning [71], and model-selection [128]. The main idea behind these methods is that when using the correct dictionary, signals of interest can often be represented sparsely whereas the confounding signal elements (i.e., the interference) is not sparsely representable. Many of the example applications cited above rely on the fact that the projections of a signal onto a dictionary are can be inherently sparse. Using a non-linear approximation scheme to infer a highly sparse representation of the signal will likely improve performance in these tasks.

In another novel direction known as "compressed sensing", signals are captured using a small number of linear measurements with a dictionary of random sensing elements [13, 50]. Signal reconstruction is then preformed by inferring a sparse representation that is consistent with the random measurements. This approach will enable novel sensing strategies for many different applications.

Because of the utility of sparse representations, researchers have developed a variety of sparse approximation algorithms. The two most popular types of algorithms are convex relaxation and very efficient *greedy algorithms* that often work well in practice [147]. Greedy algorithms are iterative schemes where each iteration selects the single best dictionary element to represent the residual signal without regard to which other elements will be selected in future iterations.

However, these existing algorithms have two significant drawbacks that make them unlikely for neural implementation. First, they would be difficult to implement with the parallel computational primitives used by neural systems. These algorithms are fundamentally centralized operations that would be implausible for neural systems *and* difficult to implement on a hardware platform. Second, existing algorithms do not efficiently handle time-varying signals. By focusing on sparsity at each time instant, these algorithms miss the larger picture of encoding a signal at a sequence of time steps. Time-varying signals are obviously critical for biological system behavior and increases in data collection abilities have made them increasingly important in signal processing tasks.

We introduce and study a new class of neurally plausible sparse approximation algorithms based on the principles of *thresholding* and *local competition* that addresses many of the drawbacks observed in existing methods. In our Locally Competitive Algorithms (LCAs) [136], neurons in a population continually compete with neighboring neurons using (usually one-way) lateral inhibition to calculate coefficients representing an input using an overcomplete dictionary. Unlike greedy algorithms that irrevocably select the single best dictionary element at each iteration, parallel competition allows many coefficients to become part of the representation simultaneously, perhaps even working together to suppress a coefficient that had previously been selected. Our continuous-time LCA is described by the dynamics of a system of nonlinear ordinary differential equations (ODEs) that govern the internal state (membrane potential) and external communication (short-term firing rate) of units in a neural population. We

show that each LCA corresponds to an optimal sparse approximation problem that minimizes an energy function combining reconstruction mean-squared error (MSE) and a sparsity-inducing cost function.

A sparse coding system built on a realistic neural architecture must possess three critical properties.

- Stability: the system must be well-behaved under normal operating conditions;
- Sparsity: the system must perform its primary task well, finding a good tradeoff between sparsity and representation error when coding a fixed image; and
- Regularity: the system must handle time-varying inputs efficiently, producing coefficients that reflect the smooth character of natural input signals.

In addition to being built on a neurally realistic architecture, we illustrate that LCAs also exhibit these three properties.

Specifically, LCAs can produce representations for static signals with approximately the same sparsity as other known techniques, including greedy algorithms and convex relaxation. Additionally, the LCA coefficients representing time-varying signals are much more regular than the coefficients produced by greedy algorithms. In particular, LCA coefficients display *inertia* that regularizes the time variation of the sparse coefficients for time-varying inputs by encouraging as many coefficients as possible to retain their state when the input changes. This regularity makes the coefficients much more predictable, making it easier for higher-level structures to identify and understand the changing content in the time-varying stimulus.

This chapter develops a neural architecture for LCAs, shows their correspondence to a broad class of sparse approximation problems, and details their advantages over existing sparse coding algorithms (particularly greedy algorithms). The LCA methods presented here point toward information representation methods possibly used by sensory neural systems employing sparse coding and represent a first step toward a testable neurobiological model for these systems.

5.2 Background and Related Work

5.2.1 Sparse approximation

Given an N-dimensional stimulus $\boldsymbol{x} \in \mathbb{R}^N$ (e.g., an N-pixel image), we seek a representation in terms of a dictionary $\mathcal{D}$ composed of K vectors $\{\boldsymbol{\phi}_k\}$ that span the space $\mathbb{R}^N$. In optimal sparse approximation, we seek the coefficients having the fewest number of non-zero entries by solving the minimization problem

$$\min_{\boldsymbol{a}} \|\boldsymbol{a}\|_0 \quad \text{subject to} \quad \boldsymbol{x} = \sum_{k=1}^{K} a_k \boldsymbol{\phi}_k. \tag{5.1}$$

Unfortunately, this combinatorial optimization problem is NP-hard [103].

In the signal processing community, two primary approaches are typically used to efficiently find acceptable suboptimal solutions to the optimal sparse approximation problem. The first general approach substitutes alternate sparsity measures to convexify the ℓ^0 norm. One well-known example is Basis Pursuit (BP) [19], which replaces the ℓ^0 norm with the ℓ^1 norm

$$\min_{\boldsymbol{a}} \|\boldsymbol{a}\|_1 \quad \text{subject to} \quad \boldsymbol{x} = \sum_{k=1}^{K} a_k \boldsymbol{\phi}_k. \tag{5.2}$$

BP is especially important for signal processing applications because (5.2) is a convex optimization problem equivalent to a linear program, solvable using modern interior point methods. If the signal is sparse compared to the nearest pair of dictionary elements (e.g., a vector has a solution to the optimization problem in (5.1) that is very sparse so that $\|\boldsymbol{a}\|_0 < \min_{k \neq n} \frac{1}{2}\left[1 + 1/\langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_n \rangle\right]$) BP has the same solution as the optimal sparse approximation problem [51].

The perfect reconstruction imposed by (5.2) may be too strict in many cases, especially in the presence of noise. Instead, we may make a tradeoff between reconstruction mean-squared error (MSE) and sparsity by forming an objective function that is a weighted combination of MSE and the ℓ^1 coefficient norm. For example, Basis Pursuit De-Noising (BPDN) [19], corresponds to the unconstrained optimization problem:

$$\min_{\boldsymbol{a}} \left(\left\| \boldsymbol{x} - \sum_{k=1}^{K} a_k \boldsymbol{\phi}_k \right\|_2^2 + \lambda \|\boldsymbol{a}\|_1 \right), \tag{5.3}$$

where λ is a tradeoff parameter. BPDN provides the ℓ^1-sparsest approximation for a given reconstruction quality. There are many algorithms that can be used to solve the BPDN optimization problem, with interior point-type [156] methods being the most common choice. Though the term BPDN only specifies a general optimization problem (specified in (5.3)), our use of the term BPDN specifically implies that interior point methods are used to find a numerical solution.

The second general approach employed by signal processing researchers uses iterative greedy algorithms to constructively build up a signal representation [147]. The canonical example of a greedy algorithm is known in the signal processing community as Matching Pursuit (MP) [100]. The MP algorithm is initialized with a residual $\boldsymbol{r}_0 = \boldsymbol{x}$. At the k^{th} iteration, MP finds the index of the single dictionary element best approximating the current residual signal, $\zeta_k = \arg\max_k |\langle \boldsymbol{r}_{k-1}, \boldsymbol{\phi}_k \rangle|$. The coefficient $d_k = \langle \boldsymbol{r}_{k-1}, \boldsymbol{\phi}_{\zeta_k} \rangle$ and index ζ_k are recorded as part of the reconstruction, and the residual is updated, $\boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \boldsymbol{\phi}_{\zeta_k} d_k$. After J iterations, the signal approximation using MP is given by $\widehat{\boldsymbol{x}} = \sum_{k=1}^{J} \boldsymbol{\phi}_{\zeta_k} d_k$. Though they may not be optimal in general, greedy algorithms often efficiently find good sparse signal representations in practice.

5.2.2 Sparse coding in neural systems

Recent research has found compelling evidence that V1 population responses to natural stimuli may be the result of a sparse approximation. For example, it has been shown that V1 receptive fields [39, 82] are consistent with optimizing the coefficient sparsity when encoding natural images [113]. Additionally, simultaneous V1 recordings show activity levels (corresponding to the coefficients $\{a_k\}$) becoming more sparse as neighboring units are also stimulated using natural scenes [150]. These populations are typically very overcomplete [114], allowing great flexibility in the coefficients used to represent a stimulus. Using this flexibility to pursue sparse codes might offer many advantages to sensory neural systems, including enhancing the performance of subsequent processing stages, increasing the storage capacity in associative memories, and increasing the energy efficiency of the system [114].

A neural system using sparse coding must employ an algorithm that is computable using networks of parallel computational elements. While existing sparse approximation algorithms are effective, their implementations do not correspond to plausible neural architectures. For example, the BPDN objective function in (5.3) can be minimized through direct gradient descent by the network implementation mentioned in [113]. However, this implementation has several theoretical and practical shortcomings: it requires continuous inhibition between all units with overlapping receptive fields; direct gradient descent can have difficulty minimizing objective functions that decay steeply for very small coefficients; and it lacks a natural mechanism to make small coefficients identically zero. Similarly, neural circuits implementing MP [119, 120] would require tightly coupled timing of "winner-take-all" circuits [60] (because mistakes cannot be corrected) and separate memory areas to hold the current approximation and the current residual.

Beyond implementation considerations, existing sparse approximation algorithms also do not consider the time-varying stimuli faced by neural systems. A time-varying input signal $\boldsymbol{x}(t)$ would be represented with a set of time-varying coefficients $\{a_k(t)\}$.[1] Most sparse approximation schemes have a single goal: minimize the sparsity of the representation for a fixed signal. However, higher level processing areas would also desire coefficients sequences that change in a way matching the way the stimulus changes over time. In particular, sparse coefficients should have smooth temporal variations in response to smooth changes in the image. Simply recomputing a new sparse approximation at each time step produces coefficients that contain significant temporal variations due to idiosyncrasies of the particular sparse approximation algorithm used rather than structure contained in the image, particularly when using greedy algorithms.

[1] While we denote $\boldsymbol{x}(t)$ in continuous time, we will consider it to be a piecewise constant signal over a sampling period. For example, we will often take $\boldsymbol{x}(t)$ to be a video sequence, consisting of a series of still images held fixed for 1/30 s. The reason for this discrete approximation is twofold: we simulate these continuous-time systems on a digital computer, and neural systems appear to have finite temporal resolution (e.g., the persistence of vision phenomenon at the flicker fusion rate).

In Section 5.3 we develop our LCAs, in which dictionary elements continually fight for the right to represent the stimulus. These LCAs adapt their coefficients continually over time as the input changes without having to build a new representation from scratch at each time step. This evolution induces inertia in the coefficients, regularizing the temporal variations for smoothly varying input signals. In contrast to the problems seen with purely greedy approaches, our LCAs are plausible for neural implementation and encourage both sparsity and smooth temporal variations in the coefficients as the stimulus changes.

5.2.3 Other related work

There are several sparse approximation methods that do not fit into the two primary approaches of pure greedy algorithms or convex relaxation. For example, Sparse Bayesian Learning (SBL) [146,155] maximizes the posterior distribution on the coefficients using an iterative expectation-maximization procedure [43]. SBL involves computing a matrix inverse at each iteration, making it relatively time-consuming and not amenable to neural circuit implementation. In still another direction, several researchers have parallelized greedy algorithms by selecting a set of coefficients at each iteration and correcting for the resulting correlations between coefficients [52,59,118]. These extensions do speed up convergence, but selected elements are committed to the representation and cannot be changed as in our LCAs. In another direction, many extensions of MP have been introduced [37, 126] that iteratively commit dictionary elements to the representation but progressively change the coefficient values through a computationally expensive orthogonalization step that also neurally implausible.

There are also several sparse approximation methods built on a parallel computational framework that are related to our LCAs [64, 77, 90, 127]. These approaches will be discussed in more detail after describing the LCA system architecture (see Section 5.3.5).

Finally, there have been attempts to adapt MP to handle video sequences on several fronts. As one example, the work of Zakhor and collaborators [3, 104] applied MP to coding residual video sequences after applying a motion prediction stage to remove as much variability from sequential frames as possible. The motion prediction necessary to make this coding scheme effective is a very complex algorithm, and it is difficult to imagine that such a procedure could be implemented easily on a hardware platform. As another example, Rahmoune, et al. [125] applied MP to video sequences using a dictionary of 3-D spatio-temporal basis functions adaptively constructed from the video sequence itself. This method has several drawbacks for neural implementation: temporary storage buffers would be required to remember the stimulus over the temporal window being processed and a greedy step needs to be applied at the end of each temporal window that potentially fosters large coefficient changes at these window boundaries.

5.3 Locally Competitive Algorithms for Sparse Coding

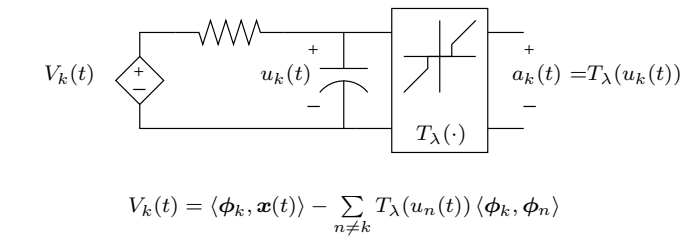
5.3.1 Architecture of locally competitive algorithms

Sensory neural systems (e.g., V1) represent a stimulus by the collective spiking activity of the neural population. Our LCAs associate each neuron with an element of the dictionary $\boldsymbol{\phi}_k \in \mathcal{D}$. The system is presented with an input image $\boldsymbol{x}(t)$, momentarily assumed to be a constant (time-varying video sequences will be considered in Section 5.6). The collection of nodes evolve according to fixed dynamics (described below) and settle on a collective output $\{a_k(t)\}$, corresponding to the short-term average firing rate of the neurons. The goal is to define the LCA system dynamics so that few coefficients have non-zero values while approximately reconstructing the input, $\widehat{\boldsymbol{x}}\,(t) = \sum_k a_k(t)\,\boldsymbol{\phi}_k \approx \boldsymbol{x}(t)$.

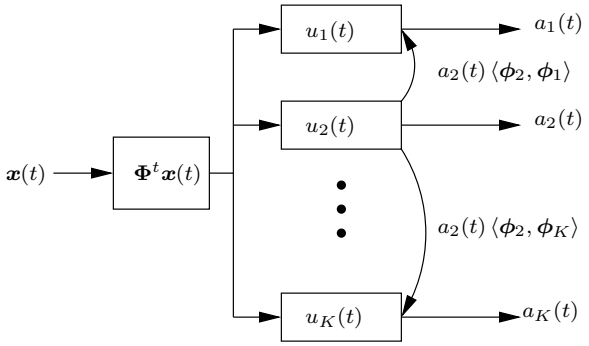
The LCA dynamics follow several properties observed in neural systems: inputs cause the membrane potential to "charge up" like a leaky integrator; membrane potentials over a threshold produce "action potentials" for extracellular signaling; and these super-threshold responses inhibit neighboring units through lateral connections. We represent each unit's sub-threshold value by a time-varying *internal state* $u_k(t)$ that evolves according to the dynamical system equation $\dot{u}_k(t) = \frac{1}{\tau}\left[m_k(t) - u_k(t)\right]$, where τ is the membrane time-constant. The unit's excitatory input current is proportional to how well the image matches with the node's receptive field, $m_k(t) = \langle \boldsymbol{\phi}_k, \boldsymbol{x}(t) \rangle$. With a constant input, these internal state variables converge to the projections of the image onto the dictionary elements. These values represent all of the information in the image, but are not sparse. We must introduce lateral interactions between the units to allow stronger nodes to inhibit weaker nodes, sparsifying the population response.

When the internal state u_k of a node becomes significantly large, the node becomes "active" and produces an output signal a_k used to represent the stimulus and inhibit other nodes. This output coefficient is the result of an activation function applied to the membrane potential, $a_k = T_\lambda(u_k)$, parameterized by the system threshold λ. Though similar activation functions have traditionally taken a sigmoidal form, we consider activation functions that operate as thresholding devices — they are essentially zero for values below λ and essentially linear for values above λ. Note that for analytical simplicity we allow positive and negative coefficients, but rectified systems could use two physical units to implement one LCA node.

The nodes best matching the stimulus will have internal state variables that charge at the fastest rates. To achieve the competition allowing these nodes to win over the weaker nodes, we have active nodes inhibit other nodes with an inhibition signal proportional to both the activity level of the active node and the similarity of the nodes' receptive fields. Specifically, the inhibition signal from the active node k to any other node n is proportional to the activity level a_k and to the inner product between the node receptive fields, measured by $\mathbf{G}_{k,n} = \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_n \rangle$. The possibility of unidirectional inhibition gives strong nodes a chance to prevent weaker nodes from



(a)



(b)

Figure 5.2: Schematic and structural illustration of LCAs. (a) LCA nodes behave as a leaky integrators, charging with a speed that depends on how well the input matches the associated dictionary element and the inhibition received from other nodes. (b) A system diagram shows the inhibition signals being sent between nodes. In this case, only node 2 is shown as being active (i.e., having a coefficient above threshold) and inhibiting its neighbors. Since the neighbors are inactive then the inhibition is one-way.

becoming active and initiating counter-inhibition, thus making the search for a sparse solution more efficient.

