UNIVERSITY OF CALIFORNIA
Santa Barbara

Scalable Front End Designs for Communication and Learning

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Aseem Wadhwa

Committee in Charge:

Professor Upamanyu Madhow, Chair

Professor João P. Hespanha

Professor Kenneth Rose

Professor Alberto G. Busetto

December 2014

The Dissertation of
Aseem Wadhwa is approved:

Professor João P. Hespanha

Professor Kenneth Rose

Professor Alberto G. Busetto

Professor Upamanyu Madhow, Committee Chairperson

December 2014

Scalable Front End Designs for Communication and Learning

by

Aseem Wadhwa

Acknowledgements

I am extremely grateful to Professor Upamanyu Madhow for his expert guidance and steady support throughout the five years of my graduate research. His feedback and incisive insights have been invaluable and I feel extremely fortunate for having him as my advisor. During my PhD I have had the good fortune of collaborating with Prof Hespanha, Prof Shanbhag, Prof Ashby and Prof Eckstein. I got an opportunity to learn a lot of new things from each one of them. It was a pleasure working and interacting with their students, Jason, Yingyan, Ben, Erick and Emre.

I thank Prof Rose, Prof Hespanha and Prof Busetto for serving on my thesis committee.

It has been a joy spending these years with my colleagues at the WCSL lab. I would like to thank lab "seniors" Sriram, Sumit, Sandeep, Hong and Eric for being great guides and helping me ease into the life of a graduate student. I greatly cherish the time spent with them and my other colleagues who joined the lab along with me or after me, Dinesh, Andrew, Francois, Babak, Maryam, Hossein, Farukh and Zhinus.

Finally, I would like to thank my parents, my brother-in-law Nikhil and my sisters, Astha and Pooja, for their unconditional love and support.

Curriculum Vitæ
Aseem Wadhwa

Education

December 2014	Doctor of Philosophy, Electrical and Computer Engineering, University of California, Santa Barbara
June 2011	Master of Science, Electrical and Computer Engineering, University of California, Santa Barbara
August 2009	Bachelor of Technology, Electrical Engineering, Indian Institute of Technology Delhi

Publications

- E. Akbas, A. Wadhwa, M. Eckstein and U. Madhow. "A Framework for Machine Vision based on Neuro-Mimetic Front End Processing and Clustering". *Proc. of 52nd Allerton Conference on Communication Control and Computing*, October 2014
- A. Wadhwa, U. Madhow and N. Shanbhag. "Space-time Slicer Architectures for Analog-to-Information Conversion in Channel Equalizers". *Proc. of IEEE International Conference on Communications (ICC'14)*, Sydney, Australia, June 2014
- A. Wadhwa and U. Madhow. "Blind phase/frequency synchronization with low-precision ADC: a Bayesian approach". *Proc. of 51st Allerton Conference on Communication Control and Computing*, Oct 2013
- A. Wadhwa, U. Madhow, J. Hespanha and B.Sadler. "Following an RF trail to its source". *Proc. of 49th Allerton Conference on Communication Control and Computing*, Sept 2011

Abstract

Scalable Front End Designs for Communication and Learning

Aseem Wadhwa

In this work we provide three examples of estimation/detection problems, for which customizing the Front End to the specific application makes the system more efficient and scalable. The three problems we consider are all classical, but face new scalability challenges. This introduces additional constraints, accounting for which results in front end designs that are very distinct from the conventional approaches. The first two case studies pertain to the canonical problems of synchronization and equalization for communication links. As the system bandwidths scale, challenges arise due to the limiting resolution of analog-to-digital converters (ADCs). We discuss system designs that react to this bottleneck by drastically relaxing the precision requirements of the front end and correspondingly modifying the back end algorithms using Bayesian principles. The third problem we discuss belongs to the field of computer vision. Inspired by the research in neuroscience about the mammalian visual system, we redesign the front end of a machine vision system to be neuro-mimetic, followed by layers of unsupervised learning using simple k-means clustering. This results in a framework that is intuitive, more computationally efficient compared to the approach of supervised

deep networks, and amenable to the increasing availability of large amounts of unlabeled data.

We first consider the problem of blind carrier phase and frequency synchronization in order to obtain insight into the performance limitations imposed by severe quantization constraints. We adopt a mixed signal analog front end that coarsely quantizes the phase and employs a digitally controlled feedback that applies a phase shift prior to the ADC, this acts as a controllable dither signal and aids in the estimation process. We propose a control policy for the feedback and show that combined with blind Bayesian algorithms, it results in excellent performance, close to that of an unquantized system.

Next, we take up the problem of channel equalization with severe limits on the number of slicers available for the ADC. We find that the standard flash ADC architecture can be highly sub-optimal in the presence of such constraints. Hence we explore a "space-time" generalization of the flash architecture by allowing a fixed number of slicers to be dispersed in time (sampling phase) as well as space (i.e., amplitude). We show that optimizing the slicer locations, conditioned on the channel, results in significant gains in the bit error rate (BER) performance.

Finally, we explore alternative ways of learning convolutional nets for machine vision, making it easier to interpret and simpler to implement than currently used purely supervised nets. In particular, we investigate a framework that combines a neuro-mimetic front end (designed in collaboration with the neuroscientists from

the psychology department at UCSB) together with unsupervised feature extraction based on clustering. Supervised classification, using a generic support vector machine (SVM), is applied at the end. We obtain competitive classification results on standard image databases, beating the state of the art for NORB (uniform-normalized) and approaching it for MNIST.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Problems that require some form of estimation/detection are ubiquitous across different fields of study. A common feature of such problems is the presence of an underlying quantity which either takes some mathematical value(s) or belongs to a particular class. Through various processes it gets modified and/or distorted by nature. It is then presented to the "estimator" in a noisy form, whose goal is to recover the true value or class. For example, in communication systems, the underlying quantity is the stream of symbols generated by the transmitter. This gets modified partly by design, when the transmitter converts the discrete sequence into a continuous analog waveform, and partly by the channel, which includes the physical transmission medium and the receiver circuit. The channel can introduce distortions such as inter symbol interference and phase/frequency offsets. The receiver (estimator) then tries to recover the symbols from the noisy continuous valued signal it receives. This process typically requires an implicit or explicit estimation of the channel. Another example is the object recognition

system in computer vision. The underlying quantity to be detected is generally a broad object category, for instance a "car". The process of capturing the image introduces distortions such as rotation, translation, variations in illumination etc. The receiver (recognition system) tries to guess the true class from the raw image of pixel intensities, striving to be invariant to the distortion effects which are irrelevant for detecting the category.

The processing at the receiver can generally be split into two high level blocks: the "front end" and the "back end". The former is responsible for preprocessing the received signal and converting it into a form more convenient for the algorithms running in the back end. For instance, in communication systems, front end performs downconversion to the baseband, followed by the analog to digital (A/D) conversion. The back end then operates on the resulting discrete samples. In vision, front end can be thought of as comprising of the preprocessing operations on raw images such as luminance normalization, extraction of edge information etc. The back end implements the classifier that operates on the features generated by the front end and learns to predict the object category.

Conceptual division of the receiver architecture, as discussed above, is useful. It simplifies design by splitting the overall problem, researchers can focus on smaller blocks in isolation, which are easier to optimize, conditioned on the specifications of the other blocks. For example, a hardware engineer can focus on developing a circuit that delivers precise samples at a fixed rate with high fidelity.

A system engineer can bank on the availability of such a digital signal, safely ignoring the errors due to quantization and clock jitters and concentrate on devising efficient algorithms for estimation. Similarly, in machine learning for example, keeping the lower layer preprocessing and feature extraction fixed, a researcher can direct her energies towards finding the optimal strategy for regularizing the classifier that takes these features as inputs.

Such a design process naturally results in development of generic blocks, which become standard and are reused across several different systems. For example, an A/D front end that minimizes the quantization error and samples at the Nyquist rate, thereby preserving the shape of the continuous waveform, is one such standard block that is used across different systems involving analog and digital interfaces (sensor networks, control, communication systems etc). Similarly, a feature extractor such as SIFT [54] and classifier such as SVM [12] are standard black boxes used in computer vision.

Using the generic blocks usually works in most cases, but issues arise when resources become more constrained as systems scale up. As we find out, in such scenarios, there is great scope of improvement by redesigning the components taking the additional constraints into account. In this dissertation, we revisit the system design for three specific problems. We show that redesigning the front end in a manner that is more adapted to the application at hand leads to better

efficiency and scalability. Qualitatively, following characteristics are desired from an *efficient* front end:

- preserves complete information about the desired quantity, while removing most of the influences irrelevant for estimation (Minimalism).
- is amenable to low cost circuit implementation, which translates to the requirement of the processing being power efficient and computationally efficient (Scalability).

Depending on the application these requirements drive the design process in interesting ways. Of the three problems discussed in this work, two are from the field of communication systems and one from machine vision.

In communication systems, the conventional A/D frontend, as discussed earlier, strives to preserve the shape of the continuous waveform. This helps in maintaining the linearity of the overall system under Gaussian noise and results in a simple back end. However as the system scales up in bandwidth, this approach is no more feasible as the cost of high precision ADCs (analog to digital converters) becomes enormous at high sampling rates. A natural solution is to relax the requirement on bits of precision, and compensate for the added nonlinearity in the back end using sophisticated algorithms in DSP (digital signal processing). This complexity trade off between the front end and the back end is justified due to the Moore's law, which has resulted in much more favorable

scaling of the DSP compared to the ADC technology. In this dissertation, we investigate new architectures for the front end and adapt the corresponding back end algorithms using Bayesian principles to handle the severe non-linearity.

The front end of a machine vision system has the task of extracting features suitable for classification. Compared to the other two problems discussed in this dissertation, this problem is very different as there is a lack of a well defined system and noise model. Several different approaches have been employed for solving the recognition problem, but most solutions use an architecture known as the convolutional neural network (CNN). In recent years, most of the high performing solutions use supervised deep networks, a specific implementation of CNNs. However currently they suffer from a few drawbacks, there is a lack of clarity on exactly how they work and complications in implementation due to the large number of parameters to be tuned and the increased complexity. Our objective is to somehow significantly simplify the system without giving too much away in terms of the performance. This is a difficult objective but we take a few encouraging initial steps towards it in this work. In the absence of well defined models, we look to leverage the next best thing available to us: the mammalian eye. The eye has evolved over thousands of years and the neuroscience literature contains detailed descriptions of the retinal processing. Inspired by this we build a neuro-mimetic front end for preprocessing the raw images. This front end when combined with the neurally plausible idea that our visual system extracts a set

of "universal features", leads to the principle of unsupervised learning using k-means clustering followed by a standard supervised classifier in the final stage. Even though we only present a preliminary study in this work, this framework holds promise for an intuitive implementation that has low complexity and is thus scalable in terms of the size of the dataset. Moreover, other characteristics of this architecture like the requirement of high sparsity in the neural activations (discussed in detail later), makes it a potential candidate for low power hardware implementations.

Sections (1.1),(1.2) and (1.3) introduce the problems considered in this dissertation and summarize our contributions. Detailed discussions of these problems are presented in chapters 2, 3 and 4 respectively.

1.1 Blind Phase/frequency Synchronization

Modern communication transceiver designs leverage Moore's law for low-cost implementation by using DSP to perform sophisticated functionalities such as synchronization, equalization, demodulation and decoding. The central assumption in such designs is that analog signals can be faithfully represented in the digital domain, typically using ADCs with 8-12 bits of precision. However this approach runs into a bottleneck with emerging communication systems employing bandwidths of multiple GHz, such as emerging millimeter wave wireless networks

(e.g., using the 7 GHz of unlicensed spectrum in the 60 GHz band), as well as high speed links in wide bandwidth systems such optical communications and wireline backplane channels. The key reason for this bottleneck is the ADC: as signal bandwidths scale up to multiples of GHz the cost and power consumption of high-resolution ADCs become prohibitive [60].

Since we would like to continue taking advantage of Moore's law despite this bottleneck, it is natural to ask whether DSP-centric architectures with samples quantized at significantly less precision (e.g., 1-4 bits) can be effective. Shannon-theoretic analysis (for idealized channel models) has shown that the loss in channel capacity due to limited ADC precision is relatively small even at moderately high signal-to-noise ratios (SNRs) [76]. This motivates a systematic investigation of DSP algorithms for estimating and compensating for channel non-idealities (e.g., asynchronism, dispersion) using severely quantized inputs.

In particular, we first consider a canonical problem of blind carrier phase/frequency synchronization based on coarse *phase-only quantization* (implementable using digitally controlled linear analog preprocessing of I and Q samples, followed by one-bit ADCs), and develop and evaluate the performance of a Bayesian approach based on joint modeling of the unknown data, frequency and phase, and the known quantization nonlinearity. The case of channel dispersion is taken up in chapter 3. To aid phase/frequency recovery in the face of severe quantization we adopt a

mixed signal architecture that employs a digitally controlled feedback that applies a phase shift prior to the ADC. This is described next.

Receiver architecture: We consider differentially encoded QPSK over an AWGN channel. In order to develop fundamental insight into carrier synchronization, we do not model timing asynchronism or channel dispersion. In the model depicted in Fig. 1.1, the *analog preprocessing front-end* performs downconversion, ideal symbol rate sampling, and applies a digitally controlled *derotation phase* on the complex-valued symbol rate samples before passing it through the *ADC block.* The derotation phase feedback provides a controllable and variable phase offset that acts as a dither signal. Properly designed dither aids in faster estimation and is crucial at high SNRs to ensure diversity in the quantized phase measurements. The ADC block quantizes the phase of the samples into a small number of bins. Phase quantization (which suffices for hard decisions with PSK constellations) has the advantage of not requiring automatic gain control (AGC), since it can be implemented by passing linear combinations of the in-phase and quadrature components through one-bit ADCs (quantization into $2n$ phase bins requires n such linear combinations) [77]. The quantized phase observations are processed in DSP by the *estimation and control block:* this runs algorithms for nonlinear phase and frequency estimation, computes feedback for the analog preprocessor (to aid in estimation and demodulation), and outputs demodulated symbols. Design of this estimation and control block is the subject of this work.

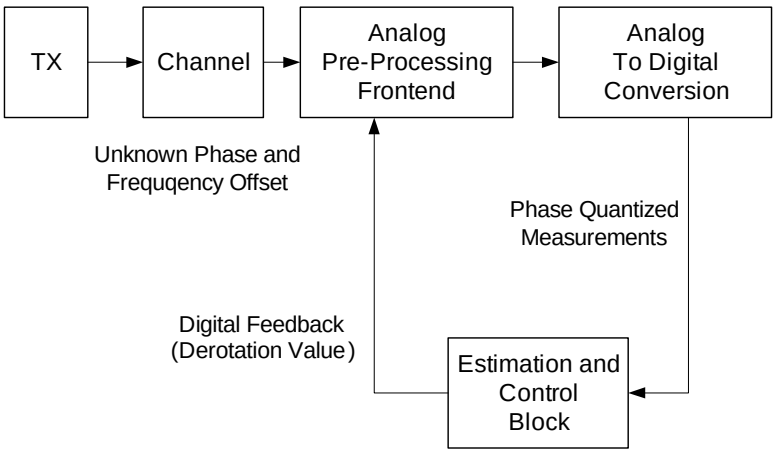


Figure 1.1: Receiver Architecture

Contributions: We break the synchronization problem into two steps (a) rapid blind *acquisition* of initial frequency/phase estimates, (b) continuous *tracking* while performing data demodulation. For solving (a) we develop a Bayesian algorithm for blind phase estimation and propose an information theoretic policy for setting the dither signal. We discuss various properties of this policy in detail and show, via simulations, that it is not far away from the optimal in terms of achieving the minimum mean square error of phase for a given number of symbols. For part (b) we propose an extended Kalman filter (EKF) for frequency/phase tracking. We provide numerical results demonstrating the efficacy of our approach for both steps, and show that the bit error rate with 8-12 phase bins (implementable using linear I/Q processing and 4-6 one bit ADCs) is close to that of a coherent system, and is significantly better than that of standard differential demodulation (which does not require phase/frequency tracking) with unquantized observations.