Putting all of the above components together, our LCA node dynamics are expressed by the non-linear ordinary differential equation (ODE)

$$\dot{u}_k(t) = \frac{1}{\tau} \left[m_k(t) - u_k(t) - \sum_{n \neq k} \mathbf{G}_{k,n} a_n(t) \right]. \tag{5.4}$$

This ODE is essentially the same form as the well-known continuous Hopfield network [80]. Figure 5.2 shows a LCA node circuit schematic and a system diagram illustrating the lateral inhibition. To express the system of coupled non-linear ODEs that govern the whole dynamic system, we represent the internal state variables in the vector $\boldsymbol{u}(t) = [u_1(t), \ldots, u_K(t)]^t$, the active coefficients in the vector $\boldsymbol{a}(t) = [a_1(t), \ldots, a_K(t)]^t = T_\lambda(\boldsymbol{u}(t))$, the dictionary elements in the columns

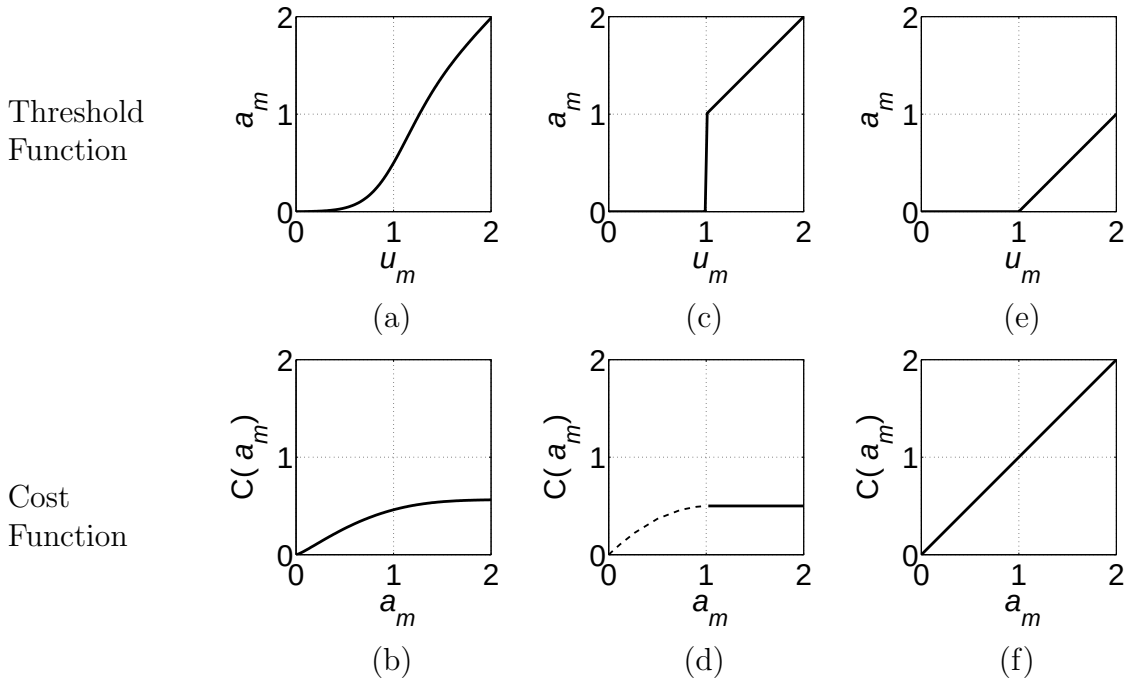


Figure 5.3: Relationship between the threshold function $T_{(\alpha,\gamma,\lambda)}(\cdot)$ and the sparsity cost function $C(\cdot)$. Only the positive half of the symmetric threshold and cost functions are plotted. (a) Sigmoidal threshold function and (b) cost function for $\gamma = 5$, $\alpha = 0$ and $\lambda = 1$. (c) The ideal hard thresholding function ($\gamma = \infty$, $\alpha = 0$, $\lambda = 1$) and the (d) corresponding cost function. The dashed line shows the limit, but coefficients produced by the ideal thresholding function cannot take values in this range (e) The ideal soft thresholding function ($\gamma = \infty$, $\alpha = 1$, $\lambda = 1$) and the (f) corresponding cost function.

of the $(N \times K)$ matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K]$ and the driving inputs in the vector $\boldsymbol{m}(t) = [m_1(t), \ldots, m_K(t)]^t = \mathbf{\Phi}^t \boldsymbol{x}(t)$. The function $T_\lambda(\cdot)$ performs element-by-element thresholding on vector inputs. The stimulus approximation is $\widehat{\boldsymbol{x}}(t) = \mathbf{\Phi}\boldsymbol{a}(t)$, and the full dynamic system equation is

$$\begin{aligned} \dot{\boldsymbol{u}}(t) = f(\boldsymbol{u}(t)) &= \frac{1}{\tau}\left[\boldsymbol{m}(t) - \boldsymbol{u}(t) - \left(\mathbf{\Phi}^t\mathbf{\Phi} - I\right)\boldsymbol{a}(t)\right], \\ \boldsymbol{a}(t) &= T_\lambda(\boldsymbol{u}(t)). \end{aligned} \tag{5.5}$$

5.3.2 Sparse approximation by locally competitive algorithms

The LCA architecture described by (5.5) solves a family of sparse approximation problems. Specifically, LCAs descend an energy function that combines the recon-

struction MSE and a sparsity-inducing cost penalty $C(\cdot)$,

$$E(t) = \frac{1}{2} \left\| \boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\,(t) \right\|^2 + \lambda \sum_k C(a_k(t))\,.$$

Though we consider this energy function for its own sake, this optimization problem can be interpreted in several ways. First, it can be viewed as the Langrangian of a constrained optimization problem, seeking solutions that minimize MSE subject to a sparsity constraint (or vice-versa, as in (5.1)). Second, minimizing this energy function can be viewed as calculating the *maximum a posteriori* (MAP) estimate of the coefficients in an inference task with a Gaussian noise model and a sparse prior distribution on the coefficients (related to the cost function $C(\cdot)$).

The specific form of the cost function $C(\cdot)$ is determined by the form of the thresholding activation function $T_\lambda(\cdot)$.[2] For a given threshold function, the cost function is specified (up to a constant) by

$$\lambda \frac{dC(a_k)}{da_k} = u_k - a_k = u_k - T_\lambda(u_k). \tag{5.6}$$

This correspondence between the thresholding function and the cost function can be seen by computing the derivative of E with respect to the active coefficients, $\{a_k\}$ (see Appendix C). If (5.6) holds, then letting the internal states $\{u_k\}$ evolve according to $\dot{u}_k \propto -\frac{\partial E}{\partial a_k}$ yields the equation for the internal state dynamics above in (5.4). As long as a_k and u_k are related by a monotonically increasing function, the $\{a_k\}$ will also descend the energy function E. This method for showing the correspondence between a dynamic system and an energy function is essentially the same procedure used by Hopfield [80] to establish network dynamics in associative memory systems.

We focus specifically on the cost functions associated with thresholding activation functions. Thresholding functions limit the lateral inhibition by allowing only "strong" units to suppress other units and forcing most coefficients to be identically zero. For our purposes, thresholding functions $T_\lambda(\cdot)$ have two distinct behaviors over their range: they are essentially linear with unit slope above threshold λ, and essentially zero below threshold. Among many reasonable choices for thresholding functions, we use a smooth sigmoidal function

$$T_{(\alpha,\gamma,\lambda)}(u_k) = \frac{u_k - \alpha\lambda}{1 + e^{-\gamma(u_k - \lambda)}}, \tag{5.7}$$

where γ is a parameter controlling the speed of the threshold transition and $\alpha \in [0, 1]$ indicates what fraction of an additive adjustment is made for values above threshold. An example sigmoidal thresholding function is shown in Figure 5.3a. We are par-

[2] The correspondence between the threshold function and the cost function was originally observed and calculated by Bruno Olshausen.

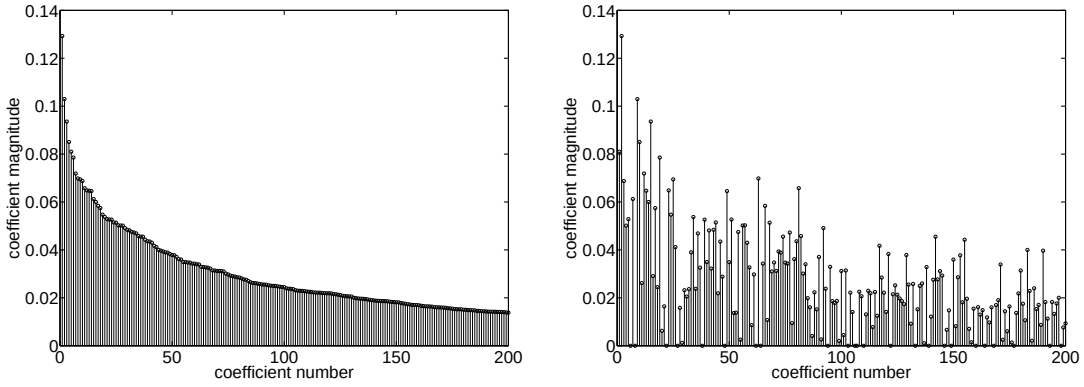


Figure 5.4: Contrasting the coefficients calculated by BPDN solvers and SLCA. (a) The top 200 coefficients from a BPDN solver sorted by magnitude. (b) The same coefficients, sorted according to the magnitude ordering of the SLCA coefficients. While there is a gross decreasing trend noticeable, the largest SLCA coefficients are not in the same locations as the largest BPDN coefficients. While the solutions have equivalent energy functions, the two sets of coefficients differ significantly.

ticularly interested in the limit of this thresholding function as $\gamma \to \infty$, a piecewise linear function we denote as the *ideal thresholding function*. In the signal processing literature, $T_{(0,\infty,\lambda)}(\cdot) = \lim_{\gamma\to\infty} T_{(0,\gamma,\lambda)}(\cdot)$ is known as a "hard" thresholding function and $T_{(1,\infty,\lambda)}(\cdot) = \lim_{\gamma\to\infty} T_{(1,\gamma,\lambda)}(\cdot)$ is known as a "soft" thresholding function [47].

Combining (5.6) and (5.7), we can integrate numerically to determine the cost function corresponding to $T_{(\alpha,\gamma,\lambda)}(\cdot)$, shown in Figure 5.3b. For the ideal threshold functions we derive a corresponding *ideal cost function*,

$$C_{(\alpha,\infty,\lambda)}(a_k) = \frac{(1-\alpha)^2\lambda}{2} + \alpha\,|a_k|\,. \tag{5.8}$$

Details of this derivation are in Appendix D. Note that unless $\alpha = 1$ the ideal cost function has a gap because active coefficients cannot take all possible values, $|a_k| \notin [0, (1-\alpha)\,\lambda]$ (i.e., the ideal thresholding function is not invertible).

5.3.3 Special case: Soft-thresholding locally competitive algorithm (SLCA)

As we see in Section 5.3.2, a LCA can optimize a variety of different sparsity measures depending on the choice of thresholding function. One special case that we will look at in detail is the soft thresholding function, corresponding to $\alpha = 1$ and shown graphically in Figures 5.3e and 5.3f. The soft-thresholding locally competitive

algorithm (SLCA) applies the ℓ^1 norm as a cost function on the active coefficients,

$$C_{(1,\infty,\lambda)}(a_k) = |a_k|.$$

Thus, the SLCA minimizes a combination of the ℓ^2 reconstruction error and ℓ^1 sparsity penalty, meaning that it is simply another solution method for the general BPDN problem described in Section 5.2.1. Despite minimizing the same convex energy function, SLCA and BPDN solvers will find different sets of coefficients, as illustrated in Figure 5.4. The connection between soft-thresholding and BPDN is well-known in the case of orthonormal dictionaries [19], and recent results [56] have given some justification for using soft-thresholding in overcomplete dictionaries. The SLCA provides another formal connection between the soft-thresholding function and the ℓ^1 cost function.

Though BPDN uses the ℓ^1 norm as its sparsity penalty, we often expect many of the resulting coefficients to be identically zero (especially when $K \gg N$). However, like many numerical methods, interior point BPDN solvers will drive coefficients *toward* zero but will never make them identically zero. While an ad hoc threshold could be applied to the results of a BPDN solver, the SLCA has the advantage of incorporating a natural thresholding function that keeps coefficients identically zero during the computation unless they become large enough to go active. In other words, while BPDN solvers often start with many non-zero coefficients and try to force coefficients down, the SLCA starts with all coefficients equal to zero and only lets a few grow up. This advantage is especially important for neural systems that must expend energy for non-zero values throughout the entire computation.

5.3.4 Special case: Hard-thresholding locally competitive algorithm (HLCA)

Another important special case is the hard thresholding function, corresponding to $\alpha = 0$ and shown graphically in Figures 5.3c and 5.3d. Using the relationship in (5.6), we see that this hard-thresholding locally competitive algorithm (HLCA) applies an ℓ^0-like cost function by using a constant penalty regardless of magnitude,

$$C_{(0,\infty,\lambda)}(a_k) = \frac{\lambda}{2}\,\mathbf{1}_{(|a_k|>\lambda)}.$$

As with the SLCA, the HLCA also has connections to known sparse approximation principles. If node k is fully charged, the inhibition signal it sends to other nodes would be exactly the same as the update step when the k^{th} node is chosen in the MP algorithm. However, due to the continuous competition between nodes before they are fully charged, the HLCA will not find the same sparse representation as MP in general.

As a demonstration of the power of competitive algorithms over greedy algorithms, consider a canonical example used to illustrate the shortcomings of greedy

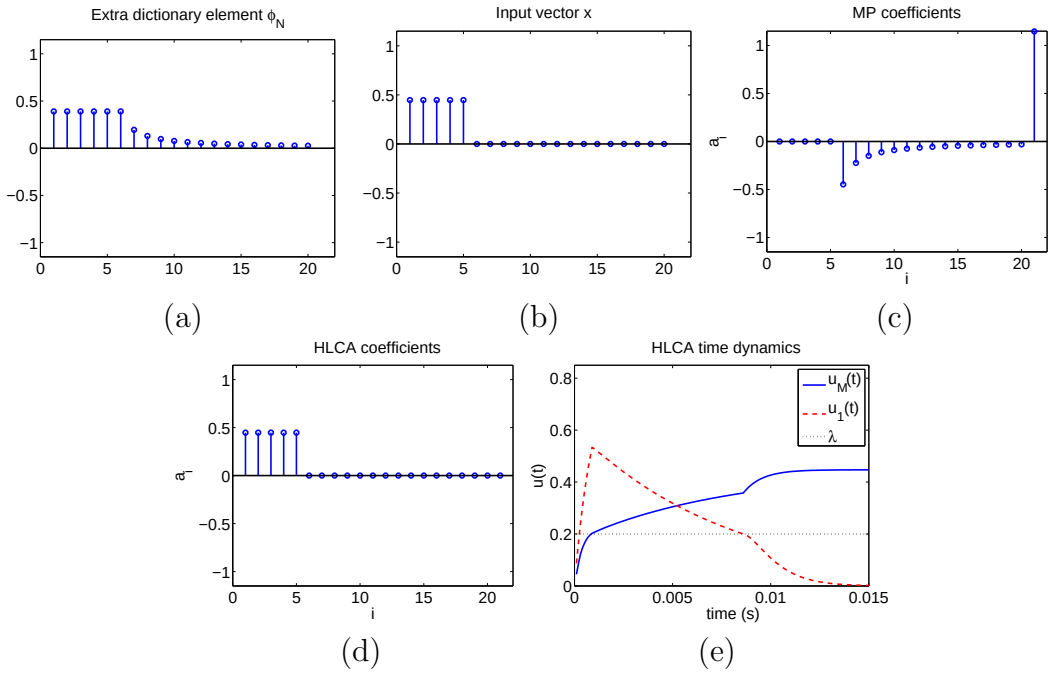


Figure 5.5: A pathologic example where greedy MP algorithms perform poorly but HLCA can find the optimal coefficients. (a) The dictionary in this example has one "extra" element that consists of decaying combinations of all other dictionary elements. (b) The input vector has a sparse representation in just a few dictionary elements. (c) MP initially chooses the "extra" dictionary element, preventing it from finding the optimally sparse representation (coefficients shown after 100 iterations). (d) In contrast, the HLCA system finds the optimally sparse coefficients. (e) The time-dynamics of the HLCA system illustrate its advantage. The "extra" dictionary element is the first node to activate, followed shortly by the nodes corresponding to the optimal coefficients. The collective inhibition of the optimal nodes causes the "extra" node to die away.

algorithms [19,45]. For this example, specify a positive integer $\widetilde{K} < N$ and construct a dictionary $\mathcal{D}$ with $K = N + 1$ elements to have the following form:

$$\boldsymbol{\phi}_k = \begin{cases} \boldsymbol{e}_k & \text{if } k \leq N \\ \sum_{n=1}^{\widetilde{K}} \kappa \boldsymbol{e}_n + \sum_{n=\widetilde{K}+1}^{N} \left(\kappa / (n - \widetilde{K}) \right) \boldsymbol{e}_n & \text{if } k = N + 1, \end{cases}$$

where $\boldsymbol{e}_k$ is the canonical basis element (i.e., it contains a single 1 in the k^{th} location) and κ is a constant to make the vectors have unit norm. In words, the dictionary is essentially the canonical basis but with one "extra" element that is a decaying combination of all other elements (illustrated in Figure 5.5, with $N = 20$ and $\widetilde{K} = 5$). The input signal is sparsely represented in the first $\widetilde{K}$ dictionary ele-

ments, $\boldsymbol{x} = \sum_{k=1}^{\widetilde{K}} \frac{1}{\sqrt{\widetilde{K}}} \boldsymbol{e}_k$. The first MP iteration chooses $\boldsymbol{\phi}_K$, introducing a residual with decaying terms. Even though $\boldsymbol{x}$ has an exact representation in $\widetilde{K}$ elements, MP iterates forever trying to atone for its bad initial choice. This pathological example highlights the main danger in purely greedy algorithms such as MP: optimal local choices cannot be corrected even if they induce undesirable behavior throughout the rest of the coefficients. In contrast, the LCA competition allows the system to correct bad choices and eliminate them if they are sufficiently globally undesirable. In this example, the HLCA initially activates the K^{th} node. The collective inhibition from nodes $1, \ldots, \widetilde{K}$ causes this extra node to eventually die away leaving the optimal set of coefficients.

5.3.5 Closely related methods

Fischer, et al. [64] developed an algorithm that has a similar flavor to our LCAs but with significant differences. Their iterative procedure generates new coefficients by selecting a subset of old coefficients and adding adjustments to retain perfect reconstruction. Coefficients are selected by thresholding a separate set of state variables representing coefficient values over previous iterations. While this system does have parallel characteristics, keeping a set of auxiliary state variables not directly related to the current coefficients requires two separate neural populations. Additionally, this algorithm would not map naturally to time-varying inputs because the state variables would be reset with each new input.

Kingsbury and Reeves [90] have described a sparsity-inducing iterative scheme for complex wavelets that is similar to a discrete approximation of the HLCA. Their algorithm is described by a system of equations that has two primary differences from the HLCA system equations. First, instead of the "charging up" behavior used by the HLCA, their algorithm starts with projecting the stimulus onto the basis set and then iteratively altering those coefficients. If this algorithm were initialized with all zero coefficients, then the first iteration would jump directly to the projection coefficients and continue from there. It is not immediately clear if this strategy enables the algorithm to adapt smoothly to tracking stimulus changes. The second significant difference is that their algorithm uses both the dictionary and the dual set (i.e., the analysis and synthesis vectors in the frame representation [20]). Calculating the dual vectors is computationally expensive and recalculation must be performed whenever the dictionary changes. Such a system is less resilient to adaptations or failures. This algorithm is also very similar to an iterative scheme described by Herrity et al. [77], though the Herrity algorithm does not employ the dual dictionary.

Rehn and Sommer [127] have also recently described a neurally plausible sparse coding mechanism that is similar to our approach. This algorithm is described by a system of equations that have three primary differences from the HLCA system equations. First, as with the work in [90], this algorithm also does not use a "charging up" approach and but rather works by modifying the projection coefficients. Second,

this system uses a variable threshold for activating each unit, making it easier for neurons to become active when they have a larger driving input (i.e., neurons with a larger input have a lower threshold to become active). Finally, this system uses a constant amount of inhibition that is not related to the current activity level of the active neuron.

Figueiredo, et al. [63] have introduced a projection pursuit algorithm for solving the BPDN optimization problem. Their iterative (discrete) algorithm also shares some common features with our SLCA because it uses alternating projections between the signal and the coefficient domain. The main operation of their algorithm uses an adaptive step size to use when following the gradient of the energy function plus a recovery step to push their solution back into the constraint set of their particular problem statement.

Each of these methods shares some similarities with the LCAs we have described. However, the LCAs are unique in that they are the only method exhibiting the following combination of characteristics: they employ a charging process in conjunction with a thresholding operation to naturally enforce ℓ^0-sparsity, they are parallel and continuous time systems by design to facilitate (hardware or neural) analog implementation, they quantifiably minimize (at least locally) a specific objective function, and they naturally address the problems of representing time-varying signals.

5.4 Property I: Stability

To be a neurally plausible sparse coding mechanism, LCAs must exhibit several critical properties: stability, sparsity, and regularity for time-varying stimuli. We focus our assessment of these properties on the HLCA both because it yields the most interesting results and because it is notationally the cleanest to discuss. In general, the analysis principles we use will apply to all LCAs through straightforward (through perhaps laborious) extensions.

Any proposed neural system must remain well-behaved under normal conditions. Linear systems theory has an intuitive and uniform notion of stability: well-behaved systems produce outputs with bounded energy as long as the input also has bounded energy [66]. Unfortunately, no such unifying concept of stability exists for non-linear systems [88]. Instead, non-linear systems are characterized in a variety of ways, including their input-output relationship and their behavior near an equilibrium point $\boldsymbol{u}^*$ where $f(\boldsymbol{u}^*) = 0$.

While there is no universal input-output stability test for general non-linear systems, we observe that the LCA system equations are linear and only change when a node crosses threshold (from above or below). A branch of control theory specifically addresses these *switched systems* [40]. To express the HLCA system as a switched system, we define $\mathcal{K} \subseteq [1, \ldots, K]$ as the current set of active nodes (i.e., $k \in \mathcal{K}$ if $|u_k(t)| \geq \lambda$). We also define a $(K \times K)$ selection matrix $\mathbf{S}_\mathcal{K}$ as being all zeros

except for ones on the diagonal corresponding to the active nodes,

$$[\mathbf{S}_{\mathcal{K}}]_{k,n} = \begin{cases} 1 & \text{if } k = n \text{ and } k \in \mathcal{K} \\ 0 & \text{if } k \neq n \text{ or } k \notin \mathcal{K}. \end{cases}$$

Defining the system matrix $\mathbf{A}_{\mathcal{K}} = \frac{1}{\tau}\left[\left(I - \mathbf{\Phi}^t\mathbf{\Phi}\right)\mathbf{S}_{\mathcal{K}} - I\right]$, the HLCA is written as a *switched linear system*,[3]

$$\dot{\boldsymbol{u}}\left(t\right) = \frac{1}{\tau}\mathbf{\Phi}^t\boldsymbol{x}(t) + \mathbf{A}_{\mathcal{K}}\boldsymbol{u}(t)\,.$$

In addition to drawing on some existing results in the switched linear systems literature, this viewpoint of the LCAs will be valuable in several of the succeeding sections.

5.4.1 LCA stability criteria

The various stability notions we consider for the LCA will depend on a common criteria. Define $\mathcal{K}_{\boldsymbol{u}(t)} \subseteq [1,\ldots,K]$ as the set of nodes that are above threshold in the internal state vector $\boldsymbol{u}(t)$, $\mathcal{K}_{\boldsymbol{u}(t)} = \{k : |u_k(t)| \geq \lambda\}$. We say that the LCA meets the *stability criteria* if for all time t the set of active vectors $\{\boldsymbol{\phi}_k\}_{k\in\mathcal{K}_{\boldsymbol{u}(t)}}$ is linearly independent. This criteria will be important for several later results. In the meantime, it makes some intuitive sense that this condition is important to a LCA: if a collection of linearly dependent nodes are active simultaneously, the nodes could have active (and possibly growing) coefficients with a net effect of no contribution to the reconstruction.

Satisfying the stability criteria comes down to two questions: how likely are small subsets of dictionary elements to be linearly dependent, and how likely is the LCA to activate such a subset? Small subsets of dictionary elements are unlikely to be linearly dependent unless the dictionary is designed with this property. This result has been quantified recently for random dictionaries in work related to the field of "compressive sensing" [148] (c.f. [49] for an equivalent problem). Though these results are for random dictionaries, they do give insight for large ambient signal dimensions and provide analytic tools to explore a specific dictionary.

Regardless of the properties of the dictionary, sparse coding systems are actively trying to select dictionary subsets so that they can use many *fewer* coefficients then the dimension of the signal space, $\left|\mathcal{K}_{\boldsymbol{u}(t)}\right| \ll N \ll K$. While the LCA lateral inhibition signals discourage linear dependent sets from activating, the stability criteria could be violated when a collection of nodes becomes active too quickly, before inhibition can take effect. In practice, this situation could occur when the threshold is too low compared to the system time constant. We expect (and our simulations confirm) that

[3] All LCA systems using ideal thresholding functions can be written as a similar switched system, but the thresholding functions with additive correction terms require a cumbersome definition of proxy state variables that we omit here.

under normal operating conditions with biologically relevant parameters, the LCAs satisfy the stability criteria.

5.4.2 Equilibrium points

In a LCA presented with a static input, we look to the steady-state response (where $\dot{\boldsymbol{u}}(t) = 0$) to determine the coefficients. The character of the equilibrium points $\boldsymbol{u}^*$ ($f(\boldsymbol{u}^*) = 0$) and the system's behavior in a neighborhood around an equilibrium point provides one way to ensure that a system is well-behaved. With the exception of the SLCA, all LCAs based on the ideal thresholding functions have non-convex energy functions and therefore settle on solutions that are local minima. The number and placement of these equilibrium points in this setting will be critical to understanding whether the system will produce distinct solutions depending on the initial conditions or whether the solutions are all grouped together.

Consider the ball around an equilibrium point $\mathcal{B}_\epsilon(\boldsymbol{u}^*) = \{\boldsymbol{u} : \|\boldsymbol{u} - \boldsymbol{u}^*\| < \epsilon\}$. The tools of Lyapunov stability [88] are frequently employed to answer an intuitive question: if the system is perturbed within this ball, does it then run away, stay where it is, or get attracted back? Specifically, a system is said to be *locally asymptotically stable* [4] at an equilibrium point $\boldsymbol{u}^*$ if one can specify an $\epsilon > 0$ such that

$$\boldsymbol{u}(0) \in \mathcal{B}_\epsilon(\boldsymbol{u}^*) \implies \lim_{t\to\infty} \boldsymbol{u}(t) = \boldsymbol{u}^*.$$

Previous research [22, 97, 158] has used the tools of Lyapunov functions [88] to study a Hopfield network [80] similar to the LCA architecture, but all of these analyses make assumptions that do not encompass the ideal thresholding functions. For example, [22, 97, 158] have all addressed the stability of a dynamic system similar to both a Hopfield network [80] and the present CA architecture. Unfortunately, all of these analyses make specific assumptions that do not apply to the CA systems. In particular, the ideal thresholding functions cause complications because they are not monotone increasing (i.e., they are not invertible) and they are not continuously differentiable. There is a generalized notion of Lyapunov functions [4] that are applicable to non-differentiable threshold functions, but it will not be the most efficient tool in this circumstance.

In Appendix E.1 we show that as long as the stability criteria is met, the HLCA:

- has a finite number of equilibrium points;
- has equilibrium points that are almost certainly isolated (no two equilibrium points are arbitrarily close together); and
- is almost certainly locally asymptotically stable for every equilibrium point.

The conditions that hold "almost certainly" are true as long as none of the equilibria have components identically equal to the threshold, ($u_k^* \neq \lambda, \forall k$), which holds with

overwhelming probability. With a finite number of isolated equilibria, we can be confident that the HLCA steady-state response is a distinct set of coefficients representing the stimulus. Asymptotic stability also implies a notion of robustness, guaranteeing that the system will remain well-behaved even under perturbations (Theorems 2.8 and 2.9 in [4]).

5.4.3 Robustness to system imperfections

For a system to be considered well-behaved, it must also degrade gracefully in the face of confounding situations, such as communication or implementation corruption. These robustness notions are intimately related the concept of local stability. Asymptotic stability implies a notion of robustness to two different types of perturbations: system perturbations and state perturbations.

We restate two robustness theorems from [4] without their original proofs. The system robustness theorem guarantees that if the real system function is close to the desired system function, then the equilibrium point will still be asymptotically stable.

System robustness theorem (Theorem 2.8 in [4]). *Let $\boldsymbol{u}^*$ be a locally asymptotically stable equilibrium point for the system*

$$\dot{\boldsymbol{u}} = f(\boldsymbol{u})\,,$$

so that $f(\boldsymbol{u}^*) = 0$. Then there exists $\epsilon > 0$ and a positive definite continuous function $\widetilde{g}(\boldsymbol{u}) : \mathcal{B}_\epsilon(\boldsymbol{u}^*) \to \mathbb{R}$ such that the following holds: if $g(\boldsymbol{u})$ is any function satisfying the conditions:

- *$g(\boldsymbol{u})$ is continuous*
- *$g(\boldsymbol{u}^*) = 0$*
- *$\|f(\boldsymbol{u}) - g(\boldsymbol{u})\| < \widetilde{g}(\boldsymbol{u}) \quad \forall \boldsymbol{u} \in \mathcal{B}_\epsilon(\boldsymbol{u}^*)$*

then $\boldsymbol{u}^*$ is also a locally asymptotically stable equilibrium point for the system

$$\dot{\boldsymbol{u}} = g(\boldsymbol{u})\,.$$

In a slightly different variation on system robustness, the state robustness theorem guarantees that if the real state values are close to the desired state values, then the equilibrium points will still be asymptotically stable.

State robustness theorem (Theorem 2.9 in [4]). *Let $\boldsymbol{u}^*$ be a locally asymptotically stable equilibrium point for the system*

$$\dot{\boldsymbol{u}} = f(\boldsymbol{u}^*)\,,$$

so that $f(\boldsymbol{u}) = 0$. Then there exists $\epsilon > 0$ and a positive definite continuous function $\widetilde{g}(\boldsymbol{u}) : \mathcal{B}_\epsilon(\boldsymbol{u}^*) \to \mathbb{R}$ such that the following holds: if $g(\boldsymbol{u})$ is any function satisfying the conditions:

- $g(\boldsymbol{u})$ *is continuous*
- $g(\boldsymbol{u}^*) = 0$
- $\|g(\boldsymbol{u})\| < \widetilde{g}(\boldsymbol{u}) \quad \forall \boldsymbol{u} \in \mathcal{B}_\epsilon(\boldsymbol{u}^*)$

then $\boldsymbol{u}^*$ *is also a locally asymptotically stable equilibrium point for the system*

$$\dot{\boldsymbol{u}} = f(\boldsymbol{u} + g(\boldsymbol{u}))\,.$$

These robustness theorems guarantee that the LCAs are not so fragile that they require infinite precision in their communications or their implementations to remain stable. Much more work would have to be done to characterize the practical consequences of these robustness results on system implementation.

5.4.4 Input-output stability

In physical systems it is important that the energy of both internal and external signals remain bounded for bounded inputs. One intuitive approach to ensuring output stability is to examine the energy function E. We show in Appendix E.2 that for non-decreasing threshold functions, the energy function is non-increasing ($\frac{d}{dt}E(t) \leq 0$) for fixed inputs. While this is encouraging, it does not guarantee input-output stability. To appreciate this effect, note that the HLCA cost function is constant for nodes above threshold — nothing explicitly keeps a node from growing without bound once it is active.

Appendix E.1 shows that if the set of active nodes $\mathcal{K}$ meet the stability criteria, then the corresponding subsystem matrix $\mathbf{A}_\mathcal{K}$ will be input-output stable (i.e., $\mathbf{A}_\mathcal{K}$ will only have eigenvalues in the left half-plane). Unfortunately, even if every subsystem is individually stable the switched system can still be unstable [40]. Results from the switched systems literature indicate that input-output stability can be guaranteed if the individual linear subsystems are stable, and the system doesn't switch "too fast" between these subsystems [78].

Specifically, define τ_D to be the average dwell time between changes to the set of active nodes $\mathcal{K}_{\boldsymbol{u}(t)}$. For switched linear systems, sufficient (but not necessary) conditions for input-output stability require each subsystem to be asymptotically stable and that the system doesn't switch "too often" [78]. We restate here a theorem from switched system theory in language corresponding to the LCAs.

Average dwell time theorem (Theorem 2 combined with Lemma 1 of [78]). *Given a collection of system matrices* $\mathbf{A}_{\boldsymbol{u}(t)}$ *and a positive constant* ω_0 *such that* $\mathbf{A}_{\boldsymbol{u}(t)} + \omega_0 I$ *is asymptotically stable for all* t*, then, for any* $\omega \in [0, \omega_0]$*, there is a finite constant* τ_D^* *such that as long as* $\tau_D \geq \tau_D^*$*, the switched system has a bounded internal state for piecewise constant input signals* $\boldsymbol{x}(t)$*:*

$$\left(\int_0^t e^{2\omega\tau} \|\boldsymbol{u}(\tau)\|^2 \, d\tau\right)^{1/2} \leq \kappa_1 \left(\int_0^t e^{2\omega\tau} \|\boldsymbol{x}(\tau)\|^2 \, d\tau\right)^{1/2} + \kappa_2 \|\boldsymbol{x}(0)\|\,,$$

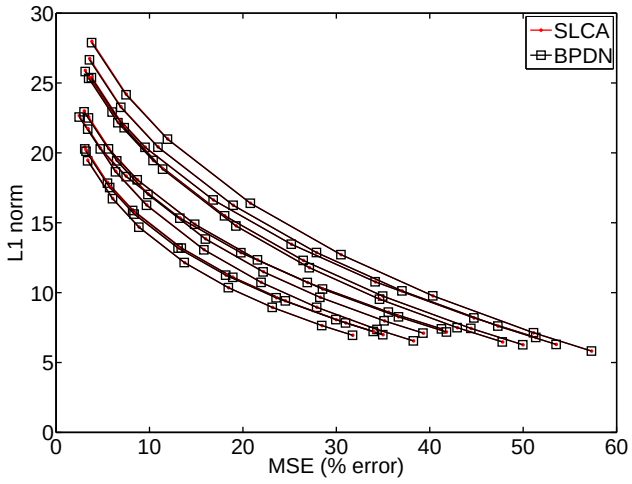


Figure 5.6: The curve depicting the tradeoff between MSE and ℓ^1 -sparsity for a series of standard test images shows that SLCA and BPDN are finding equivalent solutions in this sparsity measure. Each line on the plot indicates the tradeoff between MSE and ℓ^1 coefficient norm as λ is varied. The results for SLCA and BPDN overlap exactly, illustrating that the systems are finding equivalent minima of the energy function.

where κ_1 and κ_2 are finite constants. Similar statements can be made using ℓ^∞ norms instead of ℓ^2 norms.

The average dwell time theorem guarantees that the LCA system will have bounded energy outputs for bounded energy inputs (and initial conditions) as long as each subsystem is asymptotically stable and τ_D is not too small.

In Appendix E.2 we show that the HLCA linear subsystems are individually stable if and only if the stability criteria are met. Therefore, the HLCA is input-output stable as long as nodes are limited in how fast they can change states. We expect that the infinitely fast switching condition is avoided in practice either by the physical principles of the system implementation or through an explicit hysteresis in the thresholding function.