1.2 Slicer Architectures for Analog-to-Information Conversion in Channel Equalizers

In this section we study the problem of channel equalization under severe restrictions imposed on ADC resolution. In this low bits of precision regime (1-3 bits), it becomes natural to consider alternatives to the general-purpose ADC that are tailored to the communications application. Thus, we are interested in the design of *analog-to-information converters* enabling reliable recovery of the transmitted data, rather than accurate reproduction of the received signal as for a standard ADC. In this work, we explore this approach for communication over static dispersive channels for the simplest possible setting of binary antipodal signaling over a real baseband channel.

Our starting point is the flash ADC, a popular architecture for high sampling rates and relatively low resolutions (2-6 bits); see [82, 14] for some recent high-speed flash ADC designs. An n-bit flash ADC consists of $2^n - 1$ comparators sampling synchronously, with comparator thresholds generally spread uniformly over the input signal voltage range. While fractional sampling is known to be more robust than symbol-spaced sampling for systems in which ADC resolution is not an issue, in the regimes we are interested in, the Nyquist sampling rate is already stressing the state of the art, hence the conventional approach is to sample at the Nyquist rate. A key question we address is whether, for a fixed

number of comparators, we can do better by generalizing beyond uniform spacing and Nyquist sampling. We summarize our contributions below.

Contributions:

(1) We first analyze the standard Nyquist sampled, uniformly spaced design. For a given channel, we derive easily computable lower and upper bounds for the smallest number of comparators to avoid an error floor in the bit error rate (BER). The results give insight into the kind of channels that are worse in terms of requiring a larger number of comparators; for example, mixed-phase channels are worse than minimum/maximum phase channels. We also demonstrate via an example how, for the standard design, the BER can be sensitive to the sampling phase, and that more robust performance can be obtained by spreading the same number of slicers across time. This motivates a more systematic study of space-time architectures.

(2) We establish that there are no fundamental performance limitations imposed by spreading slicers out in space and time, by proving that the ℓ_1 distance between a pair of waveforms is preserved upon quantization by n slicers spread across time and having randomly distributed thresholds, if n is larger than a lower bound. The proof of this general result employs the Chernoff bound and the union bound, analogous to the Johnson-Lindenstrauss (JL) lemma [24]. Its application to our equalization problem guarantees the absence of an error floor if sufficiently many 1-bit measurements are obtained with random thresholds. While this result provides a sound theoretical underpinning for space-time slicer architectures, in

practice, good performance is obtained with fewer slicers with carefully chosen thresholds.

(3) We present an approximate optimization technique for adapting, as a function of the channel, the slicer thresholds for symbol-spaced and fractionally-spaced (at $T_s/2$, where T_s denotes the symbol interval) architectures. For a fixed number of slicers, the performance gains over a standard symbol-spaced uniform ADC are significant. Depending on the choice of channel, sampling phase and number of available slicers, the procedure allocates all slicers to one sampling phase or distributes them among the two phases.

1.3 A Framework for Machine Vision based on Neuro-Mimetic Front End Processing and Clustering

Neuro-inspiration has played a key role in machine learning over the years. In particular, the recent impressive advances in machine vision are based on multilayer (or "deep") convolutional nets [50, 75, 47, 15], which loosely mimic the natural hierarchy of visual processing. Neuro-inspired operations such as local contrast normalization [11, 40], rectification [62] and sparse autoencoding [66]

have been found to be central to improving performance [40]. Most of the best performing nets today are trained in supervised fashion [47, 15, 85]. Despite the state of the art classification accuracy achieved by this approach, there are a number of disconcerting features: a huge number of parameters to be trained, which leads to long training times [47] and the requirement of large amounts of labeled data [36]; lack of a systematic framework for understanding commonly used "tricks" such as DropOut/DropConnect [85]; the requirement for manual tuning of parameters such as learning rate, weight decay and momentum [47]; and the difficulty in interpreting the information being extracted at various hidden layers of the network [88].

In this work, we ask whether we can simplify both implementation and understanding of convolutional architectures, based on combining several key observations. First, while we have at best a coarse understanding of the higher layers of the visual cortex, we should be able to leverage the fairly detailed picture available for the *front end* of the visual system, including retinal ganglion cells (RGCs) and the lateral geniculate nucleus (LGN), along with the simple cells in V1. Thus, it should be possible to engineer machine learning front ends to be *faithfully neuro-mimetic rather than merely neuro-inspired.* Second, we would like to build on the intuition that our visual system extracts a set of "universal" features for any object being viewed, irrespective of whether a classification task is to be performed. Research in the field of *transfer learning* [26], where parameters

of a neural net trained with a dataset have been found to work reasonably well with other datasets, seems to support this assumption. This implies that a system which focuses most of its effort on unsupervised learning for feature extraction, and takes on supervised classification at the end, should have a reasonable chance of success. Indeed, such an approach has been shown to work reasonably well by a few researchers, but further effort is needed to provide classification performance competitive with supervised nets tuned for the purpose of classification. Third, if we shift the focus to unsupervised learning, then the task becomes one of clustering, for which there are simple, well-established algorithms with little need for parameter tuning.

Based on the preceding concepts, we propose and evaluate a convolutional architecture that attains classification performance comparable to the state of the art (beating the state of the art for the NORB image database, and coming close to it for the MNIST handwritten digit database), while lending itself to relatively straightforward interpretation.

Our design approach and contributions are summarized as follows. We would like to mention here that most of the work related to the building of the front end model based on the neuroscience literature has been done by Emre Akbas, a student of Professor Miguel Eckstein in the Psychology Department at UCSB.
(1) As the first part of our neuro-mimetic front end, we build retinal ganglion cells (RGCs) with center-surround characteristics, with center-on cells responding

when the center is brighter than the surround, and the center-off cells responding in the reverse situation. The number of such cells and the receptive cell size are matched to the resolution of the images being processed based on the known parameters of the fovea, the center of the retinal field with the greatest concentration of RGCs. The RGC outputs can be viewed as being directly transported to the lateral geniculate nucleus (LGN), with a one-to-one mapping between RGCs and LGN neurons. Thus, we may view this part of the model as applying to the cascade of the RGC and LGN. We perform local contrast normalization on the RGC/LGN outputs, with the neighborhood used determined by reported experimental parameters. We then rectify these outputs before feeding them to the next layer.

(2) Our second front end stage is a model for V1 simple cells layered on top of RGC/LGN. These are edge detectors constructed using the rough parameters determined by the classical experiments of Hubel and Wiesel [38, 39]. We quantize the edge orientations into bins of width $\pi/8$ (the actual binning in visual cortex may be finer-grained, but we choose a relatively coarse bin size to limit complexity). We use several different kinds of edge detectors, so that there are 48 edge detectors centered at each spatial location. We perform local contrast normalization and rectification on the simple cell outputs. The front end is fixed, with the only tunable parameter being the "viewing distance". (3) Beyond simple cells, neuroscientific guidance sufficient for constructing a complete model of the next

layer is no longer available. We therefore use clustering based on k-means for unsupervised learning henceforth. We first use k-means clustering of outputs from simple cells to obtain centroids (each of which can be interpreted as a neuron). Feature vectors are given by soft assignments to these centroids (which can be viewed as thresholded neuron outputs), and feature vectors from adjacent regions are pooled to obtain the final feature vector. A similar procedure (k-means, soft assignments, and pooling) can be used to build successive layers on top of this. Note that the structure remains convolutional (the same set of centroids slides across the image), but we are zooming out (creating feature vectors for larger segments of the image) as we go up in the hierarchy.

4) After the fixed front end and the unsupervised learning we finally perform classification via supervised learning of a standard support vector machine (SVM) [19] with a radial basis function (RBF) kernel. The best error rates we achieve are: 0.66% on MNIST [50], which is comparable to the best rates reported on this dataset without data augmentation and 2.52% on NORB (uniform-normalized [51]), which improves on the state of the art for this dataset.

Chapter 2

Blind Phase/frequency Synchronization

In this chapter, we discuss the phase/frequency synchronization problem using the mixed signal receiver architecture shown in Fig. 1.1, which implements a very coarse phase quantization. A crucial component of this architecture is the feedback control or the dither signal, whose design constitutes a significant portion of this chapter. We observe that the frequency offsets between transmitter and receiver are typically much smaller than the symbol rate, hence the phase is well approximated as constant over multiple symbols. This enables us to break the synchronization problem into two components: a phase only estimation problem and a frequency tracking problem after the initial phase has been correctly locked. First, we develop a Bayesian algorithm for blind phase estimation, which includes design of the feedback to the analog preprocessor to aid in estimation. Solving for the optimal feedback control policy is equivalent to finding

a solution to a Partially Observable Markov Decision Problem (POMDP) which is computationally intractable. Instead, we propose an information-theoretically motivated greedy strategy that chooses a feedback that evolves with the posterior distribution of the phase. This strategy is easy to implement and as seen via simulations performs almost as well as a genie based optimal strategy. For the tracking step, we use a two-tier algorithm: decision-directed phase estimation over blocks, ignoring frequency offsets, and an extended Kalman filter (EKF) for long-term frequency/phase tracking. The feedback to the analog preprocessor now aims to compensate for the phase offset, in order to optimize the performance of coherent demodulation.

Map of this Chapter: We begin by discussing the related literature on estimation using quantized observations in section 2.1. The system model is described in section 2.2. Next, in section 2.3 we present the derivation of observation probability densities and the formulation of the Bayesian estimator conditioned on the feedback. We end this section by giving two examples that show the importance of carefully designing the feedback signal. In section 2.4 we present the greedy entropy policy for choosing the feedback and place it in the context of related research in the field of designing optimal control for estimation. We end the chapter by presenting the EKF based tracking algorithm in section 2.5.

2.1 Related Work

Section 2.4 describes our proposed feedback policy and the literature related to the field of sequential control and estimation is discussed there. A phase-quantized carrier-asynchronous system model similar to ours was studied in [78]. However, instead of explicit phase/frequency estimation and compensation as in this paper, block noncoherent demodulation, approximating the phase as constant over a block of symbols, was employed in [78]. Whereas a performance degradation of about 2 dB compared to the unquantized block noncoherent case was reported in [78], the algorithm proposed in this paper performs better, with bit error rates almost identical to the unquantized coherent system. Moreover, the analog preprocessing used in the tracking step is simpler compared to the dither scheme proposed in [78]. A receiver architecture similar to ours (mixed signal analog front-end and low-power ADC with feedback from a DSP block) was implemented for a Gigabit/s 60 GHz system in [80], including blocks for both carrier synchronization and equalization. While the emphasis in [80] was on establishing the feasibility of integrated circuit implementation rather than algorithm design and performance evaluation as in this paper, it makes a compelling case for architectures such as those in Fig. 1.1 for low-power mixed signal designs at high data rates. Some of the other related work on estimation using low-precision samples includes frequency estimation [37], amplitude estimation for PAM signaling [81],

channel estimation [21], equalization [84] and multivariate parameter estimation from dithered quantized data [22].

2.2 System Model

We now specify a mathematical model for the receiver architecture depicted in Fig. 1.1. The analog preprocessor applies a phase derotation of $e^{-j\theta_k}$ for the kth sample. In order to simplify digital control of the derotation, we restrict the allowable *derotation values* θ to a finite set of values, denoted by $\mathbb{C}$; in our simulations, we consider a phase resolution of the order of $2\pi/180$. After derotation, the sample is quantized using n 1-bit ADCs into one of $M = 2n$ phase bins: $\left[(m-1)\frac{2\pi}{M}, m\frac{2\pi}{M}\right)$ for $m = 1, ..., M$. In our simulations, we consider $M = 8$ and $M = 12$ (Figs. 2.2(a) and 2.3(a)). As mentioned earlier, such phase quantization can be easily implemented by taking n linear combinations of I and Q samples followed by 1-bit ADCs. For example, $M = 8$ bins can be obtained by 1-bit quantization of I, Q, $I + Q$ and $I - Q$. We always include boundaries coinciding with the I and Q axes, since these are the ML decision boundaries for coherent QPSK demodulation.

Denoting the phase-quantized observation corresponding to the k^{th} symbol by z_k, we therefore have the following complex baseband measurement model:

$$z_k = Q_M\left(\arg\left(b_k e^{j(\phi + k\cdot 2\pi T_s \Delta f)} e^{-j\theta_k} + w_k\right)\right) \tag{2.1}$$

where,

- M:= number of bins over $[0, 2\pi)$ for phase quantization;
- $z_k \in \{1, 2, ..., M\}$ are the observations,
- $Q_M : [0, 2\pi) \to \{1, 2, ..., M\}$ denotes the quantization function, $Q_M(x) = \left\lceil x \cdot \frac{M}{2\pi} \right\rceil$ for $x \in [0, 2\pi)$,
- $b_k \in \left\{e^{j\pi/4}, e^{j3\pi/4}, e^{j5\pi/4}, e^{j7\pi/4}\right\}$ normalized QPSK symbol transmitted, assumed to be uniformly distributed,
- $\phi, \Delta f$:= the unknown phase and frequency offset,
- T_s:= symbol time period,
- $\theta_k \in \mathbb{C} = \{\text{mod}(i \cdot d\theta, 2\pi)\}, i \in \mathbb{I}$, the derotation value for the k^{th} symbol, $d\theta$ denoting the phase resolution,
- w_k:= independent complex AWGN, $\text{Re}(w_k) = \text{Im}(w_k) \sim \mathcal{N}(0, \sigma^2)$, where SNR per bit $= \frac{E_b}{N_0} = \frac{1}{2\sigma^2}$.

The carrier frequency offset Δf is typically of the order of 10-100 ppm of the carrier frequency. For example, for a 60 GHz link, the offset could be as large as 6 MHz, but is still orders of magnitude smaller than the symbol rate, which is of the order of Gsymbols/sec. Thus, it can be set to zero without loss of generality in the acquisition step (described in Sections 2.3 and 2.4), where we derive estimates of

the unknown phase ϕ based on a small block of symbols. We model the frequency offset in the tracking step (Section 2.5).