5.5 Property II: Sparsity and Representation Error

Viewing the sparse approximation problem through the lens of rate-distortion theory [7], the most powerful algorithm produces the lowest reconstruction MSE for a given sparsity. When the sparsity measure is the ℓ^1 norm, there is little variation in the solution quality: solving the convex optimization principle in (5.3) produces a solution with minimum MSE for that ℓ^1 norm. Therefore, even though SLCA produces different coefficients from an interior point BPDN solver, they will both

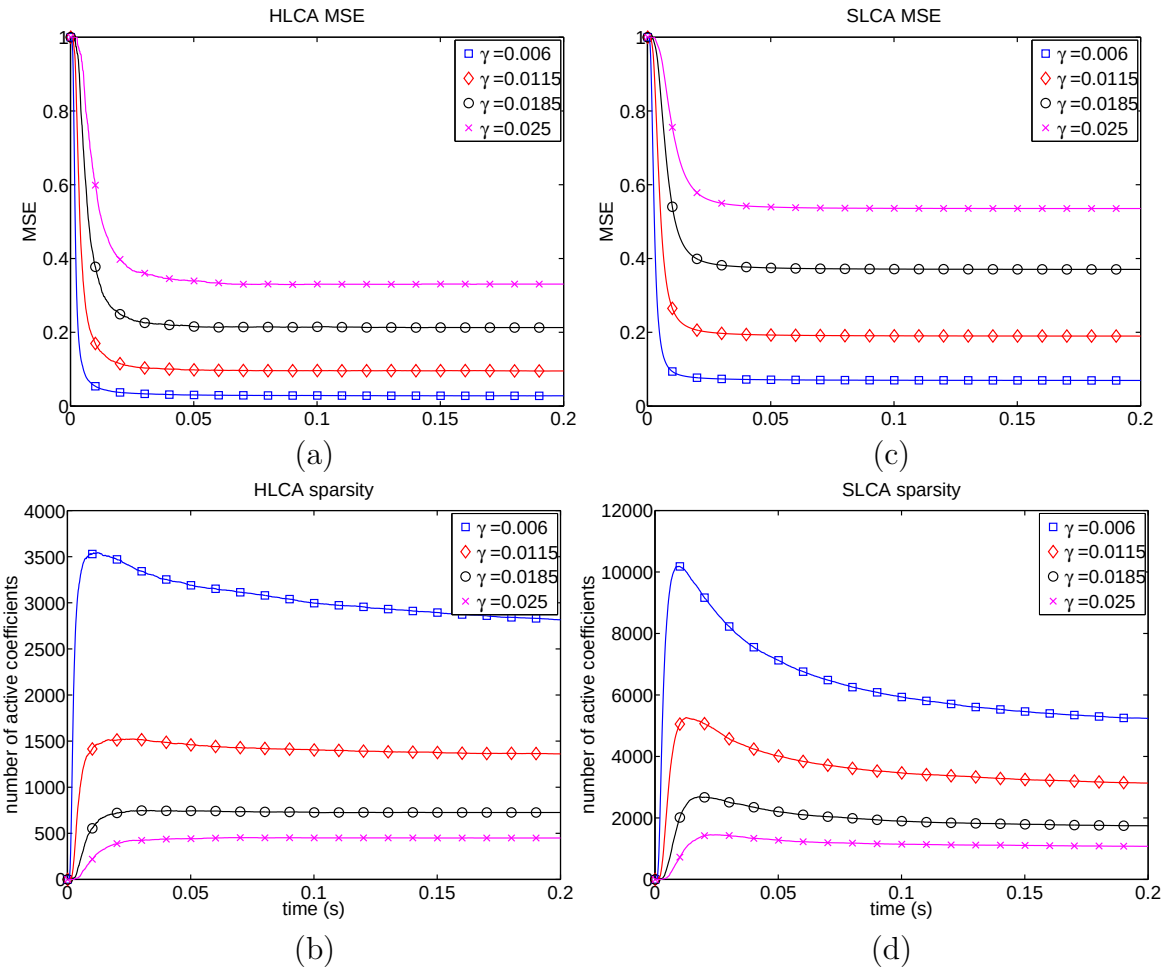


Figure 5.7: The time response of the HLCA and SLCA (τ = 10 ms) for a single fixed image patch. (a) The MSE decay and (b) the ℓ^0 sparsity for HLCA. (c) The MSE decay and (d) the ℓ^0 sparsity for SLCA. The error converges within 1-2 time constants and the sparsity often approximately converges within 3-4 time constants. In some cases sparsity is reduced with a longer running time.

achieve optimality with regard to ℓ^1 sparsity (demonstrated in Figure 5.6). The SLCA is unique in that it is the only LCA corresponding to a convex energy function.

Despite the analytic appeal of the ℓ^1 norm as a sparsity measure, many systems concerned with energy minimization (including neural systems) likely have an interest in minimizing the ℓ^0 norm of the coefficients. The HLCA is appealing because of its ℓ^0-like sparsity penalty, but this objective function is not convex and the HLCA may find a local minimum. We will show that while HLCA cannot guarantee the ℓ^0 sparsest solution, it produces coefficients that demonstrate comparable sparsity to MP for natural images.

5.5.1 Reconstruction error of steady-state coefficients

Insight about the HLCA reconstruction fidelity comes from rewriting the LCA system equation

$$\dot{\boldsymbol{u}}\left(t\right) = \frac{1}{\tau}\left[\boldsymbol{\Phi}^{t}\left(\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\left(t\right)\right) - \boldsymbol{u}(t) + T_{(\alpha,\infty,\lambda)}(\boldsymbol{u}(t))\right]. \tag{5.9}$$

For a constant input, HLCA equilibrium points ($\dot{\boldsymbol{u}}\left(t\right) = 0$) occur when the residual error is orthogonal to active nodes and balanced with the internal state variables of inactive nodes.

$$\langle \boldsymbol{\phi}_k, \boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\left(t\right)\rangle = \begin{cases} u_k(t) & \text{if } |u_k| \leq \lambda \\ 0 & \text{if } |u_k| > \lambda \end{cases}.$$

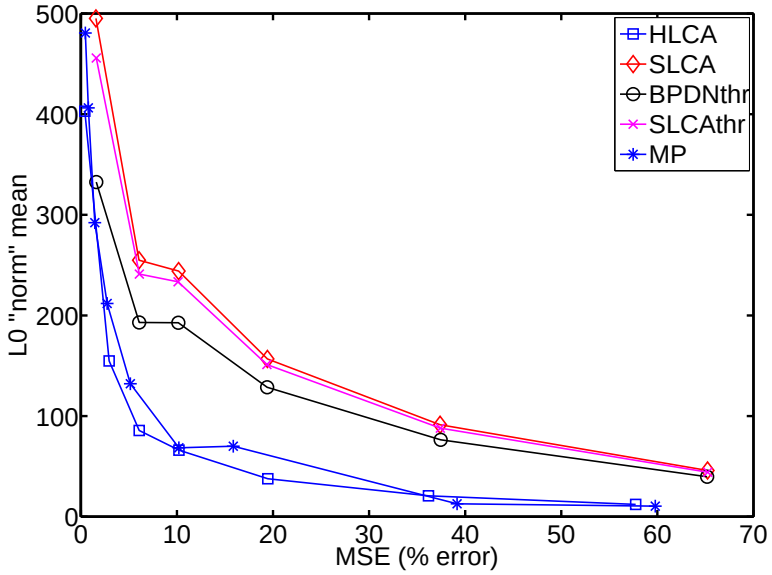
Therefore, when HLCA converges the coefficients will perfectly reconstruct the component of the input signal that projects onto the subspace spanned by the final set of active nodes. Using standard results from frame theory [20], we can bound the HLCA reconstruction MSE in terms of the set of inactive nodes

$$\|\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\left(t\right)\|^2 \leq \frac{1}{\eta_{\min}} \sum_{k \notin \mathcal{K}_{\boldsymbol{u}(t)}} |\langle \boldsymbol{\phi}_k, \boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\left(t\right)\rangle|^2 \leq \frac{\left(K - |\mathcal{K}_{\boldsymbol{u}(t)}|\right)\lambda^2}{\eta_{\min}},$$

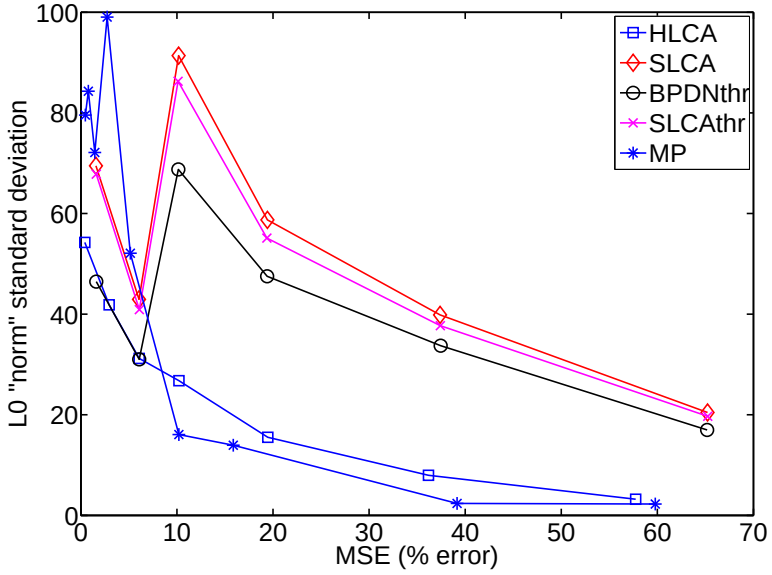
where $\eta_{\min}$ is the minimum eigenvalue of $\left(\boldsymbol{\Phi}\boldsymbol{\Phi}^{t}\right)$.

5.5.2 Efficiency of steady-state coefficients

Though the HLCA is not guaranteed to find the globally optimal ℓ^0 sparsest solution, we would like to ensure that the system is still being reasonably efficient. While the system nonlinearity makes it impossible to analytically determine the LCA steady-state coefficients, it is possible to rule out some sets as *not* being possible. For example, let $\mathcal{K} \subseteq [1, \ldots, K]$ be an arbitrary set of active coefficients. Using linear systems theory we can calculate the steady-state response $\widetilde{\boldsymbol{u}}^{\mathcal{K}} = \lim_{t\to\infty} \boldsymbol{u}(t)$ assuming that $\mathcal{K}$ stays fixed. If $|\widetilde{u}_k^{\mathcal{K}}| < \lambda$ for any $k \in \mathcal{K}$ or if $|\widetilde{u}_k^{\mathcal{K}}| > \lambda$ for any $k \notin \mathcal{K}$, then $\mathcal{K}$ cannot describe the set of active nodes in the steady-state response and we call it *inconsistent*. We show in Appendix F that when the stability criteria are met, the following statement is true for the HLCA: *If $\boldsymbol{x} = \boldsymbol{\phi}_k$, then any set of active coefficients $\mathcal{K}$ with $k \in \mathcal{K}$ and $|\mathcal{K}| > 1$ is inconsistent.* In other words, the HLCA may use the k^{th} node or a collection of other nodes to represent $\boldsymbol{x}$, but it cannot use a combination of both. This result extends intuitively beyond one-sparse signals: each component in an optimal decomposition is represented by either the optimal node or another collection of nodes, but not both. While not necessarily finding the optimal representation, the system does not needlessly employ both the optimal and extraneous nodes.



(a)



(b)

Figure 5.8: The tradeoff between MSE and ℓ^0 -sparsity for normalized (32×32) patches from a standard set of test images. For a given MSE range, we plot the mean (a) and standard deviation (b) of the ℓ^0 sparsity.

5.5.3 Numerical simulations to verify sparsity

We have also verified numerically that the LCAs achieve a combination of error and sparsity comparable with known methods. In these experiments we simulated the LCAs until steady-state on normalized (32×32) test image patches. We used a biologically plausible membrane time constant of $\tau = 10$ ms [38] and a dictionary consisting of the bandpass band of a steerable pyramid [141] with one level and four orientation bands (i.e., the dictionary is approximately four times overcomplete).

The test image patches are created by filtering out the lowpass and residual bands of the steerable pyramid, so that any given test image can be fully described using the dictionary.[4] Figure 5.7 shows the time evolution of the reconstruction MSE and ℓ^0 sparsity for SLCA and HLCA responding to an individual image, and Figure 5.8 shows the mean steady-state tradeoff between ℓ^0 sparsity and MSE. For comparison, we also plotted the results obtained from using MP, a standard BPDN solver followed by thresholding to enforce ℓ^0 sparsity (denoted "BPDNthr") and SLCA with the same threshold applied (denoted "SLCAthr"). Most importantly, note that the HLCA and MP are almost identical in their sparsity-MSE tradeoff. Though the connections between the HLCA and MP were pointed out in Section 5.3.4, these are very different systems and there is no reason to expect them to produce the same coefficients. Additionally, note that the SLCA is producing coefficients that are nearly as ℓ^0-sparse as what we can achieved by thresholding the results of a BPDN solver even though the SLCA keeps most coefficients zero throughout the calculation.

5.6 Property III: Regularity for Time-varying Stimuli

Biological sensory systems are faced with constantly changing stimuli due to both external movement and internal factors (e.g., organism movement, eye saccades, etc.). Neurally plausible sparse coding systems must therefore efficiently handle time-varying stimuli such as video sequences. Any sparse approximation scheme suitable for hardware implementation must produce efficient coefficient sequences for video inputs.

5.6.1 Strategies for encoding time-varying signals

One strategy for applying a typical sparse approximation scheme to a video sequence is to independently find a set of coefficients representing each frame using a 2-D (spatial) dictionary, thereby producing a sequence of coefficients (i.e., a coefficient time series) for each dictionary element. However, optimizing a sparsity criteria at each frame can cause the selected coefficients to vary significantly from one frame to

[4] We eliminate the lowpass band because it accounts for a large fraction of the total image energy in just a few dictionary elements and it is unlikely that much gain could be achieved by sparsifying these coefficients.

the next even when there are relatively small changes in the input. Greedy algorithms are especially prone to generating such "brittle" representations. This temporal irregularity is undesirable for higher level systems trying to interpret the content of the scene because characteristics of a smoothly varying stimulus would not be reflected in the encoding.

In video compression, irregularity makes it more difficult to find efficient encodings for the video sequence. Encoding a video sequence into a bit-stream involves explicitly indicating both the location and values of non-zero coefficients. When the locations of the non-zero coefficients are highly variable from frame to frame, the locations of the active coefficients are unpredictable and more bits are needed to encode that video. While neural coding is fundamentally different from video compression, this same type of regularity is likely to be helpful for higher level systems understanding the content of the stimulus. Regularity in the locations of active coefficients can let the system focus on the changing values of a few locations rather than constantly monitoring the appearance and disappearance of many locations.

Several methods are used in video compression schemes to try and avoid this irregularity. For example, video compression algorithms often apply motion compensation, where the content of one frame is predicted from the previous frame. In this scheme, only the prediction residual is encoded at each frame, increasing the regularity of the coefficients when the stimulus is undergoing predictable changes. Motion prediction, in conjunction with coefficient selection algorithms (including MP), have been successful in video compression. However, effective motion prediction algorithms are very complex and it is difficult to imagine them being explicitly implemented in analog hardware.

Other methods to work around coefficient irregularity typically include processing several frames of video simultaneously. One such scheme employs 3-D dictionary elements, having both temporal and spatial components. Another related idea is the recent work by Rahmoune, et. al [125], which uses MP with a dynamically built dictionary on motion compensated frames. The dictionary is built to try and employ representation elements that are best matches for the time-varying signal. While effective as video coders, the complexity of these schemes along with the need to hold several frames in memory simultaneously makes hardware implementation impractical.

5.6.2 Coefficient inertia

In contrast to the approaches described above, LCAs naturally produce smoothly changing outputs in response to smoothly changing time-varying inputs. Assuming that the system time constant τ is faster than the temporal changes in the stimulus, the LCA will evolve to capture the stimulus change and converge to a new representation. While local minima in an energy function are typically problematic, the LCAs can use these local minima to find coefficients that are "close" to their previous coefficients even if they are not optimally sparse. While permitting suboptimal co-

efficient sparsity, this property allows the LCA to exhibit *inertia*0 that smooths the coefficient sequences.

The inertia property exhibited in LCAs can be seen by focusing on a single node in the system equation (5.9):

$$\dot{u}_k(t) = \frac{1}{\tau} \begin{cases} \langle \boldsymbol{\phi}_k, (\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\,(t)) \rangle - u_k(t) & \text{when } |u_k(t)\,| < \lambda \\ \langle \boldsymbol{\phi}_k, (\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}\,(t)) \rangle - \alpha\lambda & \text{when } |u_k(t)\,| \geq \lambda. \end{cases}$$

A new residual signal drives the coefficient higher but suffers an additive penalty. Inactive coefficients suffer an increasing penalty as they get closer to threshold while active coefficients only suffer a constant penalty $\alpha\lambda$ that can be very small (e.g., the HLCA has $\alpha\lambda = 0$). This property induces a "king of the hill" effect: when a new residual appears, active nodes move virtually unimpeded to represent it while inactive nodes are penalized until they reach threshold. This inertia encourages inactive nodes to remain inactive unless the active nodes cannot adequately represent the new input.

5.6.3 Numerical simulations to verify regularity

To illustrate this inertia, we applied the LCAs to a sequence of 144×144 pixel, bandpass filtered, normalized frames from the standard "foreman" test video sequence with the same experimental setup described in Section 5.5. The LCA input is switched to the next video frame every (simulated) 1/30 seconds. The results are shown in Figure 5.9, along with comparisons to MP and BPDN applied independently on each frame. The changing coefficient locations are nodes that either became active or inactive at each frame. Mathematically, the number of changing coefficients at frame n is: $\left|\mathcal{K}_{\boldsymbol{u}(n-1)} \oplus \mathcal{K}_{\boldsymbol{u}(n)}\right|$, where $\oplus$ is the "exclusive OR" operator and $\boldsymbol{u}(n)$ are the internal state variables at the end of the simulation for frame n.

This simulation highlights that the HLCA uses approximately the same number of active coefficients as MP but is much more efficient in how it chooses those coefficients. The HLCA is significantly more likely to re-use active coefficient locations from the previous frame without making significant sacrifices in the sparsity of the solution. This difference is highlighted when looking at the ratio of the number of changing coefficients to the number of active coefficients, $\left|\mathcal{K}_{\boldsymbol{u}(n-1)} \oplus \mathcal{K}_{\boldsymbol{u}(n)}\right| \,/\, \left|\mathcal{K}_{\boldsymbol{u}(n)}\right|$. MP has a ratio of 1.7, meaning that MP is finding almost an entirely new set of active coefficient locations for each frame. The HLCA has a ratio of 0.5, meaning that it is changing approximately 25% of its coefficient locations at each frame. SLCA and BPDNthr have approximately the same performance, with regularity falling between HLCA and MP. Though the two systems can calculate different coefficients, the convexity of the energy function appears to be limiting the coefficient choices enough so that SLCA cannot smooth the coefficient time series substantially more than BPDNthr.

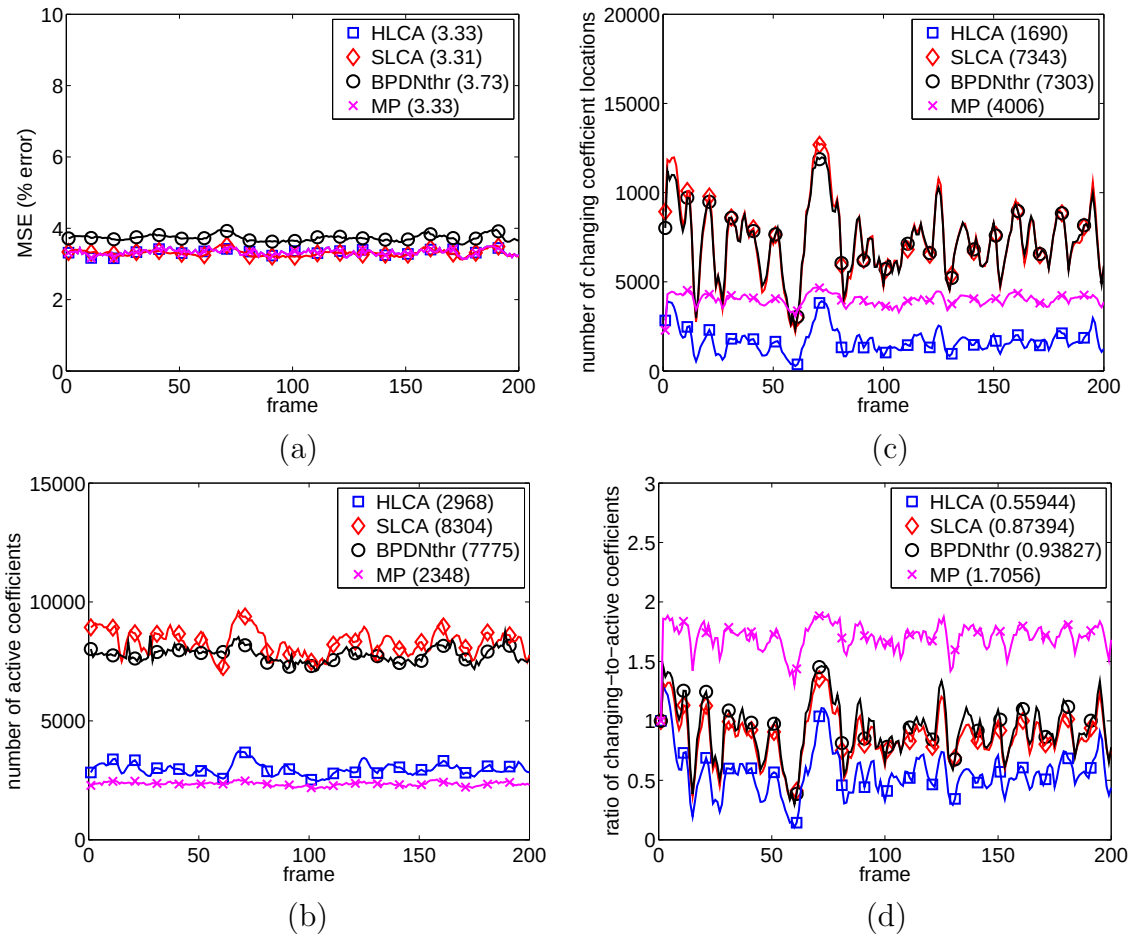
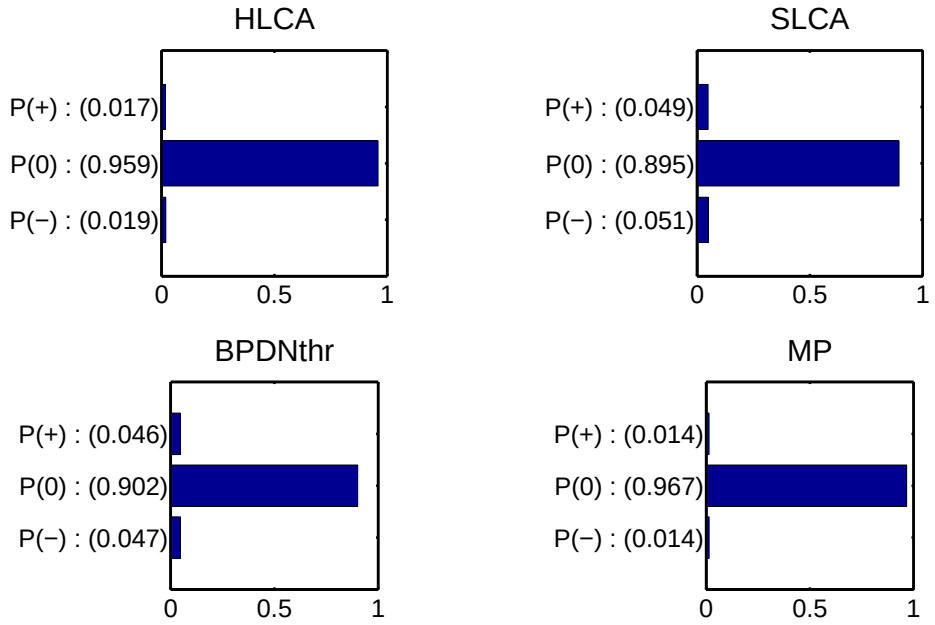
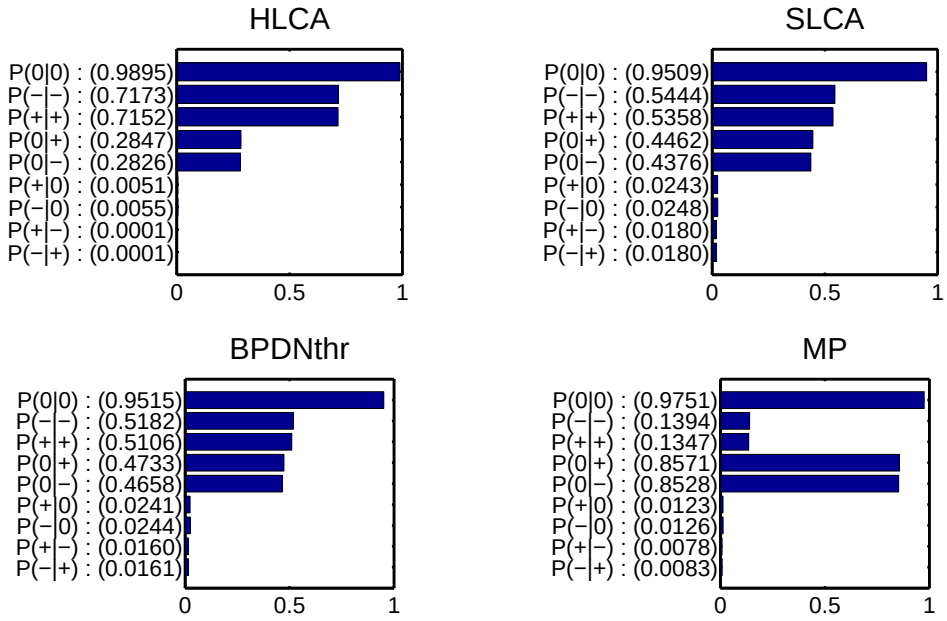


Figure 5.9: The HLCA and SLCA systems simulated on 200 frames of the "foreman" test video sequence. For comparison, MP coefficients and thresholded BPDN coefficients are also shown. Average values for each system are notated in the legend. (a) Per-frame MSE for each coding scheme, designed to be approximately equal. (b) The number of active coefficients in each frame. (c) The number of changing coefficient locations for each frame, including the number of inactive nodes becoming active and the number of active nodes becoming inactive. (d) The ratio of changing coefficients to active coefficients. A ratio near 2 (such as with MP) means that almost 100% of the coefficient locations are new at each frame. A ratio near 0.5 (such as with HLCA) means that approximately 25% of the coefficients are new at each frame.



(a)



(b)

Figure 5.10: Marginal and transition probabilities characterizing the Markov chain behavior of the coefficient states. (a) The marginal probabilities denoting the fraction of the time coefficients spent in the three states: negative, zero and positive ($-$, 0, and $+$). (b) The transition probabilities denoting the probability of a node in one state transitioning to another state on the next frame. For example, $P(0|+)$ is the probability that a node with an active positive coefficient will be inactive (i.e., zero) in the next frame.

5.6.4 Quantifying temporal coefficient regularity

The simulation results indicate that the HLCA is producing time series coefficients that are much more regular than MP. This regularity is visualized in Figure 5.11 by looking at the time-series of example HLCA and MP coefficients. Note that though the two coding schemes produce roughly the same number of non-zero entries, the HLCA does much better than MP at clustering the values into consecutive runs of positive or negative values. This type of smoothness better reflects the regularity in the natural video sequence input.

We can quantify this increased regularity by examining the Markov state transitions. Specifically, each coefficient time-series is Markov chain [107] with three possible states at frame n:

$$\varsigma_k(n) = \begin{cases} - & \text{if } u_k(n) < -\lambda \\ 0 & \text{if } -\lambda \leq u_k(n) \leq \lambda \\ + & \text{if } u_k(n) > \lambda. \end{cases}$$

Figure 5.10 shows the marginal probabilities $P(\cdot)$ of the states and the conditional probabilities $P(\cdot|\cdot)$ of moving to a state given the previous state. The HLCA and MP are equally likely to have non-zero states, but the HLCA is over five times more likely than MP to have a positive coefficient stay positive ($P(+|+)$). Also, though the absolute probabilities are small, MP is roughly two orders of magnitude more likely to have a coefficient swing from positive to negative ($P(-|+)$) and vice-versa ($P(-|+)$).

To quantify the regularity of the active coefficient locations we calculate the entropy [27] of the coefficient states at frame n conditioned on the coefficient states at frame (n-1):

$$\begin{aligned} H(\varsigma_k(n)\,|\varsigma_k(n-1)) = &-P(+)\,[P(-|+) + P(0|+) + P(+|+)] \\ &- P(0)\,[P(-|0) + P(0|0) + P(+|0)] \\ &- P(-)\,[P(-|-) + P(0|-) + P(+|-)]\,, \end{aligned} \tag{5.10}$$

plotted in Figure 5.11. This conditional entropy indicates how much uncertainty there is about the status of the current coefficients given the coefficients from the previous frame. Note that the conditional entropy for MP is almost double the entropy for the HLCA, while SLCA is again similar to BPDNthr. The principle contributing factor to the conditional entropy appears to be the probability a non-zero node remains in the same state (i.e., $P(+|+)$ and $P(-|-)$). To illustrate, Figure 5.11 shows the change in conditional entropy is almost linear with varying $P(+|+)$ (assuming $P(-|-) = P(+|+)$ and all other transition probabilities are kept fixed).

The substantial decrease in the conditional entropy for the HLCA compared to MP quantifies the increased regularity in time-series coefficients due to the inertial properties of the LCAs. The HLCA in particular encourages coefficients to maintain

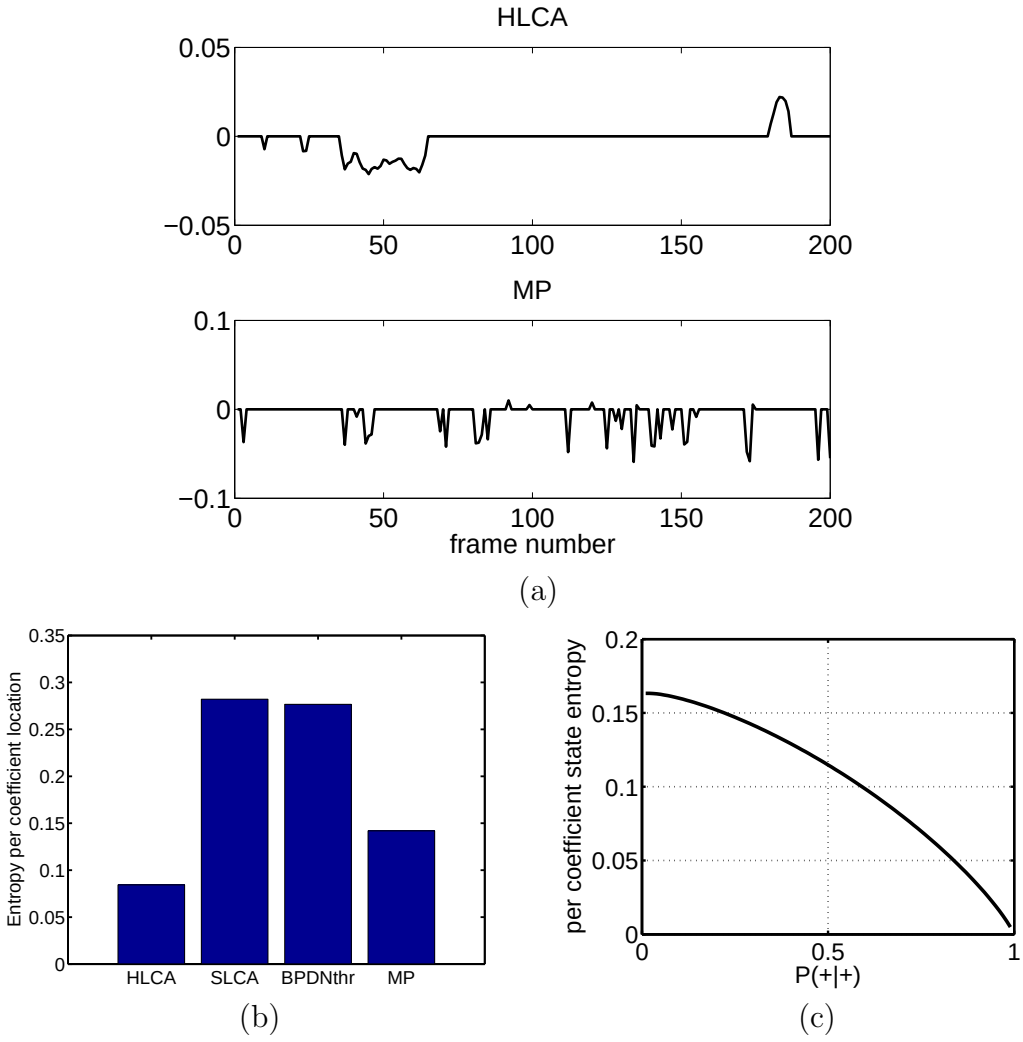


Figure 5.11: The LCA coefficients display inertia that regularizes (smooths) their behavior and makes them more predictable. (a) An example time-series coefficient for the HLCA and MP (top and bottom, respectively) encodings for the test video sequence. HLCA clusters non-zero entries together into longer runs while MP switches more often between states. (b) The empirical conditional entropy of the coefficient states (−,0,+) during the test video sequence. (c) The conditional entropy is calculated analytically while varying $P(+|+)$ and equalizing all other transition probabilities to the values seen in HLCA and MP. The tendency of a system to group non-zero states together is the most important factor in determining the entropy.

their present state (i.e., active or inactive) if it is possible to find an adequate stimulus representation. While some sparsity may be sacrificed in this strategy, the smoothness induced in the coefficients by grouping active states together in time better reflects the character of the natural time-varying stimuli and could be useful for higher-level computations.

5.7 Summary and Future Work

5.7.1 Summary of contributions

Sparse approximation is an important paradigm in neural coding, though the plausible mechanisms to achieve these codes have remained unknown. We have proposed an architecture for a class of locally competitive algorithms that solve a series of sparse approximation problems (including BPDN as a special case). These LCAs are neurally plausible in the sense that they can be implemented using a parallel network of simple elements that match well with the known neurobiology of sensory cortical areas such as V1. Though these LCA systems are non-linear, we have shown that they are well-behaved under nominal operating conditions.

While the LCA systems (other than SLCA) are not generally guaranteed to find a globally optimal solution to their energy function, we have proven that the systems will be efficient in a meaningful sense. In practice, we have seen that the LCAs are very efficient in producing coefficients for natural images. The SLCA system produces coefficients with sparsity levels comparable to BPDN solvers, but uses a more efficient and natural physical implementation than these solvers. Perhaps most interestingly, the HLCA produces coefficients with almost identical sparsity as MP. This is significant because greedy methods such as MP are widely used in signal processing practice because of their efficiency, but HLCA offers a much more natural neural implementation.

LCAs are particularly appropriate for time-varying data such as video sequences. The LCA ODE not only encourages sparsity but also introduces an inertia into the coefficient time-series that we have quantified using both raw counts of changing coefficient location and through the conditional entropy of the coefficient states. By allowing suboptimal sparsity in exchange for more regularity in the set of active coefficients, the LCAs produce smoother coefficient sequences that better reflect the structure of the time-varying stimulus. This property could prove valuable for higher levels of analysis that are trying to interpret the sensory scene from a set of sparse coefficients.