2.3 Phase Acquisition: Bayesian Estimation

Setting $\Delta f = 0$, the measurement model (2.1) specializes to

$$\begin{aligned} z_k &= Q_M(u_k) \\ u_k &= \arg\left(e^{jp_k\frac{\pi}{4}}e^{j\beta_k} + w_k\right) \\ \beta_k &= \phi - \theta_k \end{aligned} \tag{2.2}$$

where u_k denotes the unquantized phase, β_k is the amount of *net* rotation of the transmitted QPSK symbol. p_k's are independent and uniformly distributed over $\{1, 3, 5, 7\}$, since we are interested in blind estimation (without the use of training symbols). We now drop the subscript k to simplify notation. Conditioned on β the density of u is given by (derivation is presented in the appendix A.1):

$$\begin{aligned} f_u(\alpha;\beta) &= \sum_{i=1}^{4}\frac{1}{4}f_{u|p=2i-1}(\alpha;\beta) \quad ; \quad \alpha \in [0, 2\pi) \\ f_u(\alpha;\beta) &= \sum_{i=1}^{4}\frac{1}{4}\left[\frac{a_i(2-\mathrm{erfc}(\frac{a_i}{\sigma\sqrt{2}}))e^{\frac{a_i^2-1}{2\sigma^2}}}{2\sigma\sqrt{2\pi}} + \frac{e^{-\frac{1}{2\sigma^2}}}{2\pi}\right] \\ &\text{where} \quad a_i = \cos\left((2i-1)\frac{\pi}{4} + \beta - \alpha\right) \end{aligned} \tag{2.3}$$

For $\beta = 0$ define $f_u(\alpha) := f_u(\alpha; 0)$. We can infer from the expression above that the density at non-zero values of β can be evaluated simply by circular shifts (by

2π) of $f_u(\alpha)$. Due to the uniform distribution over the QPSK constellation, $f_u(\alpha)$ is periodic with period 90° (as seen in Fig. 2.1). Distribution of the quantized measurements conditioned on β (ϕ and θ) is expressed in terms of the integrals of $f_u(\alpha)$ as follows:

$$p_\phi^\theta(z=m) = P\left(z=m|\beta\right) = \int_{(m-1)\frac{2\pi}{M}}^{m\frac{2\pi}{M}} f_u(\alpha;\beta)\, d\alpha \tag{2.4}$$

$$\text{where} \quad m \in \{1, 2, ..., M\}$$

The single step likelihood of the phase offset, conditioned on the phase measurement in bin m and derotation $\theta = 0$, is given by $l(\phi|m) = \log(p_\phi^0(z = m))$. Nonzero θ simply results in a circular shift of $l(\phi|m)$. Due to the periodicity of $f_u(\alpha)$, it suffices to limit ϕ to the interval $[0, 90°)$. The Bayesian estimator, as discussed next, essentially involves successively adding these single step likelihoods as more measurements are made. An interesting property to note is the periodicity of $l(\phi|m)$ in m with period $M/4$, which follows from the symmetry induced by equiprobability of the transmitted symbols. For example, if $M = 8$ (Fig. 2.2(a)), a measurement z in bin 1 or bin 3 results in the same likelihood function. Fig. 2.1 shows the three distinct likelihoods for $M = 12$ (6 one-bit ADCs).

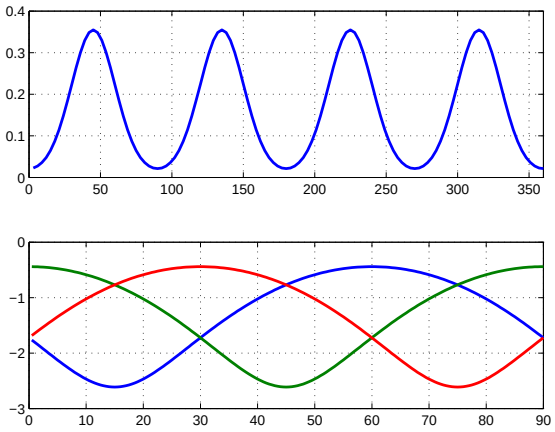


Figure 2.1: (top) Probability Density of unquantized phase u at $\beta = 0$, $f_u(\alpha)$ **(bottom)** Single step likelihoods $l(\phi|m)$ given $z = m$ and $\theta = 0°$ ($M = 12$, SNR=5dB). blue: $l(\phi|1) = l(\phi|4) = l(\phi|7) = l(\phi|10)$, green: $l(\phi|2) = l(\phi|5) = l(\phi|8) = l(\phi|11)$, red: $l(\phi|3) = l(\phi|6) = l(\phi|9) = l(\phi|12)$ (The plot is best viewed in color)

2.3.1 Bayesian Estimation given Derotation Phases θ_k

Conditioned on the past derotation values θ_1^k (which are known) and the quantized phase observations z_1^k, applying Bayes rule gives us a recursive equation for updating the posterior of the unknown phase as:

$$p(\phi|z_1^k, \theta_1^k) = \frac{p(z_k|\phi, \theta_k)p(\phi|z_1^{k-1}, \theta_1^{k-1})}{p(z_k|\theta_k)} \tag{2.5}$$

Normalizing the pdf obviates the need to evaluate the denominator. We now go to the log domain to obtain an additive update for the cumulative log likelihood. Denoting by $l_{1:k}(\phi) = \log\left(p\left(\phi|z_1^k, \theta_1^k\right)\right)$ the cumulative update up to the k^{th} symbol, we update it recursively simply by adding the single step update

$l_k(\phi) = \log\left(p\left(z_k|\phi, \theta_k\right)\right)$, as follows:

$$l_{1:k}(\phi) = l_{1:k-1}(\phi) + l_k(\phi) \tag{2.6}$$

The maximum a posteriori (MAP) estimate after N symbols is given by

$$\hat{\phi}_{\text{MAP};N} = \text{argmax } \; p(\phi|z_1^N, \theta_1^N) = \text{argmax } \; l_{1:N}(\phi)$$

We start with a uniform prior $p(\phi)$ over $[0°, 90°)$. Single step likelihoods, $l(\phi|m)$ for $m = 1, ..., M/4$, can be precomputed and stored offline, and circularly shifted by the derotation phase θ_k as the estimation proceeds. The recursive update (2.6) requires only the latest posterior to be stored.

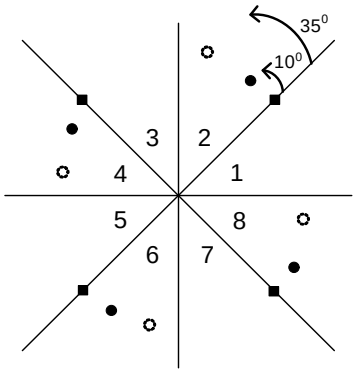
2.3.2 Choosing Derotation Phases θ_k: Two Examples

Setting the values of the derotation phases provides a means of applying a *controlled dither* prior to quantization. In the next section, we investigate whether it could be used for *speeding up* the phase acquisition. We start by looking at two motivating scenarios where the naive strategy of setting $\theta_k = \text{constant } \; \forall \; k$ fails to give satisfactory results.

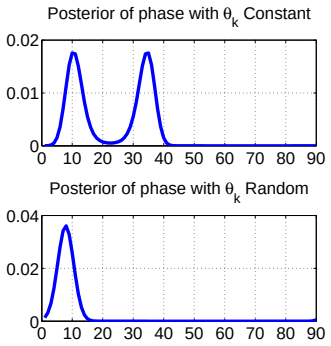
Example 1: Consider 8 phase quantization bins and $\phi = 10°$ (Fig. 2.2). Choosing $\theta_k = 0° \; \forall \; k$ results in a bimodal posterior with a spurious peak at $\phi = 35°$. Due to symmetry of the phase boundaries and equiprobable distribution over the transmitted symbols, the set of observations (1,3,5,7) and (2,4,6,8) leads to the

posterior being updated in identical ways. With probability of getting bin 3 for $\phi = 35^\circ$ being equal to the probability of getting bin 1 for $\phi = 10^\circ$, there is an unresolvable ambiguity between the two phases. In general for any phase α, we have $P\left(z_k = i|\phi = \alpha, \theta_k = 0\right) = P\left(z_k = j|\phi = 45^\circ - \alpha, \theta_k = 0\right)\ \forall\ i, j \in \{1, 3, 5, 7\}$ or $\forall\ i, j \in \{2, 4, 6, 8\}$; which gives rise to a bimodal posterior with peaks at α and $45^\circ - \alpha$. Such ambiguities were also noted in the block noncoherent system considered in [77]. One approach to alleviate this ambiguity is to dither θ_k randomly; this dithers the spurious peak while preserving the true peak, leading to a unimodal distribution for the posterior computed over multiple symbols. Another approach is to break the symmetry in the phase quantizer, using 12 phase bins instead of 8. However, even this strategy can run into trouble at very high SNR, as shown by the next example.

Example 2: Now consider 12 phase bins and no noise (or very high SNR), again with true phase offset $\phi = 10^\circ$. Since there is no noise, all observations fall in bins 2,5,8,11, resulting in a flat phase posterior over the interval $[75^\circ, 90^\circ] \cup [0^\circ, 15^\circ]$ if there is no dither ($\theta_k \equiv 0^\circ$). This could lead to an error as high as 25° (Fig. 2.3). On the other hand, using randomly dithered θ_ks results in an accurate MAP estimate, with the combination of shifted versions (shifted by θ_k) of the flat posterior leading to a unimodal posterior with a sharp peak.



(a) $\phi = 10°$

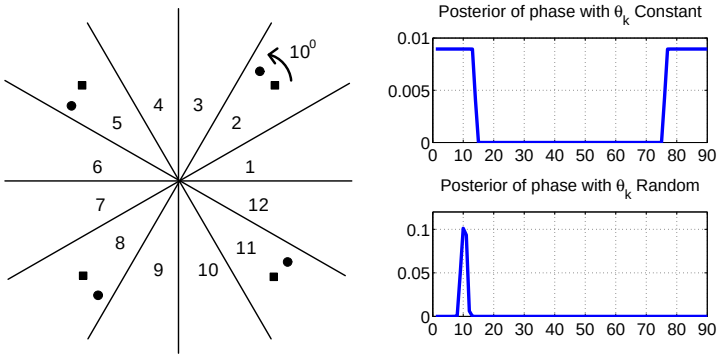


(b) Posterior for ϕ after 100 symbols (top) Derotation value θ_k kept constant (bottom) θ_k varied randomly

Figure 2.2: Example 1: SNR=5dB, 8 uniform quantization regions

2.4 Phase Acquisition: Feedback Control

While randomly dithered derotation is a robust design choice which overcomes the shortcomings of the naive strategy of no dither, it is of interest to ask whether we can do better. In particular, we are interested in finding a dither strategy that reduces the mean square error of the phase estimate *faster* (i.e. requiring fewer symbols), compared to the random dither. The problem concerning us here belongs to the category of problems related to sequential estimation and control, which has a large body of research. Most of the relevant references can be found in the following recent papers : [61, 65, 9], which discuss control policies for multihypothesis testing, and [4] which looks at control for estimating a continuous



(a) $\phi = 10°$

(b) Posterior for ϕ after 30 symbols (top) Derotation value θ_k kept constant (bottom) θ_k varied randomly

Figure 2.3: Example 2: SNR=35dB, 12 uniform quantization regions

valued parameter, a scenario similar to ours. These problems are either set over a finite horizon, then the goal is to find the control policy that minimizes a metric like the mean square error at the end; or over a variable horizon and the cost function to be minimized is the sum of the expected number of observations plus a penalty term for the final estimation being wrong (for the continuous case this could correspond to the expected mean square error). In the latter case, a stopping criterion also needs to be provided. As discussed in the literature, both these formulations can be mapped to a Partially Observable Markov Decision Problem (POMDP), which is intractable to solve optimally. The approach then is to either employ approximate solutions (which can still be very complex) or focus on

characterizing asymptotically optimal solutions (in the limit of large number of observations and a large coefficient for the penalty term). References [61, 65, 4] discuss the latter approach. Hence the results obtained in these references are not directly applicable to our problem since the phase estimation is done over a span of a few tens of symbols/observations.

In the context of our problem, we find that a simple and intuitive policy which we call the *Greedy Entropy Policy* (GE) performs really well and is close to being optimal as demonstrated by the numerical results. The idea is to pick an action at each step that minimizes the *expected* entropy (an information theoretic measure of uncertainty) of the next step phase posterior. A similar policy has been discussed in the multihypothesis setting in [61] and used to derive theoretical bounds for the cost function with the penalty term. Reference [4] proposes a policy that involves maximizing the fisher information at each step, based on the latest MAP estimate of the parameter. We hereafter refer to this policy as MFI and discuss its details later. Since their problem setup is similar to ours, MFI is directly applicable to our scenario. The authors of [4] prove that MFI is asymptotically optimal but do not comment on its performance for small n. We find that GE converges to MFI as the number of observations increase, but performs better for small n, especially at low SNR. It can be easily shown that GE is equivalent to a policy, which at each step, greedily maximizes the mutual information between the new observation and the unknown phase offset. In this form it is identical to the policy discussed in

[9, 46]. In these references, mutual information between the unknown hypothesis and the set of observations over a finite horizon is used as the cost function, which is to be maximized. They show that the greedy approach achieves a value which is within a constant factor of the optimal cost function and is the best among all polynomial time algorithms. These guarantees naturally translate to our problem as well, however unlike [9, 46] we are more interested in minimizing the mean square error of the phase.

In the beginning of this section, we first discuss these two policies assuming the consistency of the MAP estimate, i.e. even with constant action the posterior converges to a unimodal distribution centered around the true value of phase. This always holds true for the M=12 case with nonzero noise. We then analyze the special case of zero noise separately, when the phase posteriors are flat and the MAP estimate is ill-defined. We show that in this case GE reduces the support of the posterior density by half at every step, thereby reducing the absolute error at an exponential rate. Finally, we discuss a simple strategy, based on randomly choosing actions at regular intervals, for ensuring a consistent unimodal posterior when M=8.

2.4.1 Greedy Entropy Policy

At step $k-1$ (i.e. after observing $k-1$ symbols) the net belief about the phase is captured by the posterior $f_{k-1}(\phi) := p\left(\phi|z_1^{k-1}, \theta_1^{k-1}\right)$. For simplifying the notation, we drop the subscript k as the equations described below remain same for all k. The entropy of the *current belief*, $f(\phi)$ is given by

$$h(f(\phi)) = -\int f(\phi)\log(f(\phi))d\phi \tag{2.7}$$

The new posterior, conditioned on the next action $\theta = \theta_k$ and observation $z = z_k$, is given by

$$f_{\text{new}}(\phi|\theta, z) = \frac{p_\phi^\theta(z)f(\phi)}{p^\theta(z)} \tag{2.8}$$

where $p_\phi^\theta(z)$ represents the conditional distribution of the observation (Eq. 2.4) given the true phase offset, ϕ, and the derotation action, θ. The normalization term in the denominator is the probability density of observing z in the next step under the effect of taking action θ, averaged over the current belief, i.e.

$$p^\theta(z) = \int p_\phi^\theta(z)f(\phi)d\phi \tag{2.9}$$

We can now compute the *expected* entropy of the new posterior if action θ is chosen, by averaging over the observation density $p^\theta(z)$

$$h^\theta(f_{\text{new}}(\phi)) = E_z\left[h(f_{\text{new}}(\phi|\theta, z)\right] = \sum_{i=1}^{M} p^\theta(z_i)h(f_{\text{new}}(\phi|\theta, z)) \tag{2.10}$$

The GE policy chooses the derotation phase that minimizes the entropy of the new posterior, i.e.

$$\theta_k = \underset{\theta}{\operatorname{argmin}}\ h^{\theta}(f_{\text{new}}(\phi)) \tag{2.11}$$

$$\Rightarrow \theta_k = \underset{\theta}{\operatorname{argmax}}\ \big(h(f\phi) - h^{\theta}(f_{\text{new}}(\phi))\big) = \underset{\theta}{\operatorname{argmax}}\ IU^{\theta} \tag{2.12}$$

Eq. (2.12) presents another way in which the policy can be expressed, i.e. maximization of the *information utility*, IU^{θ}, which is the amount by which the uncertainty (entropy) is decreased due to the action θ. Information utility can be expressed in terms of the Kullback-Leibler Divergence, which is useful for proving its equivalence to MFI as discussed later. Simple arithmetic manipulations using Eqs. (2.12), (2.7), (2.8) gives

$$IU^{\theta} = \int f(\phi) D^{\theta}(\phi) d\phi \tag{2.13}$$

where $D^{\theta}(\phi)$ is the KL divergence between densities $p_{\phi}^{\theta}(z)$ and $p^{\theta}(z)$

$$D^{\theta}(\phi) = \sum_i p_{\phi}^{\theta}(z_i) \log \frac{p_{\phi}^{\theta}(z_i)}{p^{\theta}(z_i)} \tag{2.14}$$

It is straightforward to implement the greedy entropy policy by evaluating the information utility (Eq. 2.13) over the finite set of actions. In the next subsection we discuss its relationship with the Fisher Information.