5.7.2 Future directions in neurobiology

The current limitations of neurophysiological recording mean that exploring the sparse coding hypothesis must rely on testing specific proposed mechanisms. Though the LCAs we have proposed appear to map well to known neural architectures, they

still lack the detail necessary to be experimentally testable. We will continue to build on this work by mapping LCAs to a detailed neurobiological population coding model that can produce verifiable predictions. While the model captured by the LCA framework is undoubtedly incorrect in many ways, it provides a starting point for refining our ideas about processing strategies used by cortical sensory areas.

5.7.3 Future directions in hardware implementation

By using simple computational primitives, LCAs also have the benefit of being plausible for analog hardware implementation. On an implementation front, we envision several possible applications for LCAs in imaging systems. Implementing LCAs in analog VLSI as the front end for a digital camera could produce systems that are more time and energy efficient. Such a system could directly produce compressed coefficients using an analog hardware front-end *before* digitization. This system would not waste resources digitizing data that will eventually be thrown away in a compression stage.

Similarly for video, an analog implementation of an LCA could serve as a front end capturing each individual image frame. The resulting coefficient time-series would already be compressed spatially, and the regularity of the coefficients puts them in a form that is easily compressible temporally. Such a system could produce (using analog hardware) compressed video comparable to results currently achievable only with software on a digital computer.

There are several significant challenges to implementing LCAs in analog hardware. The difficulty inherent in producing non-rectilinear receptive fields (e.g., CCD "arrays" with non-square, overlapping regions), the lack of ideal thresholding devices, and the quantity of inter-node wiring will certainly require deviations from the ideal system equations we have described. As we move toward implementations, it is important that we characterize how the system performance is affected by compromises in the ideal system operation. The robustness results implied by the local stability described in Section 5.4.2 are a beginning step in that direction.

5.7.4 Future directions in signal processing algorithms

Beyond the specific implementation suggested by LCAs, the combination of sparsity and regularity induced in these competitive algorithms could illuminate critical representational principles that enable visual perceptual tasks, including pattern recognition, source separation and object tracking. The ability to infer sparse representations of data is already central to many signal and image analysis tasks, including denoising, super-resolution, in-painting, compressed sensing, object recognition, source separation and location, and classification. Additionally incorporating the temporal structure of image sequences into the representation could yield additional improvements to our performance on these tasks for video sequences.

Chapter 6
Conclusions

To improve our design of signal processing systems and our understanding of sensory neural systems, we must understand the consequences of the information representations used by these systems. Neural systems appear to use representations with two characteristics that are viewed apprehensively by the signal processing community: redundancy and distributed processing. Though it is not currently clear what advantages neural systems acquire from this particular form of information representation, it is important to understand the advantages these characteristics could impart.

The contributions of this thesis illuminate the consequences of redundant representations for distributed sensing systems from three different viewpoints.

- Chapter 3 explores the potential robustness of redundant representations to corrupted coefficients. We extend known results for additive noise corruption of frame coefficients to include noise reduction under point process noise sources. We also use a new theory of fusion frames to characterize the fundamental noise reduction limits of redundant representations under distributed processing constraints. These results show that distributed systems pay a maximum noise reduction penalty depending on the specific way that representation elements are grouped together into local processing structures. Additionally, we demonstrate that distributed groupings forming a tight fusion frame (with tight local frames) pay no penalty in noise reduction ability over the centralized reconstruction strategies.

- Chapter 4 explores distributed strategies for turning sensed data into action in the context of wireless sensor and actuator networks. We show that linear control laws can be implemented exactly in redundant systems without a centralized controller. We also exhibit an adaptive strategy for pruning communication links to save node power. We further propose and solve an optimal power scheduling problem that accounts for the variable quality and communication cost of each sensor-actuator link to optimally allocate bitrates to each communication channel. Our solution to this resource allocation problem displays several orders of magnitude in network power savings over uniform allocation strategies in illustrative example.

- Chapter 5 explores leveraging the flexibility of redundant representations to achieve a sparse coding scheme based on distributed neurally plausible archi-

tectures. In contrast to the known sparse approximation algorithms in the signal processing literature, the locally competitive algorithms (LCAs) we developed can be implemented in parallel analog systems (hardware or neural) while still achieving the same MSE-sparsity tradeoff of centralized algorithms. Furthermore, the LCAs naturally process time-varying signals (e.g., video), producing much smoother coefficient sequences than existing algorithms.

The collective results of this thesis present several insights about redundant representations. As is known in the signal processing community, redundancy does provide a measure of robustness to corruption in the representation. Additionally, redundancy complicates the distributed processing of a data representation where no centralized controller is used. However, redundancy provides a measure of flexibility that a critically sampled system does not have. We have demonstrated that even under distributed processing constraints, this flexibility can be leveraged to produce coefficients that not only represent the signal but also have other desirable characteristics. In this thesis, we have concentrated on sparsity (explicitly) and smoothness in time (implicitly) as target characteristics, but one can imagine that many different qualities could be advantageous for a system trying to understand signal content.

The results we described also present several insights about distributed processing requirements. We have illustrated that, as expected, distributed processing requirements result in more complicated decision-making structures and in a loss of noise-reduction ability. However, when enforcing distributed processing constraints onto problems that already have functioning centralized algorithms (e.g., sparse approximation) we uncover solutions that open up new implementation avenues and algorithms that display a fundamentally different character than their centralized counterparts.

We believe that this combination of results points toward signal processing solutions that can leverage distributed redundant representations to their advantage. Rather than viewing these characteristics as a necessary evil that crop up in modern problem settings, our results indicate that there are substantial benefits to these types of representations. While we don't understand all of the advantages these representations provide neural systems, we hope that further work will elucidate a greater variety of paths that signal processing solutions can explore.

This thesis represents a starting point for exploring the advantages and overcoming the challenges of distributed redundant representations. Each chapter has indicated specific future work that will follow the results we have reported. In general, the results of this thesis center around the very abstract model of sensing systems described in Section 2.1.2. While useful for analysis, this model is still far short of an accurate model of many specific sensing systems. Many of the future research directions we highlight involve making this model more applicable to specific man-made and biological systems. We believe that there are significant gains that can be made by better understanding the role of distributed redundant representations in a wide variety of sensing systems.

Appendix A
Bounding the Trace of a Quadratic Form

Consider the symmetric square matrices $\mathbf{A}$ and $\mathbf{B}$. We would like to bound the trace of the quadratic form $\mathrm{Tr}\left[\mathbf{ABA}\right]$ in terms of the eigenstructure of $\mathbf{A}$. The trace operator is invariant under cyclic permutations, $\mathrm{Tr}\left[\mathbf{ABA}\right] = \mathrm{Tr}\left[\mathbf{AAB}\right]$. Assume the symmetric square matrix $\mathbf{A}$ has (orthonormal) eigenvectors $\boldsymbol{v}_i$ and corresponding eigenvalues ν_i. We can use standard manipulations to calculate an upper bound,

$$\begin{aligned}
\mathrm{Tr}\left[\mathbf{ABA}\right] = \mathrm{Tr}\left[\mathbf{AAB}\right] &= \mathrm{Tr}\left[\left(\sum_i \nu_i \boldsymbol{v}_i \boldsymbol{v}_i^t\right)\left(\sum_k \nu_k \boldsymbol{v}_k \boldsymbol{v}_k^t\right)\mathbf{B}\right] \\
&= \mathrm{Tr}\left[\sum_{i,k} \nu_i \nu_k \boldsymbol{v}_i \boldsymbol{v}_i^t \boldsymbol{v}_k \boldsymbol{v}_k^t \mathbf{B}\right] \\
&= \mathrm{Tr}\left[\sum_i \nu_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^t \mathbf{B}\right] \\
&\leq \nu_{\max}^2 \mathrm{Tr}\left[\sum_i \boldsymbol{v}_i \boldsymbol{v}_i^t \mathbf{B}\right] \\
&= \nu_{\max}^2 \mathrm{Tr}\left[\mathbf{B}\right]
\end{aligned}$$

Similarly, we can calculate a lower bound based on the minimum eigenvalue $\mathrm{Tr}\left[\mathbf{ABA}\right] \geq \nu_{\min}^2 \mathrm{Tr}\left[\mathbf{B}\right]$.

Appendix B
Noise Reduction in a Tight Fusion Frame

Consider a collection of subspaces $\{W_i\}$ that constitute a tight fusion frame with bounds $A^s = B^s$. The subspace W_i is spanned by a local tight frame with analysis operator $\mathbf{\Phi}_i$ and identical frame bounds $A_i = A_{\min} = B_i = B_{\max}$. The pseudoinverse operator for the local frame is given by $\mathbf{\Phi}_i^\dagger = \frac{1}{A_{\min}}\mathbf{\Phi}_i^t$.

We assume that the frame coefficients in each subspace are corrupted with independent identically distributed noise having variance σ^2. As a result, the noise vector $\boldsymbol{n}_i$ corrupting the reconstructed projection into W_i has a covariance matrix equal to $\mathbf{\Gamma}_i = \frac{\sigma^2}{A_{\min}^2}\mathbf{\Phi}_i^t\mathbf{\Phi}_i$. The covariance of the aggregate noise vector $\widetilde{\boldsymbol{n}} = \sum_i \boldsymbol{n}_i$ is

$$\widetilde{\mathbf{\Gamma}} = \sum_i \mathbf{\Gamma}_i = \frac{\sigma^2}{A_{\min}^2}\sum_i \mathbf{\Phi}_i^t\mathbf{\Phi}_i = \frac{\sigma^2 A}{A_{\min}^2}\mathbf{I} = \frac{\sigma^2 A^s}{A_{\min}}\mathbf{I},$$

where the last two equalities follow because the total collection of vectors is a tight frame with bound $A = A^s A_{\min}$. When the fusion frame is tight, the composition of the analysis and synthesis operators is a scaled identity operator, $(\mathbf{W}^*\mathbf{W})^{-1} = \frac{1}{A^s}\mathbf{I}$. Equation (3.5) then yields the total reconstruction MSE from the subspace projections

$$\frac{\mathcal{E}\left[\left\|\widehat{\boldsymbol{x}}^d - \boldsymbol{x}\right\|^2\right]}{N} = \frac{\mathrm{Tr}\left[\widetilde{\mathbf{\Gamma}}\right]}{NA^{s2}} = \frac{\sigma^2 A}{A^{s2}A_{\min}^2} = \frac{\sigma^2}{A}.$$

When reconstructing directly from the global (tight) frame with frame bound A, the result given in (3.3) directly gives the MSE for a centralized reconstruction strategy $\frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}}^c - \boldsymbol{x}\|^2\right]}{N} = \frac{\sigma^2}{A}$. Note that this is equivalent to the MSE for the distributed reconstruction case discussed above. When there are K vectors in the global frame, $A = \frac{N}{K}$ and the reconstruction MSE can be written as

$$\frac{\mathcal{E}\left[\left\|\widehat{\boldsymbol{x}}^d - \boldsymbol{x}\right\|^2\right]}{N} = \frac{\mathcal{E}\left[\|\widehat{\boldsymbol{x}}^c - \boldsymbol{x}\|^2\right]}{N} = \frac{\sigma^2 N}{K}.$$

Appendix C
Relating LCA Cost Functions and Threshold Functions

To see the correspondence between a particular choice of a threshold function $T_\lambda(\cdot)$ and the sparsity-inducing cost function $C(\cdot)$, we begin by assuming we want to minimize an energy function of the form:

$$\begin{aligned} E &= \frac{1}{2}\left\|\boldsymbol{x}-\widehat{\boldsymbol{x}}\right\|^2+\lambda\sum_k C(a_k) \\ &= \frac{1}{2}\left(\boldsymbol{x}^t\boldsymbol{x}-2\boldsymbol{m}^t\boldsymbol{a}+\boldsymbol{a}^t\boldsymbol{\Phi}^t\boldsymbol{\Phi}\boldsymbol{a}\right)+\lambda\sum_k C(a_k)\,. \end{aligned}$$

For simplicity, we suppress the time variable in the notation. To find the changes in the active coefficients $\{a_k\}$ that will most significantly minimize the energy function, we take the derivative of the energy function with respect to the active coefficients,

$$\frac{dE}{da_k}=-m_k+\sum_n \mathbf{G}_{k,n}a_n+\lambda\frac{dC(a_k)}{da_k}=-m_k+\sum_{n\neq k}\mathbf{G}_{k,n}a_n+a_k+\lambda\frac{dC(a_k)}{da_k}, \quad \text{(C.1)}$$

where we assume the vectors are unit-norm $\left\|\boldsymbol{\phi}_k\right\|^2=1$. Looking back to the dynamic system in (5.4),

$$\dot{u}_k=\frac{1}{\tau}\left[m_k-u_k-\sum_{n\neq k}\mathbf{G}_{k,n}a_n\right],$$

we can see that the dynamics on the internal state variables are proportional to the derivative of the energy function in (C.1), $\dot{u}_k\propto-\frac{dE}{da_k}$, if the active coefficients are related to the internal state variables by

$$u_k=a_k+\lambda\frac{dC(a_k)}{da_k}.$$

Appendix D
LCA Cost Functions Corresponding to Ideal Thresholding Functions

The sigmoidal threshold function specified in (5.7) is invertible, meaning that active coefficients can be related back to their underlying state variables, $u_k = T^{-1}_{(\alpha,\gamma,\lambda)}(a_k)$, though not in closed form. For notational simplicity and without losing generality, we will assume in this section positive coefficients ($a_k > 0$). Though the ideal thresholding functions are not technically invertible, we can find the limit of the inverse function:

$$T^{-1}_{(\alpha,\infty,\lambda)}(a_k) = \lim_{\gamma\to\infty} T^{-1}_{(\alpha,\gamma,\lambda)}(a_k) = \begin{cases} \lambda & \text{if} \quad a_k < (1-\alpha)\lambda \\ \lambda + a_k - (1-\alpha)\lambda & \text{if} \quad a_k \geq (1-\alpha)\lambda. \end{cases}$$

Using the result from Appendix C,

$$\lambda \frac{dC(a_k)}{da_k} = u_k - a_k = T^{-1}_{(\alpha,\gamma,\lambda)}(a_k) - a_k,$$

we integrate to find the ideal cost function

$$\begin{aligned} C(a_k) &= \frac{1}{\lambda}\int_0^{a_k} \left(T^{-1}_{(\alpha,\infty,\lambda)}(x) - x\right) dx \\ &= \frac{1}{\lambda}\left(\int_0^{a_k} (\lambda - x)\, dx + \int_{(1-\alpha)\lambda}^{a_k} (x - (1-\alpha)\lambda)\, dx\right) \\ &= \alpha a_k + \frac{\lambda\,(1-\alpha)^2}{2}. \end{aligned}$$

Appendix E
LCA Stability

E.1 Equilibrium Points

The current set of active coefficients, denoted by $\mathcal{K}$, can take only finitely many possible configurations. We further limit the possible values of $\mathcal{K}$ by only allowing sets satisfying the stability criteria (active nodes must not form linearly dependent subdictionaries). We also assume that a given fixed input $\boldsymbol{x}$ induces equilibrium points $\boldsymbol{u}^*$ that do not have any components identically equal to threshold $u_k^* \neq \lambda$. This condition appears true with overwhelming probability, and implies that there exists $r > 0$ such that $|u_k^*| - r \geq \lambda$ for all $k \in \mathcal{K}$ and $|u_k^*| + r \leq \lambda$ for all $k \notin \mathcal{K}$.

For a given $\mathcal{K}$, linear systems theory indicates that the system

$$\dot{\boldsymbol{u}} = \mathbf{A}_{\mathcal{K}}\boldsymbol{u}$$

has a single equilibrium point at the origin (i.e., is *asymptotically stable*) only if $\mathbf{A}_{\mathcal{K}}$ has all negative eigenvalues (i.e., in the general case, all eigenvalues are in the left half-plane) [66]. The matrix $\mathbf{A}_{\mathcal{K}}$ has no positive eigenvalues, so we must show that it is full rank ($\mathbf{A}_{\mathcal{K}}\boldsymbol{u} \neq 0, \forall \boldsymbol{u} \in \mathbb{R}^K$). We begin by determining the nullspace $\mathcal{N}(\cdot)$ of the composite matrix $\boldsymbol{\Phi}^t\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}$. The nullspace of $\boldsymbol{\Phi}^t$ is empty, $\mathcal{N}\left(\boldsymbol{\Phi}^t\right) = \emptyset$, because $\text{span}\{\boldsymbol{\phi}_k\} = \mathbb{R}^N$. Because the collection $\{\boldsymbol{\phi}_k\}_{k\in\mathcal{K}}$ is linearly independent and the matrix $\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}$ consists of only those selected vectors on the columns, $\mathcal{N}(\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}) = \text{span}\{\boldsymbol{e}_k\}_{k\notin\mathcal{K}}$, where $\boldsymbol{e}_k$ are the canonical basis elements. Therefore the composite matrix also has a nullspace of $\mathcal{N}\left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}\right) = \text{span}\{\boldsymbol{e}_k\}_{k\notin\mathcal{K}}$. Without losing generality, we assume that the first $|\mathcal{K}|$ entries are active, $\mathcal{K} = 1, \ldots, |\mathcal{K}|$. Consider first the case when all non-trivial internal state vectors only have non-zero values in the first $|\mathcal{K}|$ positions, $\boldsymbol{u} \in \text{span}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{|\mathcal{K}|}\}$. In this case, $\boldsymbol{u} \notin \mathcal{N}\left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}\right)$, implying that $\mathbf{A}_{\mathcal{K}}\boldsymbol{u} = -\boldsymbol{\Phi}^t\boldsymbol{\Phi}\mathbf{S}\boldsymbol{u} \neq 0$. Consider next the case when all non-trivial internal state vectors only have non-zero values in the last $(K - |\mathcal{K}|)$ positions, $\boldsymbol{u} \in \text{span}\{\boldsymbol{e}_{|\mathcal{K}|+1}, \ldots, \boldsymbol{e}_K\}$. In this case, $\boldsymbol{u} \in \mathcal{N}\left(\boldsymbol{\Phi}^t\boldsymbol{\Phi}\mathbf{S}_{\mathcal{K}}\right)$, meaning that $\mathbf{A}_{\mathcal{K}}\boldsymbol{u} = -\boldsymbol{u} \neq 0$. Taking these two cases together, we see that $\mathbf{A}_{\mathcal{K}}\boldsymbol{u} \neq 0, \forall \boldsymbol{u} \in \mathbb{R}^K$, implying that $\mathbf{A}_{\mathcal{K}}$ only has negative eigenvalues so that the system in question has a single equilibrium point.