2.4.2 Fisher Information

Fisher information provides a measure of the *sensitivity* of the estimation problem to the value of the parameter being estimated. Parameter values that result in higher fisher information can be estimated with greater accuracy or fewer measurements. The Cramer-Rao bound, which is the inverse of the fisher information, provides a lower bound on the mean square error for any unbiased estimator. For the phase offset estimation problem, the fisher information as a function of the true phase offset and the derotation action, is given by:

$$FI^{\theta}(\phi) = \sum_{i=1}^{M} \left(\frac{\partial p_{\phi}^{\theta}(z_i)}{\partial \phi} \right)^2 \cdot \frac{1}{p_{\phi}^{\theta}(z_i)} \tag{2.15}$$

The derivative of the observation density $p_{\phi}^{\theta}(z)$ can be easily computed by differentiating the function $f_u(\cdot)$ prior to integration (Eqs. 2.3 and 2.4). In Fig. 2.4 we plot the fisher information as a function of the phase offset (θ has been set to 0) for 4 different cases: SNR low or high and number of regions (M) equal to 8 or 12. We observe that in three of the cases, fisher information is maximum for phase offsets that bring the final phase after rotation to the "boundary" i.e. one of the bin edges. This is intuitive at high SNR. Note that the net phase is the phase offset ϕ plus the original QPSK phase $i \cdot \frac{\pi}{4},\ i = 1, 3, 5, 7$ (plus $-\theta$ but that is 0 here). Note that if the complex QPSK symbol ends up being in the "middle" of the quantization bin, and the SNR is high, the same measurement would be recorded at every symbol period, resulting in a flat posterior which is bad for esti-

mation. Interestingly, when the noise is high enough to knock the symbol around a lot more and the bins are narrower ($M = 12$), fisher information is maximized for a phase offset (30°) that brings the symbol to the "middle" of the quantization cone (Fig. 2.4(d)) (for instance, if the QPSK symbol $\frac{\pi}{4}$ is transmitted, the net phase is $30° + 45° = 75°$ which is exactly in between the phases thresholds at angles 60° and 90°).

The fisher information computations provide us with a "genie" optimal control policy i.e. the best action for any given phase offset value is the one that brings the net phase to a value for which the fisher information is maximized. Of course, in practice we cannot implement such a policy since knowing the true phase would obviate the need for phase estimation in the first place. However, we can use the maximal fisher information value to compute the Cramer-Rao bound which provides us a benchmark for bounding the MSE performance of the optimal control policy (and hence any other policy).

We do not know the true value of the phase offset, however in place of that we can use our best *guess*, which is the latest MAP estimate. This leads to the 'maximizing fisher information' (MFI) policy which chooses actions at each step as follows:

$$\theta = \underset{\theta}{\text{argmax}}\ FI^{\theta}(\phi_{MAP}); \text{where}\ \ \phi_{MAP} = \underset{\phi}{\text{argmax}}\ f(\phi) \tag{2.16}$$

where $FI^{\theta}(\phi)$ is computed via Eq. (2.15). $f(\phi)$ is the latest belief/posterior distribution of the phase offset. MFI chooses optimal actions if the MAP estimate is close to the true offset. This becomes increasingly true as the number of observations increase. Indeed in reference [4], it was been shown to be asymptotically optimal under consistency assumptions. However when the uncertainty in $f(\phi)$ is high, we expect a policy that takes into account the distribution, such as the GE, to perform better. MFI may not be ideal during the initial stages when the MAP estimate can be quite bad. In fact, the simulation results presented later demonstrate that in the case of high noise and coarser quantization, when the MAP estimate takes a while to settle near the true value, GE performs slightly better than the MFI policy. It is not surprising that as the uncertainty in $f(\phi)$ reduces and the estimator becomes more confident of the MAP estimate, the GE policy reduces to MFI. This is proved in the following theorem.

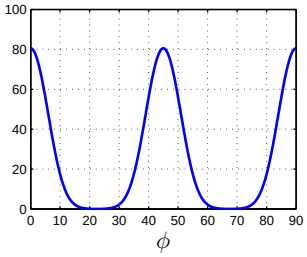
Theorem 1. *Given that the latest phase posterior is normally distributed, i.e.* $f(\phi) \sim \mathcal{N}(\phi_0, v^2)$ *where* v *is in the unit of radians; then as the variance becomes smaller, the greedy entropy policy chooses the same actions as the maximizing fisher information policy, i.e.*

$$\lim_{v \to 0} \operatorname*{argmax}_{\theta} IU^{\theta} = \operatorname*{argmax}_{\theta} FI^{\theta}(\phi_0) \tag{2.17}$$

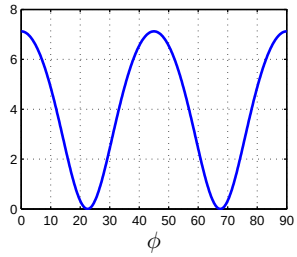
Specifically

$$\lim_{v \to 0} \frac{IU^{\theta}}{v^2} = \frac{1}{2} FI^{\theta}(\phi_0) \tag{2.18}$$

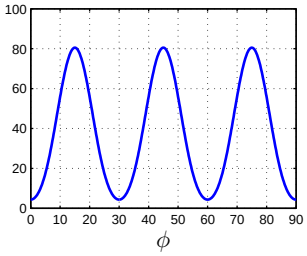
The proof is provided in the appendix (A.2). Note that $f(\phi)$ is not strictly Gaussian as its support is $[0, \frac{\pi}{2})$. However, when consistency of the estimate is guaranteed and as the number of observations increase, the property of asymptotic distribution of MLE estimators ensures that $f(\phi)$ approaches the Gaussian density with ϕ_{MAP} as the mean. The theorem then kicks in; in fact in our simulations we find that the equation $\underset{\theta}{\text{argmax}}\ IU^{\theta} \approx \underset{\theta}{\text{argmax}}\ FI^{\theta}(\phi_0)$ starts becoming true as soon as the standard deviation of $f(\phi)$ is within a few degrees. We also note from the theorem that the value of the information utility scales with the variance of the posterior density, independent of the actions.



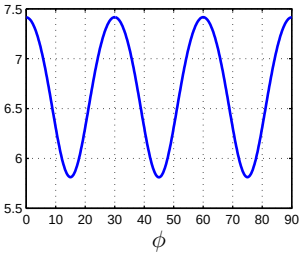
(a) $M = 8$ and SNR=15dB



(b) $M = 8$ and SNR=5dB



(c) $M = 12$ and SNR=15dB



(d) $M = 12$ and SNR=5dB

Figure 2.4: Fisher Information as a function of ϕ ($\theta = 0$)

2.4.3 Zero Noise Case

As discussed earlier, when SNR is very high, the resulting posterior density is flat i.e. uniform over a support interval determined by the set of observations. In this case a dither is really important, as keeping θ_k fixed results in the same measurement and no change in posterior. This is a common feature with all systems involving quantized measurements: at high SNR, dither acts as artificial noise and provides the necessary diversity of measurements required for estimation. In this zero noise case, the posterior remains always flat, only the support changes as we change the action. GE is equivalent to choosing the action that reduces the support the most and is hence optimal. This is established via the following lemma, whose proof is discussed in the appendix (A.3).

Lemma 1. *In the absence of noise (i.e. $w_k = 0\ \forall k$ in Eq. (2.2)), the phase posterior $f_k(\phi)$ is a uniform density for all values of k. Let S_k denote the size of its support at time k. The action chosen by the Greedy Entropy policy is the one that minimizes the expected value of S_{k+1}. Furthermore, $S_{k+1} = \frac{1}{2}S_k$, hence the absolute phase error reduces exponentially at the rate of $\frac{1}{2}$. Although MFI is not well defined as there is no unique MAP estimate, but if the MMSE estimate is used instead in Eq. (2.16), MFI chooses the same actions as the GE policy.*

2.4.4 Avoiding the phase ambiguity for $M = 8$ case

Till now we have assumed that the ϕ posterior always converges to the correct phase offset irrespective of the sequence of actions taken. And this is indeed true when $M = 12$, for which the MAP estimate is always consistent. This is because for any action θ, different values of the true phase offsets result in distinct observation densities. This can be expressed mathematically as follows

$$\text{for any } \phi \neq \phi',\ D\left(p_{\phi}^{\theta}||p_{\phi'}^{\theta}\right) > 0 \ \forall\ \theta\ (M = 12) \tag{2.19}$$

However when $M = 8$, the above condition does not hold. Due to the symmetry of the angular thresholds, for any given value of ϕ and a given derotation θ, there exists another phase offset, ϕ', which results in an identical distribution over the quantized measurements. This means that if θ is kept constant, the limiting posterior $f(\phi)$ is bimodal, with true and spurious peaks at locations ϕ and ϕ' respectively. Value of ϕ' is a function of ϕ (which remains fixed) and θ. The lemma below specifies this relationship.

Lemma 2. *When $M = 8$ and the true phase is denoted by $\phi \in [0, \frac{\pi}{2})$, for any derotation phase θ, there exists an value $\phi' \in [0, \frac{\pi}{2}) \neq \phi$, such that $D\left(p_{\phi}^{\theta}||p_{\phi'}^{\theta}\right) = 0$. This holds for $\phi' = \mod\left(2\theta - \phi + \frac{\pi}{4}, \frac{\pi}{2}\right)$.*

The proof, which is fairly straightforward, is discussed in the appendix (A.4). We see that a constant dither policy is unacceptable as it leaves a bimodal ambiguity in the value of the phase offset. Any other policy in which θ_k does not

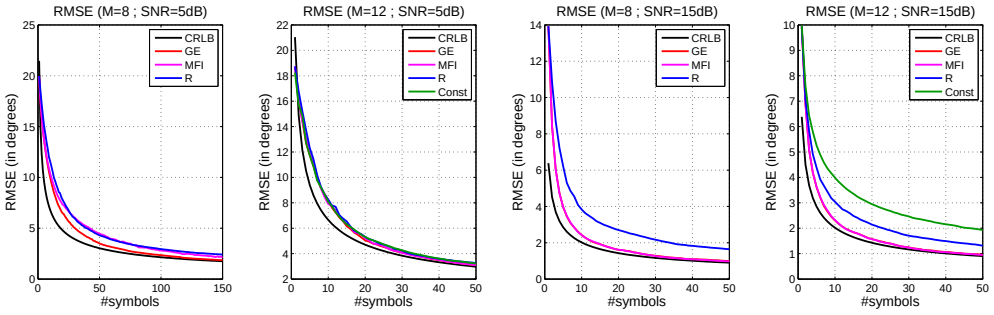
remain perfectly constant, is generally expected to eliminate bimodality, but may run into certain issues sometimes. A random dither continuously changes θ and thereby *guarantees* a correct unimodal limiting posterior. The same, however, does not necessarily hold true for the GE or MFI policies. Interestingly, with either of these policies, there is also a chance, albeit with low probability, of the final posterior being single peaked at the spurious phase offset value. This can happen in the following manner: suppose a total of N measurements are made, out of which a majority, say $N_1 \approx N$ employed a constant action (this can happen, say with MFI if ϕ_{MAP} remains same). In the remaining few steps, $N_2 = N - N_1$, different value(s) of θ were used. Recall that the final ϕ posterior is just a summation of the individual step log likelihoods, the order being irrelevant. Now it may happen that these few N_2 observations are affected by bad noise instances and the ϕ posterior, computed based on just these steps, has a larger probability mass at the spurious value. Since the posterior distribution from the other N_1 steps is perfectly bimodal, the net combined posterior ends up having a much stronger peak at ϕ'. Note that the chance for such an event is generally very small, as it requires getting multiple bad measurements during which ϕ' should appear to be more probable. However we have observed it to happen once in a while during our monte carlo runs.

A simple modification to the policies MFI/GE can guarantee vanishing probabilities for such bad events. The idea is to pick the actions randomly at regular

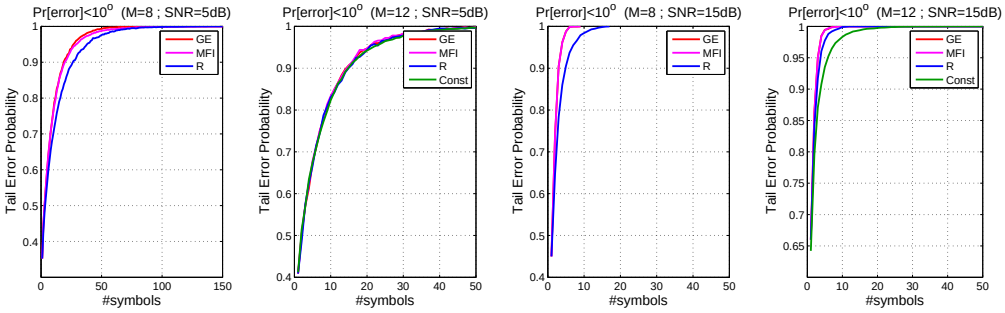
intervals for a fixed fraction, γ, of the steps. For instance, $\gamma = 0.1$ means choosing every 10^{th} action randomly, while the rest are chosen in the usual manner as dictated by the policy being employed. As N tends to infinity, the number of random dither steps γN tends to infinity as well (for any non-zero value of γ), thereby ensuring that the limiting posterior is unimodal and converges to the correct phase. Note that a more efficient scheme can also be used, as described in the reference [65], where they propose a schedule that employs randomly chosen actions at sampling times that grow exponentially. However, in our problem setup, where we are concerned with typically less than 100 measurements, the fixed rate schedule works well with almost no change in the efficiency of the GE/MFI policies.

2.4.5 Simulation Results

The performance of phase acquisition is evaluated using Monte Carlo simulations averaging over randomly generated channel phases. Fig. 2.5 plots results for two values of SNR: a low value of 5 dB and a high value of 15 dB. The performance measures are the root mean squared error (RMSE), which captures average behavior, and the probability of the phase error being smaller than a threshold, which captures the tail behavior. Errors are computed modulus 90°, for instance if the true phase offset is 80° and the estimate is 5°, this is equivalent to an error of 15°. We implement three policies: greedy entropy (GE), random dither (R) and



(a)



(b)

Figure 2.5: Results of Monte Carlo simulations of different strategies for choosing the feedback θ_k with 4 and 6 ADCs (8 and 12 phase bins) at SNRs 5dB and 15dB. Policies: Greedy Entropy (GE), Maximizing Fisher Information (MFI), Random dither (R) and Constant derotation phase (Const)

maximizing the fisher information (MFI). We also simulate the policy of keeping the derotation phase constant when M=12, the case for which it is consistent. For comparison we plot the CRLB computed by inverting the maximal fisher information (maximum over different values of the true phase offset keeping $\theta = 0$), this gives the performance of the genie optimal strategy. However, note that this does not give a valid lower bound when the number of measurements are few and the errors can be large. This is because the Cramer-Rao bound is based on the

standard notion of squared error, not the circular modulus error which is more appropriate in this problem. This is not much of an issue as more observations are made and the error reduces, the two notions of computing error become same with increasing probability. From the plots, we make the following observations: (a) The performance of GE is very close to the "genie" optimal control policy (CRLB) in all cases. (b) GE and MFI performance are almost identical, GE is slightly better at low SNR and coarser quantization (4ADCs, 5dB), when MAP estimate can be bad initially and MFI relies too much on it. (c) At low SNR, there is little to distinguish between random dithering and GE, since the noise supplies enough dither to give a rich spread of measurements across different bins. In fact at low noise and finer quantization (5dB, 12 bins), the constant action performs as well as others. However, when the quantization is more severe (8 bins), the greedy entropy policy provides performance gains over random dithering even at low SNR. To summarize, we find that efficient dithering policies could be effective for rapid phase acquisition under the scenarios of more severe quantization and higher SNRs.

Once an accurate enough phase estimate is obtained in the acquisition step, we wish to begin demodulating the data, while maintaining estimates of the phase and frequency. In the next section, we describe an algorithm for decision directed (DD) tracking. In this DD mode, the phase derotation values θ_k aim to correct for

the channel phase to enable accurate demodulation, in contrast to the acquisition phase, where the derotation is designed to aid in phase estimation.

2.5 Phase/Frequency Tracking

We must now account for the frequency offset in order to track the time-varying phase, and to compensate for it via derotation in order to enable coherent demodulation. The phase can be written as $\phi(k) = \phi_0 + 2\pi k T_s \Delta f = \phi_0 + k\eta$, where η is the *normalized frequency offset,* defined as the rate of change of phase in radians per symbol. To get a concrete idea of how fast the phase varies, consider the following typical values: $f_c = 60$ GHz, bandwidth of 6 GHz, i.e. $Ts = (6 \times 10^9)^{-1}$ secs, an offset $\Delta f = 100ppm \cdot f_c$, which leads to $\eta = 2\pi T_s \Delta f = 2\pi \cdot 10^{-3}$ *radians*; a linearly varying phase rate of 0.36° per symbol. We can therefore accurately approximate the phase as roughly constant over a few tens of symbols, while obtaining an accurate estimate of the frequency offset η would require averaging over hundreds of symbols. This motivates a hierarchical tracking algorithm. Bayesian estimates of the phase are computed over relatively small windows, modeling it as constant but unknown. The posterior computations are as in the previous section, with two key differences: the derotation phase value is our current best estimate of the phase, and we do not need to average over the possible symbols, since we operate in decision-directed mode. These

relatively coarse phase estimates are then fed to an extended Kalman filter (EKF) for tracking both frequency and phase. The filter is initialized with the phase estimate as derived in the previous section. Note that the data is differentially encoded over the QPSK symbols (this is necessary as the phase estimation was performed modulo $\frac{\pi}{2}$ in the acquisition stage).

Denote by $\hat{\phi}_{\mathrm{MAP};W}(k)$ the MAP phase estimate over a sliding window of W symbols. This is fed as a noisy measurement of the true time varying phase $\phi(k)$ to an EKF constructed as follows:

Process Model

$$x_k = Ax_{k-1} + w_k$$

$$\begin{bmatrix} \phi(k) \\ \eta(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phi(k-1) \\ \eta(k-1) \end{bmatrix} + w(k)$$

where $w(k) \sim \mathcal{N}(0, Q_k)$ is the process noise, the state vector comprises the phase and the normalized frequency offset $x_k = [\phi(k)\ \eta(k)]^T$ and the state evolution matrix $A = [1\ 1; 0\ 1]$. Note that Q_k is of the form $\sigma_p^2 \cdot [1\ 1; 1\ 1]$ since the same noise term influences both the phase and frequency offset i.e. $\eta(k) = \eta(k-1) + w_k(2)$, and $\phi(k) = \phi(k-1) + \eta(k) = \phi(k-1) + \eta(k-1) + w_k(2)$, hence $w_k(1) = w_k(2)$.

Measurement Model

$$y_k = h(x_k) + v_k$$

$$y(k) = \begin{bmatrix} \cos(4 \cdot \hat{\phi}_{\mathrm{MAP};W}(k)) \\ \sin(4 \cdot \hat{\phi}_{\mathrm{MAP};W}(k)) \end{bmatrix} = \begin{bmatrix} \cos(4 \cdot \phi(k)) \\ \sin(4 \cdot \phi(k)) \end{bmatrix} + v(k)$$

where $h(\cdot)$ is a non linear measurement function. The particular form is chosen to resolve the issue of unwrapping the phase periodically as it grows linearly: the factor of 4 inside the sine and cosine arguments chosen to obtain a period of 90°, since we are only interested in phase estimates over the range $[0, \pi/2]$. The measurement noise is $v(k) \sim \mathcal{N}(0, R_k)$. For the EKF, computation of the Jacobin of the nonlinear function $h(\cdot)$ is required, which in this case evaluates to

$$H_k = \begin{bmatrix} -4\sin(4\phi(k)) & 0 \\ 4\cos(4\phi(k)) & 0 \end{bmatrix}$$

The EKF update equations are given as follows (we refer the readers to Chapter 10 of [7] for a discussion on EKF, and to [70] for a somewhat similar application

of EKF for phase tracking).

Time Update:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1}$$

$$\hat{P}_{k|k-1} = A\hat{P}_{k-1}A^T + Q_k$$

$$K = \hat{P}_{k|k-1}H_k^T \left(H_k\hat{P}_{k|k-1}H_k^T + R_k\right)^{-1}$$

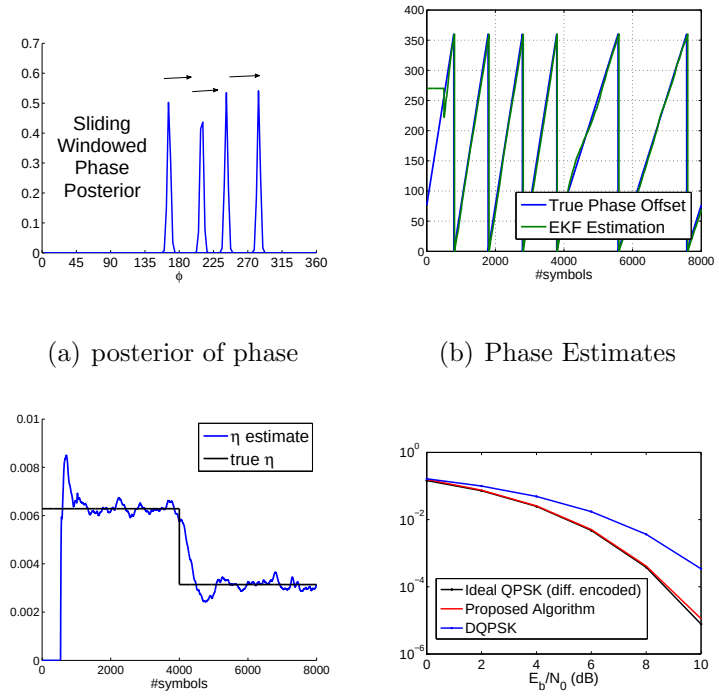
Measurement Update:

$$\hat{x}_k = \hat{x}_{k|k-1} + K\left(y_k - h(\hat{x}_{k|k-1})\right)$$

$$\hat{P}_k = (I - KH_k)\,\hat{P}_{k|k-1}$$

$\hat{P}_k$ is the estimate of the state error covariance and H_k is evaluated at $\hat{x}_{k|k-1}$. The *cleaned* state estimate, $\hat{x}_k$, provides the *latest* estimate of the frequency offset $\hat{\eta}(k) = \hat{x}_k(2)$ and a *delayed* estimate of the net phase, delayed due to the effect of sliding window. The measurement at time k, y_k, reflects the phase estimated over the time window $[k-W, k]$, hence the feedback (for undoing the phase at time k) is set according to $\theta_k = \hat{x}_k(1) + \frac{W}{2} \cdot \hat{\eta}(k)$.

Tuning the filter: Although the measurement noise covariance R_k can be calculated from the variance of the posterior of the phase, constructed over the sliding window, the filter performance was observed to be quite robust to the choice of R_k over a range of SNR. For the simulations presented in this paper, we assumed a constant $R_k = [0.1\ 0, 0\ 0.1]^T$, which worked well for SNRs 0-15dB and sliding window length of $W = 50$ symbols. The scaling of the process noise (Q_k) trades



(a) posterior of phase

(b) Phase Estimates

(c) Frequency offset estimate

(d) Bit Error Rate Plots

Figure 2.6: Performance plots of EKF based Tracking Algorithm

off steady state versus tracking performance: small Q_k results in accurate estimates but slow reaction to abrupt changes in frequency, while large Q_k improves the response to abrupt changes at the expense of increased estimation error. Since the ultimate measure of performance is the bit error rate (BER) rather than the phase estimation error itself, a sensible approach to design is to set Q_k to the largest value (and hence the fastest response to abrupt changes) compatible with phase estimation errors causing a desired level of degradation in BER relative to ideal coherent demodulation.

2.5.1 Simulation Results

Fig. 2.6 shows the tracking algorithm in action. We have used M=8 bins. Subplot 2.6(a) shows several superimposed snapshots of the windowed posterior of the phase, whose peaks (the MAP estimates) are used as measurements for the EKF. In subplot 2.6(c) η was changed from $2\pi \cdot 10^{-3}$ to $\pi \cdot 10^{-3}$ after 4000 symbols. The plot shows $\hat{\eta}$, the estimate, for choosing $Q_k = 5 \times 10^{-11}[1\ 1; 1\ 1]^T$ which enables the filter to lock onto the new value in about 1000 symbols. The last subplot 2.6(d) shows BER curves for ideal (unquantized) coherent QPSK and that of the proposed algorithm, which is almost indistinguishable from the former. Using noncoherent differential QPSK (DQPSK) obviates the need for phase synchronization but results in a 2dB performance degradation.

Chapter 3

Slicer Architectures for Analog-to-Information Conversion in Channel Equalizers

Our focus in this chapter is to explore A/D front end designs for achieving optimal BER performance with highly constrained ADC resolution. We consider antipodal signaling through a static dispersive channel with a finite, short to moderate length memory ($\sim 4-8$ symbol periods). Such channels are commonly encountered in high speed ($\sim$ 10Gbps) backplane wireline links.

A popular architecture for high sampling rates is the Nyquist sampled flash ADC, which is comprised of comparators/slicers with thresholds typically spread uniformly over the input signal range. While this architecture is suitable for minimizing the reconstruction error of the received signal, it is not efficient as an analog-to-information converter for recovering the bits sent over the communication link. This is certainly the case for a dispersive channel, but even for

non-dispersive channels, it has been shown that Nyquist sampling can be suboptimal (in terms of channel capacity) in the presence of heavy quantization. We investigate generalizations of the flash ADC for analog-to information conversion, as opposed to the standard objective of waveform preservation, for communication over dispersive channels. Our goal is to understand the performance-complexity trade off space when the slicers are allowed operate at different sampling phases and non-uniform thresholds, with the parameters potentially adapted to the channel. Since the power consumption of a high resolution ADC increases enormously with the sampling rate, and directly scales with the number of slicers used by the ADC (a $\log_2(n+1)$-bit ADC employs n slicers), we want to keep the number n as low as possible.

Map of this Chapter: We first present the related work in section 3.1. System setup is described in section 3.2. Section 3.3 discusses the performance of the uniform ADC architecture and presents its limitations in terms of being sensitive to the channel and the sampling phase. This motivates an architecture that explores multiple sampling phases, a special case of which, the 1-bit slicer structure, is analyzed in section 3.4. The result discussed in this section shows that the mutual information is preserved by randomly dispersing enough slicers in space (threshold values) and time (sampling phases). The proof of our theoretical result on ℓ_1 distance preservation is analogous to that of the JL lemma [24] which provides a theoretical basis for compressed sensing. The result also appears at

first glance to be similar to the *bit-conservation* principle articulated in [48], but the details and implications are completely different. The result in [48] considers signal reconstruction, and can be roughly paraphrased as saying that n 1-bit observations are equivalent to $n/2^k$ k-bit measurements. In contrast, our result says that n 1-bit measurements are equivalent to n infinite-precision measurements in terms of guaranteeing the feasibility of reliable data recovery in the low-noise regime (albeit with a smaller error exponent). The last section of the chapter 3.5 presents an algorithm for choosing the thresholds that approximately minimizes the bit error rate of the maximum likelihood equalizer.

3.1 Related Work

It is known that Nyquist sampling, even for strictly band-limited inputs, is not optimal for finite precision measurements. For example, Gilbert [33], Shamai [74] and Koch [44] have shown that the capacity of bandlimited systems with 1-bit measurements increases as we sample faster than the Nyquist rate. A related result is discussed by Kumar et al [48]. The effect of heavily quantized measurements on communication systems design and performance has received significant attention recently. For non-dispersive channels, the effect of coarse quantization has been studied for the ideal AWGN channel [76], carrier-asynchronous systems [77, 83],

and fading channels [59]. Reference [21] discusses channel estimation with coarsely quantized samples.

A number of recent papers [92, 13, 63, 64] consider the problem of equalization with low-precision analog front ends, and propose methods for designing ADC quantizer levels. However, the emphasis in all of these papers remains on designing multiple slicer thresholds for a given sample, rather than dispersing slicers over time as we allow. Moreover, none of these focus on designing the front end to optimize the minimum BER (based on MAP decoding) as we do. Reference [92] considers the problem of designing non-contiguous quantizers for maximizing the mutual information between i.i.d. inputs and quantized outputs. Mutual information quickly saturates with SNR, and is therefore not a good measure to optimize for the uncoded or lightly coded systems typical at high speeds. Moreover, non-contiguous quantization, if implemented by parallel comparators, does not fully utilize the available number of slicers. References [13, 63, 64] also optimize BER as we do, but they restrict attention to simpler processing (based on a linear transmit filter and DFE rather than the optimal BCJR algorithm employed here), hence their performance degrades quickly for heavy quantization and heavy precursor ISI. Our use of optimal nonlinear decoding enables significant reduction in the number of slicers while avoiding error floors: for instance, with an FR4 channel similar to the one used in [63], the BER that we obtain using only 5 slicers (equivalent to using a $\log_2(6)-$bit ADC) is much smaller than what is re-

ported there using a 3-bit ADC (7 slicers). Of course, the potential power savings in the analog front end from reducing the number of slicers must be balanced against the more complex digital backend. Such detailed tradeoffs are beyond our present scope, but as noted in the conclusions, are an important topic for future work.