Given a particular set of active nodes $\mathcal{K}$, we therefore have a single equilibrium point $\boldsymbol{u}^*$ defined by the system matrix $\mathbf{A}_{\mathcal{K}}$. All other points within a neighborhood of this equilibrium point correspond to the same set of active nodes (and therefore the same system matrix). Therefore, since each system matrix has a single equilibrium,

no other equilibrium points exist with coordinates within a neighborhood of $\boldsymbol{u}^*$,

$$|u_k^* - u_k| < r \quad \text{for any} \quad k \implies f(\boldsymbol{u}) \neq 0.$$

We know then the number of equilibrium points is finite, and each equilibrium point is isolated because no other equilibrium points can be infinitely close.

Finally we consider the stability of the system in the neighborhood of the equilibrium point $\boldsymbol{u}^*$. Because we know that the linear subsystem is asymptotically stable, we must show that there exists a $\epsilon > 0$ such that for any $\boldsymbol{u}(0) \in \mathcal{B}_\epsilon(\boldsymbol{u}^*)$, the set of active nodes $\mathcal{K}$ never changes so $\mathbf{A}_\mathcal{K}$ stays fixed. We must therefore ensure that we can specify a $\epsilon > 0$ such that for a fixed $\mathbf{A}_\mathcal{K}$ the internal states never change state, $|u_k(t)| > \lambda, \forall k \in \mathcal{K}$ and $|u_k(t)| < \lambda, \forall k \notin \mathcal{K}$. The tools of linear systems theory give this evolution [66]:

$$\begin{aligned}
\boldsymbol{u}(t) &= e^{\mathbf{A}_\mathcal{K} t}\boldsymbol{u}(0) + \int_0^t e^{(t-\tau)\mathbf{A}_\mathcal{K}}\boldsymbol{\Phi}^t \boldsymbol{x} d\tau \\
&= e^{\mathbf{A}_\mathcal{K} t}\boldsymbol{u}(0) + e^{\mathbf{A}_\mathcal{K} t}\left(\int_0^t e^{-\mathbf{A}_\mathcal{K}\tau} d\tau\right)\boldsymbol{\Phi}^t \boldsymbol{x} \\
&= e^{\mathbf{A}_\mathcal{K} t}\boldsymbol{u}(0) + e^{\mathbf{A}_\mathcal{K} t}\left(-\mathbf{A}^{-1}e^{-\mathbf{A}_\mathcal{K} t} + \mathbf{A}^{-1}\right)\boldsymbol{\Phi}^t \boldsymbol{x} \\
&= e^{\mathbf{A}_\mathcal{K} t}\boldsymbol{u}(0) + -\mathbf{A}^{-1}\boldsymbol{\Phi}^t \boldsymbol{x} + e^{\mathbf{A}_\mathcal{K} t}\mathbf{A}^{-1}\boldsymbol{\Phi}^t \boldsymbol{x} \\
&= e^{\mathbf{A}_\mathcal{K} t}\left(\boldsymbol{u}(0) - \boldsymbol{u}^*\right) + \boldsymbol{u}^*,
\end{aligned}$$

where $\lim \boldsymbol{u}(t) = -\mathbf{A}^{-1}\boldsymbol{\Phi}^t \boldsymbol{x} = \boldsymbol{u}^*$ for a linear system. From this, we bound the energy of the difference signal

$$\|\boldsymbol{u}(t) - \boldsymbol{u}^*\| = \left\|e^{\mathbf{A}_\mathcal{K} t}\left(\boldsymbol{u}(0) - \boldsymbol{u}^*\right)\right\| \leq e^{\mu_{\max} t}\|\boldsymbol{u}(0) - \boldsymbol{u}^*\| \leq \|\boldsymbol{u}(0) - \boldsymbol{u}^*\| \leq \epsilon,$$

where $\mu_{\max}$ is the largest magnitude eigenvector of $\mathbf{A}_\mathcal{K}$. This energy bound also serves as a crude bound on the individual elements of the internal state vector

$$|u_k(0) - u_k^*| \leq \|\boldsymbol{u}(t) - \boldsymbol{u}^*\| \leq \epsilon.$$

We conclude that if $\epsilon < r$, the system will not change state and behaves as a fixed, asymptotically stable linear system. Therefore, the system is locally asymptotically stable around each equilibrium point, $\boldsymbol{u}(0) \in \mathcal{B}_r(\boldsymbol{u}^*)$.

E.2 Input-output Stability

Consider the time derivative of the energy computed through the chain rule (using appropriate vector derivatives),

$$\frac{d}{dt}E(t) = \frac{dE}{d\boldsymbol{a}}\frac{d\boldsymbol{a}}{d\boldsymbol{u}}\dot{\boldsymbol{u}} \propto -\frac{dE}{d\boldsymbol{a}}\frac{d\boldsymbol{a}}{d\boldsymbol{u}}\frac{dE}{d\boldsymbol{a}},$$

where the last equality follows from the definition of the LCA dynamics as proportional (with a positive constant) to the negative gradient of E with respect to the active coefficients, $\dot{\boldsymbol{u}} \propto -\frac{dE}{d\boldsymbol{a}}$. Therefore, as long as $T_\lambda(\cdot)$ is non-decreasing, $\frac{d\boldsymbol{a}}{d\boldsymbol{u}} \geq 0$, implying that the energy function will be non-increasing with time $\frac{d}{dt}E(t) \leq 0$.

The average dwell time theorem stated in Section 5.4.4 guarantees that the LCA system will remain stable as long as each subsystem is asymptotically stable and τ_D is not too small. Appendix E.1 shows that the system matrix $\mathbf{A}_\mathcal{K}$ has only strictly negative eigenvalues. The modified system matrix $\widetilde{\mathbf{A}} = \mathbf{A}_\mathcal{K} + \omega_0 I$ has a minimum magnitude eigenvalue of $\widetilde{\mu}_{\min} = \mu_{\min} + \omega_0$. Clearly there exists a $\omega_0 > 0$ such that $\widetilde{\mu}_{\min} > 0$ if and only if $\mu_{\min} > 0$. In other words, there exists $\omega_0 > 0$ so that $\widetilde{\mathbf{A}}$ is asymptotically stable (thus satisfying the average dwell time theorem) if the stability criteria are met for every subsystem of the switched system.

Appendix F
Steady-state Sparsity of LCA Systems

The LCA at steady-state looks like a fixed linear system. If we know *a priori* the set of active nodes corresponding to the steady-state response $\mathcal{K}$ then the steady-state internal state variables are given by

$$\widetilde{\boldsymbol{u}} = \lim_{t\to\infty} \boldsymbol{u}(t) = -\frac{1}{\tau}\mathbf{A}_{\mathcal{K}}^{-1}\boldsymbol{\Phi}^t\boldsymbol{x},$$

where $\mathbf{A}_{\mathcal{K}}$ is defined as in Appendix E. While we cannot determine the set of active nodes in the limit, we can distinguish sets of nodes that *cannot* be active. When calculating the steady-state values $\widetilde{\boldsymbol{u}}$ assuming a fixed $\mathcal{K}$, if a node not in $\mathcal{K}$ is above threshold in $\widetilde{\boldsymbol{u}}$ (or a node in $\mathcal{K}$ is below threshold), the system matrix would have changed. In this case we call $\mathcal{K}$ *inconsistent*. It is important to note a subtle point: finding an inconsistency does not indicate the correct steady-state active set, but only indicates that it cannot be $\mathcal{K}$.

Given a set of candidate active nodes $\mathcal{K}$, we assume (without losing generality) the active nodes are indexed consecutively from the beginning, $\mathcal{K} = 1, \ldots, |\mathcal{K}|$. We employ the usual canonical basis elements $\boldsymbol{e}_k \in \mathbb{R}^K$ that contain a single non-zero entry in the k^{th} position (e.g., $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^t$). We will also employ what we call the *Grammian basis elements* $\boldsymbol{v}_k \in \mathbb{R}^K$ that contain the inner products of one dictionary element with all the others, $\boldsymbol{v}_k = [\langle\boldsymbol{\phi}_1, \boldsymbol{\phi}_k\rangle, \langle\boldsymbol{\phi}_2, \boldsymbol{\phi}_k\rangle, \ldots, \langle\boldsymbol{\phi}_K, \boldsymbol{\phi}_k\rangle]^t = [\mathbf{G}_{1,k}, \mathbf{G}_{2,k}, \ldots, \mathbf{G}_{K,k}]^t$. The system matrix can be expressed entirely in terms of these basis elements,

$$\mathbf{A}_{\mathcal{K}} = \frac{1}{\tau}\left[\left(I - \boldsymbol{\Phi}^t\boldsymbol{\Phi}\right)\mathbf{S}_{\mathcal{K}} - I\right] = -\frac{1}{\tau}\left[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{|\mathcal{K}|}, \boldsymbol{e}_{|\mathcal{K}|+1}, \ldots, \boldsymbol{e}_K\right],$$

where $\mathbf{S}_{\mathcal{K}}$ is the corresponding selection matrix (defined in Appendix E.1). The inverse of this system matrix has several important properties. First, for inactive nodes $k \notin \mathcal{K}$, the corresponding canonical basis vector is an eigenvector of the inverse system matrix $\mathbf{A}^{-1}\boldsymbol{e}_k = -\tau\boldsymbol{e}_k$. Similarly, for active nodes $k \in \mathcal{K}$, the inverse system matrix transforms the corresponding Grammian basis vector into the canonical basis vector, $\mathbf{A}^{-1}\boldsymbol{v}_k = -\tau\boldsymbol{e}_k$. We also note that the set $\left(\{\boldsymbol{v}_k\}_{k=1}^{|\mathcal{K}|} \bigcup \{\boldsymbol{e}_k\}_{k=|\mathcal{K}|+1}^{K}\right)$ is a basis for the space $\mathbb{R}^K$.

For now, let the input signal be proportional to a single dictionary element, $\boldsymbol{x} = \nu\boldsymbol{\phi}_n$, meaning that $\boldsymbol{\Phi}^t\boldsymbol{x} = \nu\boldsymbol{v}_n$. We will assume that the scaling coefficient is greater than the chosen threshold, $\nu > \lambda$, so the signal strength is considered

significant. There exists a unique set of coefficients $\{\xi_k\}$ such that

$$\nu \boldsymbol{v}_n = \xi_1 \boldsymbol{v}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{v}_{|\mathcal{K}|} + \xi_{|\mathcal{K}|+1} \boldsymbol{e}_{|\mathcal{K}|+1} + \cdots + \xi_K \boldsymbol{e}_K.$$

Looking at each element of this expression in turn is illuminating:

$$\begin{aligned}
\nu \left[\boldsymbol{v}_n\right]_1 &= \langle \boldsymbol{\phi}_1, \nu \boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_1, \left(\xi_1 \boldsymbol{\phi}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{\phi}_{|\mathcal{K}|}\right) \rangle \\
&&&\vdots \\
\nu \left[\boldsymbol{v}_n\right]_{|\mathcal{K}|} &= \langle \boldsymbol{\phi}_{|\mathcal{K}|}, \nu \boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_{|\mathcal{K}|}, \left(\xi_1 \boldsymbol{\phi}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{\phi}_{|\mathcal{K}|}\right) \rangle \\
\nu \left[\boldsymbol{v}_n\right]_{|\mathcal{K}|+1} &= \langle \boldsymbol{\phi}_{|\mathcal{K}|+1}, \nu \boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_{|\mathcal{K}|+1}, \left(\xi_1 \boldsymbol{\phi}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{\phi}_{|\mathcal{K}|} + \xi_{|\mathcal{K}|+1} \boldsymbol{\phi}_{|\mathcal{K}|+1}\right) \rangle \\
&&&\vdots \\
\nu \left[\boldsymbol{v}_n\right]_K &= \langle \boldsymbol{\phi}_K, \nu \boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_K, \left(\xi_1 \boldsymbol{\phi}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{\phi}_{|\mathcal{K}|} + \xi_K \boldsymbol{\phi}_K\right) \rangle.
\end{aligned}$$

The coefficients $\xi_1, \ldots, \xi_{|\mathcal{K}|}$ correspond to the best approximation of $\boldsymbol{x}$ in the subspace spanned by $\{\boldsymbol{\phi}_k\}_{k=1}^{|\mathcal{K}|}$.

Consider first the case when $n \in \mathcal{K}$. The coefficients are optimal: $\xi_n = 1$ and $\xi_k = 0$ for all $k \neq n$. Assuming the fixed system matrix $\mathbf{A}_{\mathcal{K}}$, the steady-state internal state variables are given by

$$\widetilde{\boldsymbol{u}} = -\frac{1}{\tau} \mathbf{A}_{\mathcal{K}}^{-1} \boldsymbol{\Phi}^t \boldsymbol{x} = -\frac{1}{\tau} \mathbf{A}_{\mathcal{K}}^{-1} \nu \boldsymbol{v}_n = \nu \boldsymbol{e}_n.$$

If the LCA selects the optimal set of nodes, the coefficient values are optimal. Now consider the case when the vector is not part of the active set, $n \notin \mathcal{K}$. The coefficients $\{\xi_k\}$ correspond to the steady-state values:

$$-\frac{1}{\tau} \mathbf{A}_{\mathcal{K}}^{-1} \boldsymbol{\Phi}^t \boldsymbol{x} = \xi_1 \boldsymbol{e}_1 + \cdots + \xi_{|\mathcal{K}|} \boldsymbol{e}_{|\mathcal{K}|} + \xi_{|\mathcal{K}|+1} \boldsymbol{e}_{|\mathcal{K}|+1} + \cdots + \xi_K \boldsymbol{e}_K.$$

Looking at the entries of $\nu \boldsymbol{v}_n$, each index not in the active set, $k \notin \mathcal{K}$, has the coefficient

$$\xi_k = \nu \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_n \rangle - \left(\xi_1 \langle \boldsymbol{\phi}_1, \boldsymbol{\phi}_{|\mathcal{K}|+1} \rangle + \cdots + \xi_{|\mathcal{K}|} \langle \boldsymbol{\phi}_{|\mathcal{K}|}, \boldsymbol{\phi}_{|\mathcal{K}|+1} \rangle\right).$$

The set $\mathcal{K}$ is consistent only if $\xi_k > \lambda$ for all $k \in \mathcal{K}$, and $\xi_k > \lambda$ for all $k \notin \mathcal{K}$.

These results lead us to several observations that help qualify the sparsity of the LCA solutions:

1. When $n \in \mathcal{K}$, $\widetilde{u}_k = 0 < \lambda$ for all $k \neq n$ means that if the LCA finds the optimal node, it will not include any extraneous nodes.

2. When $n \in \mathcal{K}$, $\widetilde{u}_n = \nu$ means that if the optimal node is correctly selected by the

LCA in the steady state, the system will find the optimal coefficient for that node.

3. When $n \notin \mathcal{K}$, $\xi_k > \lambda$ for all $k \in \mathcal{K}$ and $\xi_k < \lambda$ for all $k \notin \mathcal{K}$ means that the input signal can be represented by dictionary elements $\{\boldsymbol{\phi}_k\}_{k=1}^{|\mathcal{K}|}$ so the residual projection onto any other vector is less than λ. Any set of active nodes that cannot represent the input signal to this accuracy is inconsistent.

To minimize notation, we have only discussed one-sparse input signals. However, the analysis performed here is entirely linear and the same principles apply to input signals containing more than one dictionary component. In particular, a set of active nodes is inconsistent if: it cannot represent every component of the input signal so that the residual projection onto every other dictionary element is less than λ; or it contains every component of the input signal in addition to other extraneous components. Also, if the active set recovers the correct indices, the LCA steady-state coefficients will find the optimal coefficients.

Bibliography

[1] I. Akyildiz and I. Kasimoglu. Wireless sensor and actor networks: research challenge. *Ad Hoc Networks*, 2(4):351–267, October 2004.

[2] I. Akyildiz and E. Stuntebeck. Wireless underground sensors networks: Research challenges. *Ad Hoc Networks*, 4:669–686, 2006.

[3] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor. Video compression using matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1):123–143, February 1999.

[4] A. Bacciotti and L. Rosier. *Liapunov functions and stability in control theory.* Springer, New York, 2001.

[5] R. Balan, P. Casazza, C. Heil, and Z. Landau. Density, overcompleteness, and localization of frames, I. Theory. Preprint, 2005.

[6] J. Benedetto, A. Powell, and O. Yilmaz. Sigma-delta quantization and finite frames. *IEEE Transactions on Information Theory*, 52:1990–2005, 2006.

[7] T. Berger. *Rate Distortion Theory.* Prentice Hall, Englewood Cliffs, NJ, 1971.

[8] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the third international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, 2004. ACM Press.