3.2 System Model

We focus on uncoded transmission of binary symbols $\mathbf{b} = \{b_i\}$, with b_i chosen independently and equiprobably from $\{-1, +1\}$, at rate $1/T_s$ over a real baseband dispersive channel. The continuous time received signal at the input of the A/D conversion block is given by

$$x_c(t) = \sum_{i=-\infty}^{\infty} b_i h(t - iT_s) + w_c(t) \tag{3.1}$$

where $h(t) = (h_{TX} * h_c * h_{RX})(t)$ is the *effective* channel impulse response obtained by convolving the transmit filter $h_{TX}(t)$, the physical channel $h_c(t)$, and the receive filter $h_{RX}(t)$. Assuming white noise $n(t)$ with PSD σ^2 at the input to the receive filter, the noise $w_c(t) = (n * h_{RX})(t)$ at the input to the A/D block is zero mean Gaussian with autocorrelation function

$$R_{w_c}(\tau) = \sigma^2 \int h_{RX}(t) h_{RX}(t - \tau) dt \tag{3.2}$$

Input to quantizer: Let $x(k) = x(s_k)$ denote the continuous-valued discrete time samples obtained by sampling at times $\{s_k\}$. For Nyquist sampling at rate $1/T_s$, we set $s_k = (k + \tau)T_s$, where $\tau \in [0, 1)$ is the sampling phase (suppressed in subsequent notation for simplicity of exposition). We assume that the receive filter is square root Nyquist (e.g. square root raised cosine) at rate $1/T_s$, so that the noise samples $w_c(kT_s)$ are uncorrelated. However, sampling irregularly, or faster than $1/T_s$, both of which we allow, yields correlated noise samples.

Quantizer: We denote by $q(x; \mathbf{T})$ the output of a quantizer mapping a real-valued sample x to $N + 1$ values using thresholds $\mathbf{T} = \{t_1, ..., t_N\}$. For a classical n-bit quantizer, we have $N = 2^n - 1$. For a uniform quantizer over the range $[-R, R]$, we have

$$t_i = R\left(-1 + i\frac{2}{N+1}\right),\ i = 1, ..., N \tag{3.3}$$

Our goal here is to explore more flexible designs, in terms of choice of both N and $\mathbf{T}$.

In this paper, we consider three different scenarios:

1) T-spaced equalization (TSE): We consider regularly spaced samples at rate $1/T_s$, and we use a fixed quantizer for all samples. The effective discrete time channel is denoted by $\mathbf{h} = [h(0), h(T_s), .., h((L-1)T_s)]^T = [h_1, h_2, .., h_L]^T$, where L is the channel memory. We note that

$$x(k) = \langle \mathbf{h}, \mathbf{b}_k^{k-L+1} \rangle + w(k) \tag{3.4}$$

where $\mathbf{b}_k^{k-L+1} = (b_k, b_{k-1}, ..., b_{k-L+1})^T$ denotes the set of bits affecting the kth sample, and $w(k)$ are i.i.d. $N(0, \sigma^2||h_{RX}||^2)$. We assume that the same quantizer $\mathbf{T}$ is used for all samples, so that the quantized samples are given by

$$x_q(k) = q\left(x(k); \mathbf{T}\right) \tag{3.5}$$

The key question in this setting is how the performance depends on $\mathbf{T}$, where we allow channel-dependent choices of $\mathbf{T}$.

2) Fractionally spaced equalization (FSE): We consider samples spaced by $T_s/2$ (the typical choice for FSE), which yields two parallel symbol rate observations, which can be modeled as two parallel discrete time channels $\mathbf{h}_1$ and $\mathbf{h}_2$ operating on the same symbol stream:

$$x_i(k) = \langle \mathbf{h}_i, \mathbf{b}_k^{k-L+1} \rangle + w_i(k) \ , \quad i = 1, 2 \tag{3.6}$$

where L is the larger of the memory of the two parallel channels. The noise streams $w_i(k)$ are each white, but are correlated with each other. The correlations can be computed based on the autocorrelation function (3.2) of the continuous-time noise w_c. We also allow the quantizers for the two streams to differ, with thresholds $\mathbf{T}_1$ and $\mathbf{T}_2$, so that the two-dimensional quantized observation at time k is given by $x_q(k) = [q(x_1(k); \mathbf{T}_1), q(x_2(k); \mathbf{T}_2)]^T$.

3) General space-time equalization: Here we allow the sampling times $\{s_k\}$ to be arbitrary, and also allow the quantizer $\mathbf{T}_k$ for each sample to vary.

Thus, our goal is to understand how to rethink equalizer design in the classical settings of scenarios 1 and 2 when we have severe quantization constraints. In considering scenario 3, we try to provide a theoretical perspective on how flexible quantizer design can be, in terms of choice of sampling times and quantizers. In particular, we focus on high rate fractionally spaced sampling with randomly chosen and scalar $\mathbf{T}_k$, corresponding to one-bit quantization with time-varying thresholds.

We assume that the discrete time channels corresponding to the sampling points are known (e.g., see [21] and Chapter 6 in [91] for approaches for channel estimation with low-precision quantization). We employ the BCJR [6] or the Viterbi MLSE algorithm [31] to evaluate various quantizer designs (for completeness, a quick review of how these apply to our setting is provided in the appendix A.5). For irregular or faster than Nyquist sampling, the noise samples at the quantizer input are correlated, but we ignore these in running the BCJR or MLSE algorithm, which means that the performance in these settings could potentially be improved further by accounting for these correlations. However, accounting for such correlations in severely quantized observations is difficult, and we do not expect the gains to be significant at the high SNRs (typical for high-speed wireline links) considered here.

Example channels: We use three channels as running examples (see Figures 3.1(a), 3.1(a), 3.1(a)) throughout the paper. Channel A models a 20 inch FR4

backplane channel at 10GHz [63], and has discrete time channel impulse response (CIR) $\mathbf{h}_{A,0} = [.1, .25, .16, .08, .04]$ (maximum phase, as is typical for backplane channels). Channel B, taken from [69], is mixed phase with CIR $\mathbf{h}_{B,0} = [.23, .46, .69, .46, .23]$. For simulations with irregular or faster than Nyquist sampling, the continuous channel impulse waveform is required. We generate it using interpolation with a raised cosine waveform with roll-off factor 0.5. This may be interpreted as using matched square root raised cosine (SRRC) pulses for the transmit and receive filters with physical channel impulse response $h_c(t) = \sum_{i=1}^{L} h_i \delta(t-i)$ (setting $T_s = 1$ without loss of generality). Channel C is generated by SRRC transmit and receive pulses as above, with physical channel $h_c(t) = .2\delta(t-1) + .3\delta(t-1.85) + .15\delta(t-2.55) + .25\delta(t-3.35) + .05\delta(t-4.6)$. This gives a channel with a broader peak (formed from the merging of two peaks) than the other two. The impulse responses ($h(t)$) of the 3 channels are shown in the subfigures 3.1(a), 3.1(a), 3.1(a). The notation $\mathbf{h}_{A,\tau},\ 0 \leq \tau < 1$ is used to denote the CIR obtained by sampling at the sampling phase τ (i.e., the sampling times are at $(k+\tau)T_s$). For instance $\mathbf{h}_{C,1/2} = [-.03, .24, .3, .22, .03, .01]$.

3.3 Nyquist Sampled Uniform ADC

We first consider the standard setting of Nyquist sampling with uniform ADC with N thresholds as in (3.3), and ask how small N can be for a given chan-

nel while avoiding an error floor (i.e., error-free reception at infinite SNR)? An analytical characterization is intractable, but it is possible to evaluate $N_{\min}$ numerically by fixing $\sigma^2 = 0$, and increasing N until the information rate reaches its maximum value (for binary signaling) of one. The information rate can be evaluated via Monte Carlo methods using BCJR as described in [3]. However, it is interesting to explore whether there are analytical insights to be obtained by examining the channel coefficients. Intuitively, we expect that a channel with a strong dominant tap should have a lower value of $N_{\min}$, compared to a channel where the taps are comparable. The placement of the dominant tap should also have a significant effect. We make these intuitions concrete via the lemma stated next, which provides easily computable bounds for $N_{\min}$ when all the channel taps have the same sign (which is often a good approximation for backplane channels, for example). The proof of the lemma, given in the appendix A.6, is based on bounds on information rate derived by Zeitler [92].

Before stating the lemma, we note that the symmetric information rate is invariant under time reversal and scaling (under fixed SNR) of the channel. The scaling result is standard, and the time reversal result follows because the same output is generated by feeding a time reversed bit stream (which is another valid i.i.d. input) to the time reversed channel. Naturally, the bounds in the lemma also exhibit these invariances. Define $\mathbf{g} = \frac{\mathbf{h}}{\|\mathbf{h}\|_1}$ as a normalized version of $\mathbf{h}$ with unit ℓ_1 norm, and set $\tilde{\mathbf{g}}$ as the time-reversed version of $\mathbf{g}$, so that $\tilde{g}_i = g_{L-i+1}$.

$i = 1, ..., L$. Define

$$N_l = \left\lceil \frac{1}{\max_i(g_i)} - 1 \right\rceil \tag{3.7}$$

$$N_u = \min\left(\{\lceil u_i \rceil, 2 \le i \le L-1\}, \{\lceil v_i \rceil, 2 \le i \le L-1\}, \left\lceil \frac{1}{g_1} - 1 \right\rceil, \left\lceil \frac{1}{\tilde{g}_1} - 1 \right\rceil \right) \tag{3.8}$$

where

$$u_i = \frac{1}{(g_i - \sum_{j=1}^{i-1} g_j)_+} - 1 \quad ; \quad v_i = \frac{1}{(\tilde{g}_i - \sum_{j=1}^{i-1} \tilde{g}_j)_+} - 1$$

where $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ if $x \le 0$. Thus, we allow u_i, v_i to take the value $+\infty$, but the value of N_u is guaranteed to be finite because of the last two terms in the minimum. It is also easy to see that $v_{L-i+1} = \frac{1}{(g_i - \sum_{j=i+1}^{L} g_j)_+} - 1$.

Lemma 3. *The minimum number of levels for avoiding an error floor is bounded as follows:*

$$N_l \le N_{min} \le N_u$$

The lower and upper bounds capture the effect of the strength and the location of the dominant tap, respectively. An examination of the expression (3.8) for N_u shows that, if we can permute a given set of channel coefficients, maximum or minimum phase channels (most of the energy in ending or beginning taps) will generally have smaller $N_{\min}$ compared to mixed phase channels (most of the energy in the taps in the middle). Table 3.1 lists the values of $N_{\min}$ (computed numerically) for a few different channels along with the lower and upper bounds. We find that for a fixed channel, varying the sampling phase may slightly change

$N_{\min}$. However, as we show next, the shape of the BER curve and the performance at moderate SNRs may be far more sensitive to the sampling phase.

$\mathbf{h}$	N_l	N_u	$N_{\min}$
$\mathbf{h}_{B,0} = [.23\ .46\ .69\ .46\ .23]$	2	8	5
$[.46\ .69\ .46\ .23\ .23]$	2	4	2
$[.69\ .46\ .46\ .23\ .23]$	2	2	2
$\mathbf{h}_{B,1/4} = [.04\ .29\ .54\ .67\ .39\ .16]$	3	8	5
$\mathbf{h}_{B,1/2} = [.09\ .34\ .61\ .61\ .34\ .09]$	3	8	6
$\mathbf{h}_{A,0} = [.1\ .25\ .16\ .08\ .04]$	2	4	3
$\mathbf{h}_{C,0} = [.05\ .33\ .26\ .11\ .02]$	2	2	2

Table 3.1: Minimum number of thresholds required to decode with no error at high SNR. Also listed are the lower and upper bounds computed using Lemma 3.

For suboptimal linear equalization with unquantized samples, it is well known [34, 69] that fractionally spaced equalizers (FSE) are superior to symbol-spaced equalizers, providing robustness to sampling phase and avoiding error floors due to residual interference. However, when optimal BCJR or MLSE equalization is employed, the difference is not as drastic, but FSE is still more insensitive to sampling phase, which is attractive because hardware-based control of sampling phase is not always feasible. We would like to investigate if similar trends hold with severe quantization, with a quick exploration in this section followed by more detailed theory and algorithms in later sections. In order to have a fair comparison between TSE and FSE, we take the number of slicers used in a TSE and disperse them across different sampling phases to obtain a *space-time* architecture.

As an example, we plot in Fig. 3.1(c) the BER over channel B with TSE (unquantized and uniform ADC with 7 slicers) for sampling phases 0, 0.25 and 0.5. In the unquantized setting, there is a small degradation in performance (~ 1 dB at 10^{-5}) at sampling phase 0.5. However, the degradation with quantization is much larger, even though there is no error floor (see the $\mathbf{h}_{B,1/2}$ entry in Table 3.1). Even for channels with similar dynamic ranges, the performance of TSE/uniform-ADC with a fixed set of thresholds can show significant sensitivity to sampling phase. As a quick remedy, we try spreading the *same* set of slicers across time, as shown in Fig. 3.1(b). Changing the sampling phase now corresponds to shifting the whole space-time slicer structure. We see that now the performance (the BER curves in gray) is much less sensitive to the phase, although there is still some degradation for one of the sampling phases. This was a specific configuration, obtained without any design, which demonstrated the potential of space-time slicers. However, there are numerous ways in which the slicers can be spread across time, hence it of interest to develop automated procedures for arriving at good designs. It is also natural to ask the question as to whether there is any fundamental disadvantage to spreading slicers across time.

In the next section, we show that even randomly distributed slicers spread across time suffice to avoid error floors as long as the number of slicers is large enough, showing that there are no fundamental limitations imposed on the design space. Of course, the number of slicers predicted by this theoretical result is much

larger than what is required when the space-time architecture is optimized for a particular channel, and we consider this problem in Section 3.5.

3.4 One-bit Measurements with Random Thresholds

In this section, we consider the special case of 1-bit measurements spread over time. Without loss of generality, consider reliable demodulation of bit b_0. We restrict attention to measurements in the interval $[0, LT_s]$ affected by this bit. This choice of observation interval is sensible but arbitrary, and our approach applies to other choices as well. The measurements in this interval are also affected by $L-1$ "past" ISI bits $(b_{-L+1}, ..., b_{-1})$ and $L-1$ "future" ISI bits $(b_1, ..., b_{L-1})$. Denote the noiseless received waveform in this interval by $s(t)$, suppressing the dependence on the desired bit b_i and the ISI bits from the notation. Without loss of generality, we normalize $h(t)$ so that $s(t)$ lies in $[-1, 1]$. The main result in this section can be paraphrased as follows: for sufficiently many 1-bit measurements uniformly spaced in time but with thresholds chosen randomly over $[-1, 1]$, it is possible (at high SNR) to reliably distinguish between $b_0 = +1$ and $b_0 = -1$, as long as it is possible to do so with unquantized measurements.

Information rate: Let $\mathbf{x}_i^j$ denote the vector of samples (these may or may not be quantized) obtained during the interval $[iT_s, jT_s]$. For symbol spaced sampling, the length of $\mathbf{x}_i^j$ is $j-i+1$ (the length for general space-time slicers depends on the specific pattern of sampling times used). The information rate between the transmitted bits and the received samples is given by

$$\begin{aligned} I(\mathbf{b};\mathbf{x}) &= \lim_{N\to\infty}\frac{1}{N} I(\mathbf{b}_1^N;\mathbf{x}_1^N) \\ &= \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} I(b_i;\mathbf{x}_i^N|b_{i-L+1}^{i-1}) \\ &\geq \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} I(b_i;\mathbf{x}_i^{i+f}|b_{i-L+1}^{i-1}) \end{aligned} \tag{3.9}$$

Inequality (3.9), derived in [92], states that the information rate is lower bounded by the average (over the past bits) mutual information between the current bit and the measurements over the next few symbols (f), conditioned on the past bits. Numerical results in [92] show that this lower bound becomes a fairly tight approximation for $f = L$ future symbols.

Let $\mathbf{x}_0^L$ denote the vector of continuous-valued samples obtained by sampling $s(t)$ uniformly, n times, over the observation interval. Fixing the past ISI bits, we partition the noiseless waveforms corresponding to all possible realizations of the future bits into two sets, each of cardinality 2^{L-1}, corresponding to the two possible values of the "tagged bit" b_0: $\mathcal{S}_{-1} = \{s(t) \text{ s.t. } b_0 = -1\}$ and $\mathcal{S}_{+1} = \{s(t) \text{ s.t. } b_0 = +1\}$. Denote by $\mathcal{X}_{-1}$ and $\mathcal{X}_{+1}$ the corresponding sampled vectors $\mathbf{x}_0^L$. The absence of error floors can be proved by setting the noise level to zero and

checking whether the lower bound (3.9) on the information rate equals one. This happens as long as the set of observations generated by the two different values of the desired bit are mutually exclusive: $\mathcal{X}_{-1} \cap \mathcal{X}_{+1} = \emptyset$. Note that this property always holds for unquantized measurements, as long as at least one sample is obtained in the first symbol period ($[0, T_s]$) and the corresponding CIR value $h(0) \neq 0$. This follows from the fact that, since the past bits are fixed, and future ISI bits do not affect the waveform in the interval $[0, T_s]$, $b_0 = -1$ and $b_0 = +1$ result in different samples in the first entry of $\mathbf{x}_0^L$. This result is also discussed in [90], where the author considers symbol spaced samples and shows that the lower bound (and hence the information rate) goes to one as SNR increases as long as the first element of the discrete time CIR is nonzero. In general, such guarantees cannot be provided for quantized measurements. However, we show that as long as n is *large*, using randomized thresholds for one-bit quantization results in similar behavior.

In general (at any SNR), the performance depends on the amount of overlap/separability between the sets $\mathcal{X}_{-1}$ and $\mathcal{X}_{+1}$. For the purpose of our proof, we employ the *normalized* ℓ_1 distance between each pair of elements $\mathbf{x}_{-1} \in \mathcal{X}_{-1}$, $\mathbf{x}_{+1} \in \mathcal{X}_{+1}$, defined as follows:

$$\|\mathbf{x}_{-1} - \mathbf{x}_{+1}\|_1 = \sum_{i=1}^{n} \Delta \left| s_{-1}(i\Delta) - s_{+1}(i\Delta) \right| \tag{3.10}$$

where $s_{-1}(t)$ and $s_{+1}(t)$ are the corresponding continuous time waveforms from sets $\mathcal{S}_{-1}$ and $\mathcal{S}_{+1}$ respectively and Δ is the sampling interval (for uniform sampling as assumed in this section, $n\Delta = LT_s$). The scale factor Δ is included for the normalized ℓ_1 norm $\|\mathbf{x}_{-1} - \mathbf{x}_{+1}\|_1$ to approximate the continuous time ℓ_1 norm $\|s_{-1} - s_{+1}\|_1$ as n gets large. We define the minimum normalized ℓ_1 distance between the two sets as follows:

$$d = \min_{\mathbf{x}_{-1} \in \mathcal{X}_{-1}; \mathbf{x}_{+1} \in \mathcal{X}_{+1}} \|\mathbf{x}_{-1} - \mathbf{x}_{+1}\|_1 \tag{3.11}$$

For unquantized observations, as noted earlier, $\mathcal{X}_{-1} \cap \mathcal{X}_{+1} = \emptyset$, and hence $d > 0$.

Let us now consider what happens when we pass the unquantized sampled vector $\mathbf{x}$ through a series of one-bit quantizers, with the ith sample compared to threshold t_i. The vector of thresholds is denoted as $\mathbf{T} = [t_1, t_2, ..., t_n]^T$, and defines a quantization function q as follows:

$$q(\mathbf{x}) = (2\Delta)\mathbf{y} \;\; ; \quad y(i) = \begin{cases} 1 \;\text{ if } x(i) \geq t_i \\ \\ 0 \;\text{ if } x(i) < t_i \end{cases} \quad i = 1, ..., n \tag{3.12}$$

The following theorem states that, with a sufficient number of samples n, quantized with random thresholds, the quantization function $q(\cdot)$ approximately preserves the ℓ_1 norm of the unquantized differences $\|\mathbf{x}_{-1} - \mathbf{x}_{+1}\|_1$. This result bears some similarity to the JL lemma in which random projections preserve the norm for embeddings to lower dimension subspaces [1].

Theorem 2. *If each entry of the threshold array* $\mathbf{T}$ *is picked uniformly and independently from* $[-1, 1]$*, then for any constants* $\epsilon, \beta, \delta \geq 0$*, with probability at least* $1 - \delta$ *, for all* $\mathbf{x}_{-1} \in \mathcal{X}_{-1}$; $\mathbf{x}_{+1} \in \mathcal{X}_{+1}$ *we have*

$$(1 - \epsilon) \left\| \mathbf{x}_{-1} - \mathbf{x}_{+1} \right\|_1 \leq \left\| q(\mathbf{x}_{-1}) - q(\mathbf{x}_{+1}) \right\|_1 \leq (1 + \epsilon) \left\| \mathbf{x}_{-1} - \mathbf{x}_{+1} \right\|_1 \tag{3.13}$$

for

$$n \geq \frac{4T_s}{d\epsilon^2} \left(\log 2 \cdot (2L^2 + L) + L \log \delta^{-1} \right) \tag{3.14}$$

where d *is the minimum* ℓ_1 *distance defined in (3.11).*

Proof. Consider a particular pair of sampled measurements $\mathbf{x}_{-1} \in \mathcal{X}_{-1}$; $\mathbf{x}_{+1} \in \mathcal{X}_{+1}$ (corresponding to $s_{-1}(t) \in \mathcal{S}_{-1}$; $s_{+1}(t) \in \mathcal{S}_{+1}$). Define $\mathbf{z} = |q(\mathbf{x}_{-1}) - q(\mathbf{x}_{+1})|$, so that $z(i) = 2\Delta$ if t_i lies between (and hence can distinguish between) $s_{+1}(i\Delta)$ and $s_{-1}(i\Delta)$, and $z(i) = 0$ otherwise. Since t_i is uniformly picked from $[-1, 1]$, $z(i)$ is a (scaled version of a) Bernoulli random variable with parameter $p_i = \frac{1}{2} |s_{-1}(i\Delta) - s_{+1}(i\Delta)|$ and mean $2\Delta p_i$. Thus, from (3.10)

$$E\left(\left\| \mathbf{z} \right\|_1 \right) = E\left(\sum_{i=1}^{n} z(i) \right) = 2\Delta \sum_i \frac{|s_{-1}(i\Delta) - s_{+1}(i\Delta)|}{2} = \left\| \mathbf{x}_{-1} - \mathbf{x}_{+1} \right\|_1 \tag{3.15}$$

so that the quantization function $q(\cdot)$ preserves the norms of the differences in expectation. It remains to prove a concentration result using a Chernoff bound to show that the probability of deviation from the expectation goes to zero for large enough n. Given that the $z(i)$ are independent scaled Bernoulli random variables, derivation of the Chernoff bound is a straightforward exercise and we state the

final result, omitting the details. To simplify notation, we use the shorthand $\mu = \|\mathbf{x}_{-1} - \mathbf{x}_{+1}\|_1$ in the following.

$$\Pr\left(\|\mathbf{z}\|_1 > (1+\epsilon)\mu\right) \leq e^{-\frac{\mu}{2\Delta}((1+\epsilon)\log(1+\epsilon)-\epsilon)} \leq e^{-\frac{\mu n\epsilon^2}{4LT_s}} \tag{3.16}$$

where we have substituted $\Delta = \frac{LT_s}{n}$ and used $\log(1+\epsilon) \geq \epsilon$ (for $\epsilon \geq 0$) to obtain the last inequality. Proceeding along similar lines, we obtain an analogous bound for the probability of deviation below the expectation: $\Pr\left(\|\mathbf{z}\|_1 < (1-\epsilon)\mu\right) \leq e^{-\frac{\mu n\epsilon^2}{4LT_s}}$. Combining with (3.16) yields

$$\Pr\left(\|\mathbf{z}\|_1 < (1-\epsilon)\mu \text{ or } \|\mathbf{z}\|_1 > (1+\epsilon)\mu\right) \leq 2e^{-\frac{\mu n\epsilon^2}{4LT_s}} \leq 2e^{-\frac{dn\epsilon^2}{4LT_s}} \tag{3.17}$$

where the last inequality follows from the definition of d in (3.11). There are 2^{L+1} pairs of distances given the past bits (i.e. $|\mathcal{X}_{-1}| = |\mathcal{X}_{+1}| = 2^L$), and varying the L past bits, $|\mathcal{X}_{-1}| = |\mathcal{X}_{+1}| = 2^L$, and taking the union bound over all possible pairs $\mathbf{x}_{-1} \in \mathcal{X}_{-1}$; $\mathbf{x}_{+1} \in \mathcal{X}_{+1}$, we obtain

$$\Pr\left(\|\mathbf{z}\|_1 \leq (1-\epsilon)\mu \text{ or } \|\mathbf{z}\|_1 \geq (1+\epsilon)\mu\right) \leq 2^{2L} \cdot 2e^{-\frac{dn\epsilon^2}{4LT_s}} \leq \delta \tag{3.18}$$

which can be bounded as tightly as desired (3.18) by decreasing δ and ensuring that n meets the condition (3.14). □

Remarks: While we have considered uniform sampling for simplicity, this is not required for the theorem to hold. Using the continuity of the CIR, any non-uniform sampling strategy that provides sufficient density of samples to capture

the separation of $s_{-1}(t)$ and $s_{+1}(t)$ in the regions where the waveforms are apart suffices. The independence of the choice of thresholds is crucial for the concentration result.

Simulations: Due to the looseness of the union bound used to prove the theorem, picking n based on the theorem is excessively conservative. We now show via simulations that moderate values of n suffice to provide good equalization performance. Our choice of space-time slicers differs from the set-up of the theorem in two respects:

(1) We pick the thresholds from a Gaussian distribution $N(0, 0.4)$; this performs far better for moderate values of n than the uniform distribution assumed in the computations in the theorem. This is because, while the received signal is scaled to lie in $[-1, 1]$, the density of values near zero is higher (as we vary the possible choices of future ISI bits).

(2) Instead of picking n random thresholds over the entire duration of $[0, LT_s]$ corresponding to the span of the CIR, we pick thresholds randomly over a single symbol period T_s. This corresponds to an implementation of slicers operating at the symbol rate with a fixed threshold set for each slicer. This scheme reduces the amount of independence and hence averaging (since the thresholds are now periodic with period equal to the symbol interval), but it is simpler to implement, and provides good BER performance for the channels considered here with 10-20 slicers per symbol.

Figure (3.2(b)) shows the BER curve obtained by employing 15 randomly selected 1 bit slicers for the FR4 channel. The SNR is defined as $\frac{\|\mathbf{h}\|^2}{\sigma^2}$. The BER curves vary slightly for different instances of slicer thresholds, the general behavior remains the same for a fixed number of slicers and we find that ~ 15 slicers suffice to avoid the error floor. The bit error rates are computed empirically using BCJR. Note that the BER obtained for the random slicers case is actually an upper bound of the minimum BER as the BCJR algorithm used ignores the noise correlations and hence is not optimal. As also mentioned in the appendix A.5, it is non-trivial to extend BCJR for the case with quantization and colored noise (even though each these 2 scenarios alone can be handled).

While the theoretical results of this section are a reassuring testimony to the flexibility of space-time architectures, in practice, it is often simpler to place slicers at fewer locations. In the next section, we consider optimization of slicer locations for TSE and FSE.

3.5 Optimizing slicer thresholds

In the example discussed in Section 3.3, we observed that the uniform ADC performed very poorly at the sampling phase 0.5 with channel B ($\mathbf{h}_{B,1/2}$). A closer look at the error events (at 25dB) reveals that most of the errors are caused due to poor threshold locations rather than large noise samples. Fig. 3.3(a) plots the

continuous-valued signals corresponding to the correct and incorrect bit sequences from a simulation run in which bits 1 and 2 have been incorrectly decoded. Both noiseless and noisy signals are plotted, but they are barely discernible from each other (i.e., the noise samples are small). The noiseless sequences differ significantly at 4 sample locations (locations 2, 3, 5, 6) affected by bits 1 and 2, but at all of these, the thresholds separating the two waveforms are very close to at least one of them, hence even a small deviation due to noise greatly increases the possibility of an incorrect detection. This shows that, for low-precision quantization, it is critical to choose thresholds that are compatible with the channel at hand, since "off-the-shelf" uniform ADCs may not effectively separate out the waveforms corresponding to different bit sequences. Uniform thresholds are more compatible with Channel B with a different sampling phase, $\mathbf{h}_{B,0}$, but here too, the performance can be improved by choosing channel-specific thresholds. In this section, we present a procedure for designing a non-uniform ADC with thresholds chosen based on the channel, given a constraint on the number of slicers. We first consider a TSE, and then extend the algorithm to an FSE sampled at twice the Nyquist rate. We assume that the sampling phase is beyond our control.

3.5.1 Threshold design for TSE

Ideally, we would like to choose the thresholds, $\mathbf{T} = [t_1, ..., t_M]$, to minimize the *minimum* BER attained by MAP/BCJR decoding. However, this cost function is analytically intractable, hence we consider the union bound for MLSE performance and truncate it to a few dominant terms, targeting a high SNR regime. We use as our cost function an upper bound of this truncated sum, which can be computed easily for quantized observations.

The MLSE bit error probability, P_e, can be upper bounded using the union bound, which in its general form can be stated as follows (Section 5.8.1 in [57])

$$P_e \leq P_u = \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{b},\mathbf{b}'} P_B(\mathbf{b}, \mathbf{b}') w(\mathbf{e}) 2^{-w(\mathbf{e})} \; ; \quad \text{where } \; \mathbf{b}' = \mathbf{b} + 2\mathbf{e} \tag{3.19}$$

where $\mathcal{E}$ denotes the set of error events. As defined in [57] an error event is a simple error sequence whose first nonzero entry is at a fixed time, say at index 0. The elements of $\mathbf{e}$ take values in $\{0, \pm 1\}$, and are nonzero at indices where the bit sequences $\mathbf{b}$ and $\mathbf{b}'$ differ. The number of nonzero elements in $\mathbf{e}$, or its weight, is denoted by $w(\mathbf{e})$. We denote by $P_B(\mathbf{b}, \mathbf{b}')$ the pairwise error probability for binary hypothesis testing between $\mathbf{b}$ and $\mathbf{b}'$, which are separated by the error event expressed by $\mathbf{e}$. For continuous-valued measurements, $P_B(\cdot)$ depends only on $\mathbf{e}$, which reduces the summation $\sum_{\mathbf{b},\mathbf{b}'} P_B(\mathbf{b}, \mathbf{b}')$ to a single term that can be expressed as a function of the standard normal complementary CDF (or Q function; see (5.76) in [57]). Exact evaluation of $P_B(\cdot)$ is difficult for quantized

observations, hence we bound it from above. This, together with a restriction on the set of error events, yields an approximate upper bound that serves as our cost function for threshold design using K-means.