[9] D. Blatt and A. Hero. Distributed maximum likelihood estimation in sensor networks. In *International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[10] H. Bolcskei, F. Hlawatsch, and H. Feichtinger. Frame-theoretic analysis of oversampled filter banks. *IEEE Transactions on Signal Processing*, 46(12):3256–3268, 1998.

[11] P. Boufounos and A. V. Oppenheim. Quantization noise shaping on arbitrary frame expansions. *EURASIP Journal on Applied Signal Processing*, 2006, 2006.

[12] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Communications on Pure and Applied Mathematics*, 57(2):219–266, 2004.

[13] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[14] E. J. Candès, M. Rudelson, R. Vershynin, and T. Tao. Error correction via linear programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 295–308, 2005.

[15] P. Casazza. The art of frame theory. *Taiwanese Journal of Mathematics*, 4(2):129–202, 2000.

[16] P. Casazza and G. Kutyniok. Frames of subspaces. In *Wavelets, Frames and Operator Theory*, volume 345, pages 87–113. American Mathematical Society, 2004.

[17] P. Casazza, G. Kutyniok, and S. Li. Fusion frames and distributed processing. *Preprint*, 2006.

[18] V. Chandrasekaran, M. Wakin, D. Baron, and R. Baraniuk. Representation and compression of multi-dimensional piecewise functions using surflets. *Preprint*, 2006.

[19] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 43(1):129–159, 2001.

[20] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhauser, Boston, MA, 2002.

[21] M. Coates. Evaluating causal relationships in wireless sensor/actuator networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, March 2005.

[22] M. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):815–825, September–October 1983.

[23] R. Coifman and D. Donoho. *Wavelets and Statistics*, chapter Translation-Invariant Denoising. Springer-Verlag, New York, 1995.

[24] R. Coifman and M. Wickerhauser. *Wavelets and their Applications*, chapter Wavelet analysis and signal processing, pages 153–178. Jones and Bartlett, Boston, MA, 1992.

[25] Committee on Networked Systems of Embedded Computers, National Research Council. *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, Washington, DC, 2001.

[26] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume I. John Wiley & Sons, New York, 1937.

[27] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, NY, 1991.

[28] D. Cox and P. Lewis. *The Statistical Analysis of Series of Events*. John Wiley & Sons, Inc., New York, NY, 1966.

[29] S. Cui, A. Goldsmith, and A. Bahai. Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1089–1098, August 2004.

[30] S. Cui, A. Goldsmith, and A. Bahai. Joint modulation and multiple access optimization under energy constraints. In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 1, pages 151–155, 2004.

[31] S. Cui, A. Goldsmith, and A. Bahai. Energy-constrained modulation optimization. *IEEE Transactions on Wireless Communications*, 4(5):2349–2360, September 2005.

[32] S. Cui, J. Xiao, A. Goldsmith, Z. Luo, and V. Poor. Energy-efficient joint estimation in sensor networks: Analog vs. digital. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, March 2005.

[33] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[34] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[35] I. Daubechies and R. DeVore. Reconstructing a bandlimited function from very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. *Annals of Mathematics*, 158(2):679–710, 2003.

[36] I. Daubechies, A. Grossman, and Y. Meyer. Painless nonorthogonal expansions. *Journal of Mathematical Physics*, 27:1271–1283, November 1986.

[37] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions with matching pursuit. *Optical Engineering*, 33(7), July 1994.

[38] P. Dayan and A. Abbott. *Theoretical Neuroscience*. MIT Press, Cambridge, MA, 2001.

[39] R. De Valois and K. De Valois. *Spatial Vision*. Oxford University Press, New York, 1988.

[40] R. Decarlo, M. Cranicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7):1069–1082, 2000.

[41] B. Delgutte, B. Hammond, and P. Cariani. *Psychophysical and Physiological Advances in Hearing*, chapter Neural coding of the temporal envelope of speech: Relation to modulation transfer functions, pages 595–603. Whurr Publishers, Ltd., 1998.

[42] V. Delouille, R. Neelamani, and R. Baraniuk. The embedded triangles algorithm for distributed estimation in sensor networks. In *IEEE Workshop on Statistical Signal Processing*, St. Louis, MO, 2003.

[43] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38, 1977.

[44] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2):719–746, March 1992.

[45] R. DeVore and V. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5:173–187, 1996.

[46] M. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, December 2005.

[47] D. Donoho. Denoising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[48] D. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics*, 27:859–897, 1999.

[49] D. Donoho. Neighborly polytopes and sparse solutions of underdetermined linear equations. Preprint, 2005.

[50] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[51] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197–2202, March 2003.

[52] D. Donoho, Y. Tsaig, I. Drori, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. 2006.

[53] P. Dragotti, V. Velisavljevic, M. Vetterli, and B. Beferull-Lozano. Discrete directional wavelet bases and frames for image compression and denoising. In *Proceedings of the SPIE Conference on Wavelet Applications in Signal and Image Processing*, August 2003.

[54] R. Duffin, E. Peterson, and C. Zener. *Geometric Programming — Theory and Application*. Wiley, New York, NY, 1967.

[55] R. Duffin and A. Schaeffer. A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society*, 72(2):341–366, March 1952.

[56] M. Elad. Why simple shrinkage is still relevant for redundant representations? *IEEE Transactions on Information Theory*, 52(12):5559–5569, December 2006.

[57] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2033–2036, Salt Lake City, UT, May 2001.

[58] B. Farber and K. Zeger. Quantization of multiple sources using nonnegative integer bit allocation. *IEEE Transactions on Information Theory*, 52(11):4945–4964, November 2006.

[59] H. Feichtinger, A. Türk, and T. Strohmer. Hierarchical parallel matching pursuit. In *Proceedings of SPIE*, volume 2302, pages 222–232, San Diego, CA, July 1994.

[60] J. Feldman and D. Ballard. Connectionist models and their properties. *Cognitive Science: A Multidisciplinary Journal*, 6(3):205–254, 1982.

[61] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794, New York, NY, USA, 2006. ACM Press.

[62] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12:906–916, 2003.

[63] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Applications to compressed sensing and other inverse problems. *Preprint*, 2007.

[64] S. Fischer, G. Cristóbal, and R. Redondo. Sparse overcomplete Gabor wavelet representation based on local competitions. *IEEE Transactions on Image Processing*, 15(2):265–272, 2004.

[65] A. Fletcher, S. Rangan, V. Goyal, and K. Ramchandran. Denoising by sparse approximation: Error bounds based on rate-distortion theory. *EURASIP Journal on Applied Signal Processing*, 2006, 2006.

[66] G. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, Reading, MA, 1986.

[67] P. Frossard, P. Vandergheynst, R. Figueras i Ventura, and M. Kunt. A posteriori quantization of progressive Matching Pursuit streams. *IEEE Transactions on Signal Processing*, 52(2):525–535, February 2004.

[68] J. Gao, C. Harris, and S. Gunn. On a class of support vector kernels based on frames in function Hilbert spaces. *Neural Computation*, 13:1975–1994, 2001.

[69] B. Gerkey and M. Mataric. A market-based formulation of sensor-actuator network coordination. In *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, 2002.

[70] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, Norwell, MA, 1991.

[71] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, July 1998.

[72] R. Glantz, H. Nudelman, and B. Waldrop. Linear integration of convergent visual inputs in an oculomotor reflex pathway. *Journal of Neurophysiology*, 52(6):1213–1225, December 1984.

[73] V. Goyal, J. Kovačević, and J. Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10:203–233, 2001.

[74] V. Goyal, M. Vetterli, and N. Thao. Quantized overcomplete expansions in $\mathbb{R}^N$: Analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, January 1998.

[75] R. Heath and A. Paulraj. Linear dispersion codes for MIMO systems based on frame theory. *IEEE Transactions on Signal Processing*, 50(10):2429–2441, October 2002.

[76] C. Heil and D. D. Walnut. Continuous and discrete wavelet transforms. *SIAM Review*, 31:628–666, 1989.

[77] K. Herrity, A. Gilbert, and J. Tropp. Sparse approximation via iterative thresholding. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[78] J. Hespanha and A. Morse. Stability of switched systems with average dwell time. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

[79] R. Holmes and V. Paulsen. Optimal frames for erasures. *Linear algebra and its applications*, 377:31–51, 2004.

[80] J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81(10):3088–3092, May 1984.

[81] D. Hubel. *Eye, Brain and Vision*. Scientific American Library, New York, NY, 1988.

[82] D. Hubel and T. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.

[83] A. Ihler, R. Fisher III, J.W.and Moses, and A. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):809–819, April 2005.

[84] D. Johnson. Point process models of single-neuron discharges. *Journal of Computational Neuroscience*, 3(4):275–299, December 1996.

[85] D. Johnson and H. Rodriguez-Diaz. Optimizing physical layer data transmission for minimal signal distortion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[86] J. Jones and L. Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987.

[87] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neural Science*. Appleton & Lange, Norwalk, CT, third edition, 1991.

[88] H. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle Tiver, NJ, third edition, 2002.

[89] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10:234–253, 2001.

[90] N. Kingsbury and T. Reeves. Redundant representation with complex wavelets: How to achieve sparsity. In *Proceedings of the International Conference on Image Processing (ICIP)*, 2003.

[91] M. Kirk, B. Waldrop, and R. Glantz. The crayfish sustaining fibers, I. Morphological representation of visual receptive fields in the second optic neuropil. *Journal of Comparative Physiology*, 146:175–179, 1982.

[92] M. Kirk, B. Waldrop, and R. Glantz. The crayfish sustaining fibers, II. Responses to illumination, membrane properties and adaptation. *Journal of Comparative Physiology*, 150:419–425, 1983.

[93] J. Kovačević and A. Chebira. Life beyond bases: The advent of frames. *IEEE Signal Processing Magazine*, 2007. To appear.

[94] A. Laine and J. Fan. Frame representations for texture segmentation. *IEEE Transactions on Image Processing*, 5(5):771–780, May 1996.

[95] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, November 2003.

[96] M. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5:356–363, 2002.

[97] J. Li, A. Michel, and W. Porod. Qualitative analysis and synthesis of a class of neural networks. *IEEE Transactions on Circuits and Systems*, 35(8):976–986, August 1988.

[98] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao. State-centric programming for sensor and actuator network systems. *IEEE Pervasive Computing*, 2(4):50–62, 2003.

[99] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, New York, NY, second edition, 1999.

[100] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[101] T. Marshall, J. Holmes, and C. Rose. *Soil Physics*. Cambridge University Press, New York, 1979.

[102] R. McLaren and K. Cameron. *Soil Science: Sustainable production and environmental protection*. Oxford University Press, second edition, 1996.

[103] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, April 1995.

[104] R. Neff and A. Zakhor. Very low bit-rate video coding based on matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):158–171, February 1997.

[105] D. Neil. Compensatory eye movements. In D. Sandeman and H. Atwood, editors, *The Biology of Crustacea, Neural Integration and Behavior*, pages 133–163. Academic Press, New York, 1982.

[106] J. Nicholls, A. Martin, B. Wallace, and P. Fuchs. *From Neuron to Brain*. Sinauer Associates, Inc., Sunderland, MA, fourth edition, 2001.

[107] J. Norris. *Markov Chains*. Cambridge University Press, New York, 1997.

[108] R. Nowak. Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Transactions on Signal Processing*, 51:2245–2253, August 2003.

[109] R. Nowak, U. Mitra, and R. Willett. Estimating inhomogeneous fields using wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):999–1006, August 2004.

[110] H. Ochiai, P. Mitran, H. Poor, and V. Tarokh. Collaborative beamforming for distributed wireless ad hoc sensor networks. *IEEE Transactions on Signal Processing*, 53(11):4110–4124, November 2005.

[111] S. Olsen. Wi-Fi mosquito killer coming to a porch near you. Online source *CNET News.com*, November 21, 2005. `http://news.com.com/2100-11395_3-5961535.html`. Accessed April 13, 2007.

[112] B. Olshausen. Principles of image representation in visual cortex. In L. Chalupa and J. Werner, editors, *The Visual Neurosciences*, pages 1603–1615. MIT Press, 2003.

[113] B. Olshausen and D. Field. Emergence of simple cell receptive properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[114] B. Olshausen and D. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

[115] B. Olshausen and D. Field. How close are we to understanding V1? *Neural Computation*, 17:1665–1699, 2005.

[116] M. Padmanabhan and M. Picheny. Towards super-human speech recognition. In *Automatic Speech Recognition, Challenges for the New Millenium*, pages 189–194, Paris, France, September 2000.

[117] N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, and N. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, July 2005.

[118] A. Pece and N. Petkov. Fast atomic decomposition by the inhibition method. In *In Proceedings of the 15th International Conference on Pattern Recognition*, 2000.

[119] L. Perrinet, M. Samuelides, and S. Thorpe. Coding static natural images using spiking event times: Do neurons cooperate? *IEEE Transactions on Neural Networks*, 15(5):1164–1175, September 2004.

[120] L. Perrinet, M. Samuelides, and S. Thorpe. Sparse spike coding in an asynchronous feed-forward multi-layer neural network using matching pursuit. *Neurocomputing*, 57:125–134, 2004.

[121] A. Popper and R. Fay. *The Mammalian Auditory Pathway: Neurophysiology*. Springer-Verlag, New York, 1992.

[122] J. Proakis. *Digital Communications*. McGraw-Hill Science, 2000.

[123] S. Qian and D. Chen. Signal representation using adaptive normalized Gaussian functions. *Signal Processing*, 36(1):1–11, 1994.

[124] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *International Symposium on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, April 2004.

[125] A. Rahmoune, P. Vandergheynst, and P. Frossard. Flexible motion-adaptive video coding with redundant expansions. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):178–190, February 2006.

[126] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9(4):137–140, April 2002.

[127] M. Rehn and T. Sommer. A network that uses few active neurones to code visual input predicts the diverse shapse of cortical receptive fields. *Journal of Computational Neuroscience*, 22(2):135–146, April 2007.

[128] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1979.

[129] S. Roth and M. Black. Fields of experts: a framework for learning image priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, June 2005.

[130] C. Rozell, I. Goodman, and D. Johnson. Feature-based information processing with selective attention. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[131] C. Rozell and D. Johnson. Analysis of noise reduction in redundant expansions under distributed processing requirements. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, March 2005.

[132] C. Rozell and D. Johnson. Examining methods for estimating mutual information in spiking neural systems. *Neurocomputing*, 65–66C:429–434, June 2005. Also appears in *Proceedings of the Computational Neuroscience Meeting (CNS)*, Baltimore, MD, July 2004.

[133] C. Rozell and D. Johnson. Analyzing the robustness of redundant population codes in sensory and feature extraction systems. *Neurocomputing*, 69(10–12):1215–1218, June 2006. Also appears in *Proceedings of the Computational Neuroscience Meeting (CNS)*, Madison, WI, July 2005.

[134] C. Rozell and D. Johnson. Evaluating local contributions to global performance in wireless sensor and actuator networks. *Lecture Notes in Computer Science*, 4026:1–16, 2006. *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, San Francisco, CA, June 2006.

[135] C. Rozell and D. Johnson. Power scheduling for wireless sensor and actuator networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, April 2007.

[136] C. Rozell, D. Johnson, R. Baraniuk, and B. Olshausen. Neurally plausible sparse coding via thresholding and local competition. *Neural Computation*, 2007. Submitted.

[137] C. Rozell, D. Johnson, and R. Glantz. Measuring information transfer in crayfish sustaining fiber spike generators. *Biological Cybernetics*, 90(2):89–97, February 2004.

[138] M. Rudelson and R. Vershynin. Geometric approach to error correcting codes and reconstruction of signals. *International Mathematics Research Notices*, 64:4019–4041, 2005.

[139] I. Selesnick. The design of approximate hilbert transform pairs of wavelet bases. *IEEE Transactions on Signal Processing*, 50(5):1144–1152, May 2002.

[140] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[141] E. Simoncelli and W. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE Second International Conference on Image Processing*, 1995.

[142] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

[143] D. Snyder and M. Miller. *Random Point Process in Time and Space*. Springer-Verlag, New York, NY, second edition, 1991.

[144] T. Strohmer and R. Heath Jr. Grassmannian frames with applications to coding and communcations. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003.

[145] H. Taylor, S. Banks, and J. McCoy. Deconvolution with the ℓ_1 norm. *Geophysics*, 44(1):39–52, 1979.

[146] M. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[147] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[148] J. Tropp. Random subdictionaries of general dictionaries. Preprint, 2006.

[149] J. Tropp, I. Dhillon, R. Heath Jr., and T. Strohmer. Designing structured tight frames via an alternating projection method. *IEEE Transactions on Information Theory*, 51(1):188–209, January 2005.

[150] W. Vinje and J. Gallant. Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. *Journal of Neuroscience*, 22:2904–2915, 2002.

[151] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk. Wavelet-domain approximation and compression of piecewise smooth images. *IEEE Transactions on Image Processing*, 15(5):1071–1087, 2006.

[152] S. Waldron. Generalized Welch bound equality sequences are tight frames. *IEEE Transactions on Information Theory*, 49(9):2307–2309, September 2003.

[153] C. Wiersma and T. Yamaguchi. The neuronal components of the optic nerve of the crayfish as studied by single unit analysis. *Journal of Comparative Neurology*, 138:337–358, 1966.

[154] R. Willett and R. Nowak. Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Transactions on Medical Imaging*, 22(3):332–350, March 2003.

[155] D. Wipf and B. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, August 2004.

[156] S. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publications, 1997.

[157] J. Xiao, S. Cui, Z. Luo, and A. Goldsmith. Power scheduling of universal decentralized estimation in sensor networks. *IEEE Transactions on Signal Processing*, 54(2):413–422, February 2006.

[158] H. Yang and T. Dillon. Exponential stability and oscillation of Hopfield graded response neural network. *IEEE Transactions on Neural Networks*, 5(5):719–729, September 1994.

[159] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2004.