Truncated Union Bound

While there are infinitely many error events in $\mathcal{E}$, at high SNR, it suffices to consider a small set of most likely events which dominate the summation (3.19). For continuous-valued measurements, these correspond to the most slowly decaying Q function terms, which correspond to low weight error sequences [57]. For quantized observations, it is more difficult to identify the dominant error events, but for the channels considered here, and using the uniform quantizer starting point, simulations yield the expected result: weight one and two error patterns, $\mathbf{e}_1 = \{\pm 1, 0, 0, 0, ...\}$ and $\mathbf{e}_2 = \{\pm 1, \pm 1, 0, 0, 0, ...\}$, are by far the most dominant. We therefore restrict attention to these in truncating the union bound (3.19), as follows:

$$P_u \approx P_{ut} = \sum_{\mathbf{b},\mathbf{b}' \in E_1} P_B(\mathbf{b}, \mathbf{b}') w(\mathbf{e}_1) 2^{-w(\mathbf{e}_1)} + \sum_{\mathbf{b},\mathbf{b}' \in E_2} P_B(\mathbf{b}, \mathbf{b}') w(\mathbf{e}_2) 2^{-w(\mathbf{e}_2)} \quad (3.20)$$

where $E_i = \{\mathbf{b}, \mathbf{b}' \; s.t. \; \mathbf{b}' = \mathbf{b} + 2\mathbf{e}_i\}$, $i = 1, 2$ and $w(\mathbf{e}_1) = 1$, $w(\mathbf{e}_2) = 2$. Note that $|E_1| = 2^{(L-1)} \cdot 2^{(L-1)}$. This is because the observations that depend on the bit in error, b_0, are only affected by the truncated bit sequence $\mathbf{b}_{-(L-1)}^{L-1}$. Similarly we

get $|E_2| = 2^{(L-1)} \cdot 2^{(L-1)} \cdot 2$. For a channel with $L = 6$, $|E_1| = 1024$, $|E_2| = 2048$ which gives the total terms to be summed over to be $N = |E_1| + |E_2| = 3072$.

Bounding the Pairwise Error Probability

We now wish to bound the pairwise error probabilities $P_B(\mathbf{b}, \mathbf{b}', \mathbf{T})$ for a particular set of thresholds $\mathbf{T}$. Denoting pairs of bit sequences $(\mathbf{b}, \mathbf{b}')$ by Ω for brevity, consider the corresponding noiseless unquantized signals $\mathbf{x} = \langle \mathbf{h}, \mathbf{b} \rangle$ and $\mathbf{x}' = \langle \mathbf{h}, \mathbf{b}' \rangle$. Since we are only interested in simple error sequences, $\mathbf{x}$ and $\mathbf{x}'$ differ at most in, say K, consecutive locations. That is, $x(i) = x'(i) \ \ \forall\, i \leq 0, i \geq K+1$. Note that $K = L$ for $\mathbf{b}, \mathbf{b}' \in E_1$ and $K = L + 1$ for E_2 (changing a given bit can have an effect over at most L output samples when convolved with a channel of length L). The binary hypothesis problem of choosing one of $\mathbf{b}$ and $\mathbf{b}'$ then reduces to selecting one of the two vectors, $\mathbf{X}_0$ or $\mathbf{X}_1$ given by

$$H_0 \ : \ \mathbf{X}_0 = \mathbf{x}(1:K), \ H_1 \ : \ \mathbf{X}_1 = \mathbf{x}'(1:K) \ \ ; \ \ P_B(\Omega, \mathbf{T}) = P_B(\mathbf{X}_0, \mathbf{X}_1, \mathbf{T})$$

Fig. 3.3(a) shows an example of $\mathbf{X}_0$ and $\mathbf{X}_1$ corresponding to a particular bit sequence pair in E_2. The vectors $\mathbf{X}_0$ and $\mathbf{X}_1$ are of length K, after quantization each element takes one of $M + 1$ values, as there are M thresholds. We can now obtain a simple upper bound on the pairwise error probability by considering the probability of error in separating the scalars $X_0(i)$ and $X_1(i)$. The pairwise error probability if we only use the ith component depends only on the threshold in $\mathbf{T}$

that is closest to $\frac{X_0(i)+X_1(i)}{2}$. As a function of this scalar threshold t, we obtain that

$$P_B(X_0(i), X_1(i), t) = 2^{-(2L-2)} \left(Q\left(\frac{t - X_{\min}}{\sigma}\right) + Q\left(\frac{t - X_{\max}}{\sigma}\right)\right) \tag{3.21}$$

$$\text{where } X_{\min} = \min\left(X_0(i), X_1(i)\right) \quad X_{\max} = \max\left(X_0(i), X_1(i)\right)$$

The factor of $2^{-(2L-2)}$ is included due to the prior on the truncated bit sequences. Fig. 3.3(b) plots this function for different indices $i = 1, ..., 7$ for $\mathbf{h}_{B,1/2}$. The probability of error for deciding between the hypothesis H_0 and H_1 can be upper bounded by each of the probabilities of error based on the scalar components as we vary i, hence minimizing over i provides an upper bound:

$$\begin{aligned} P_B(\mathbf{X}_0, \mathbf{X}_1, \mathbf{T}) &\leq \min_{i=1,..,K} P_B(X_0(i), X_1(i), \mathbf{T}) \\ &= \min_i \min_{t\in\mathbf{T}} P_B(X_0(i), X_1(i), t) = \min_{t\in\mathbf{T}} \min_i P_B(X_0(i), X_1(i), t) \end{aligned} \tag{3.22}$$

Defining

$$g(\Omega, t) = \min_i P_B(X_0(i), X_1(i), t) \tag{3.23}$$

we can rewrite the upper bound as

$$P_B(\mathbf{b}, \mathbf{b}', \mathbf{T}) = P_B(\Omega, \mathbf{T}) \leq \min_{t\in\{t_1,...t_M\}} g(\Omega, t) \tag{3.24}$$

Fig. 3.3(c) shows an example plot of the function $g(\Omega, t)$.

Optimization using K-means

Applying Eq. (3.24) to Eq. (3.20), we get an upper bound on the truncated union bound, which is our cost function

$$\sum_{\Omega \in E_1 \cup E_2} P_B(\Omega) w(\Omega) 2^{-w(\Omega)} \leq \sum_{n=1}^{N} \min_{t \in \mathbf{T}} g(\Omega_n, t) w(\Omega_n) 2^{-w(\Omega_n)}$$
$$= \sum_{n} \min_{t \in \mathbf{T}} f(\Omega_n, t) \tag{3.25}$$

Defining

$$f(\Omega, t) = g(\Omega, t) w(\Omega) 2^{-w(\Omega)} \tag{3.26}$$

where $w(\Omega)$ denotes the weight of the error event $\mathbf{e} = \frac{\mathbf{b}' - \mathbf{b}}{2}$ corresponding to $\Omega = (\mathbf{b}, \mathbf{b}')$.

The problem of finding the thresholds now reduces to the following minimization problem

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t \in \{t_1, .., t_M\}} f(\Omega_n, t) = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{n=1}^{N} f(\Omega_n, t_n^*) \tag{3.27}$$

We note that the above formulation is identical to the *clustering* problem where we are given N *data points* Ω_n, which are required to be grouped into M clusters to minimize the total distortion. The distortion function is specified by $f(\Omega, t)$ and the M cluster centers represent the thresholds. We can therefore apply the standard K-means [56] algorithm to obtain candidate solutions. This involves two alternating steps:

Assignment Step: At the i^{th} iteration we have the M cluster centers/thresholds

$\{t_1^i, ..., t_M^i\}$. A 'data point' Ω_n gets assigned to the threshold with index $j^* = \underset{j=1,..,M}{\text{argmin}}\, f(\Omega_n, t_j^i)$

Update Step: The j^{th} threshold gets updated as $t_j^{i+1} = \min_t \sum_{\Omega \in t_j^i} f(\Omega, t)$; where the summation is over the data points assigned to t_j^i in the previous iteration.

The functions $f(\Omega, t)$ can be easily computed, and we compute and store them for each Ω over a grid for the parameter t. This makes the minimization in the update step straightforward. We use a grid of size 200, after first normalizing the channel to limit the range of the unquantized channel output to $[-1, 1]$, and then using a grid of size .01 for t. The K-means algorithm typically converges in a small number of iterations (< 10).

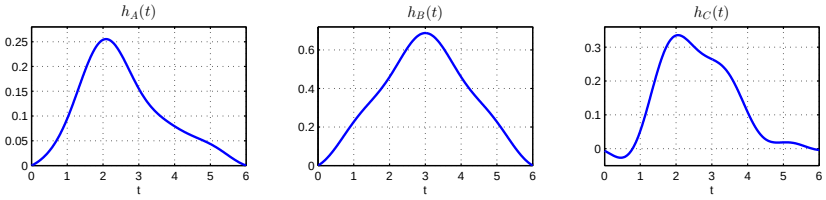
Simulations

The BER attained with the non-uniform ADCs designed using the preceding procedure is plotted in Fig. 3.4. The algorithm was run at the SNR of 20dB; higher SNR gives the same values for the thresholds. Since K-means has the tendency to get stuck in local minima, we run it several times with different random initializations and pick the best. We see a drastic improvement for $\mathbf{h}_{B,1/2}$ and a considerable gain even for $\mathbf{h}_{B,0}$. Note that, even though the cost function (plotted in gray curves) is an approximate (and rather loose) upper bound, it seems to follow a shape similar to the BER curves, and the benefit of minimizing it gets translated to the actual BER.

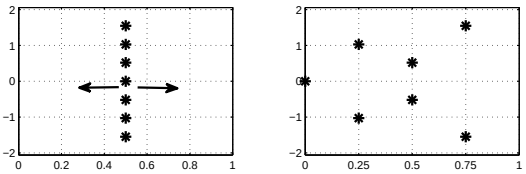
3.5.2 Threshold design for FSE $T_s/2$

Now, consider the problem of designing thresholds for slicers spread across two sampling phases separated by half a symbol period i.e. an FSE $T_s/2$ architecture. We now have two parallel discrete channels, $\mathbf{h}_1$ and $\mathbf{h}_2$. Fixing the total budget of slicers to M, suppose that we fix M_1, the number of slicers placed at the first phase (so that $M - M_1$ are placed at the second phase), then the threshold values can be computed using exactly the same machinery as earlier. We then optimize by searching over the values of M_1. The results for TSE are then a special case corresponding to $M_1 = 0$ or $M_1 = M$, and indeed, in several examples, it turns out that allocating all available slicers to one of the two sampling phases results in the lowest cost. For instance, for channel B, it is best to put all the 7 slicers at sampling phase 0 ($\mathbf{h}_{B,0}$). When we increase the number of slicers to $M = 9$ a 7-2 split configuration turns out to be the best, but it is only marginally better than having all 9 at $\mathbf{h}_{B,0}$. This makes sense, since in this case the sampling phase 0 is a good choice. For channel C, with sampling phases 0 and 0.5 and a budget of $M = 3$ slicers (2 slicers are enough for this channel to ensure no error floor, see Table 3.1), we find that the optimal configuration is a 2-1 split (Fig. 3.4). We notice a 2dB (1dB) gain compared to using a TSE non-uniform architecture at the sampling phase 0.5 (0).

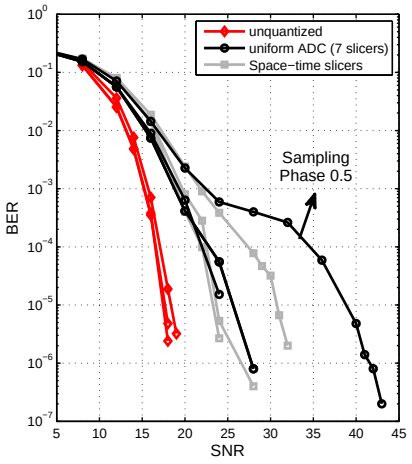
Our overall observation is that TSE with channel-optimized thresholds significantly outperforms the standard uniform ADC. The additional gain obtained by generalizing to FSE depends on the channel and the sampling phase. Of course, the trends might be quite different if BCJR decoding is replaced with lower-complexity algorithms. For example, for continuous-valued observations, FSE is much better than TSE for linear equalizers, but is typically only marginally better with BCJR decoding.



(a)

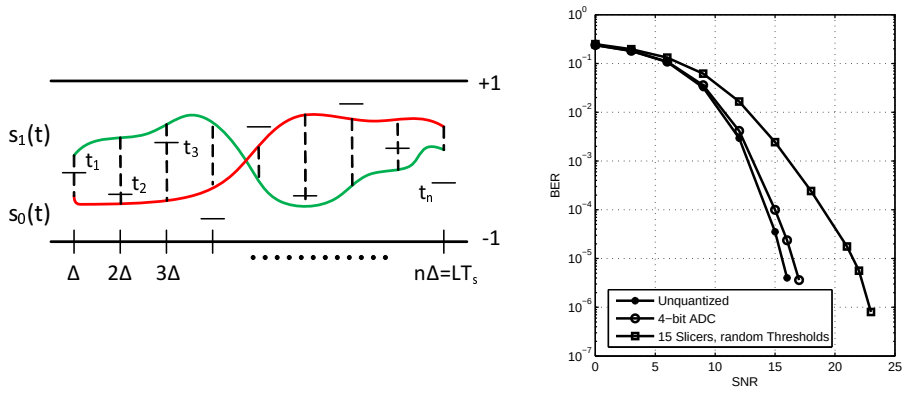


(b)



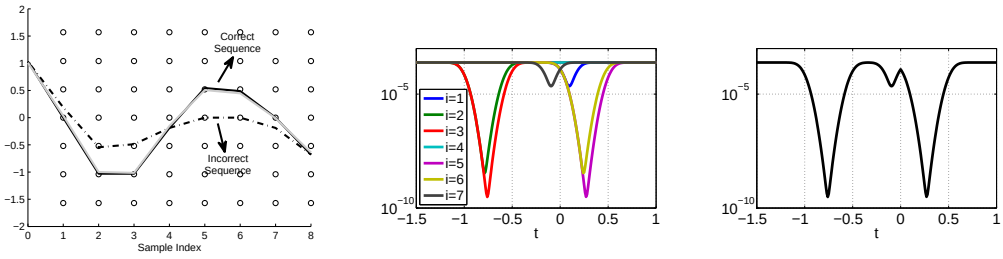
(c)

Figure 3.1: (a) Channel A,B,C (left, center, right) **(b)** TSE ADC architecture (left) and Space-time architecture (right) **(c)** Bit error rate curves for channel B corresponding to different sampling phases 0, 0.25, 0.5



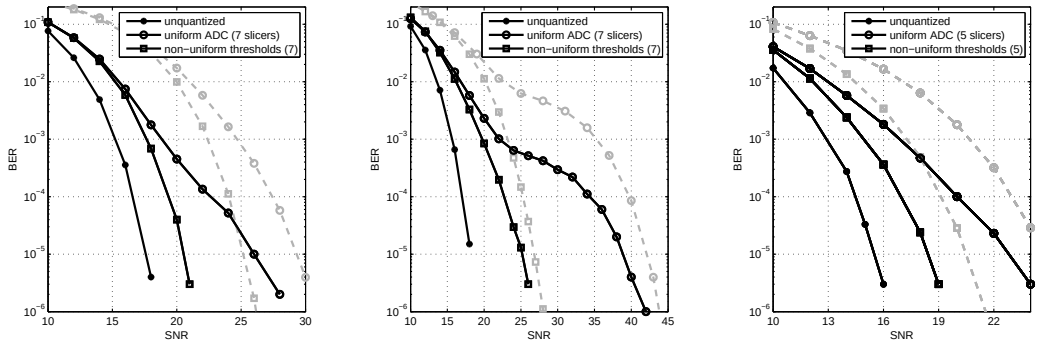
(a) (b)

Figure 3.2: **(a)** One-bit measurements with randomly varying thresholds **(b)** Bit error rates for the channel $\mathbf{h}_{A,0} = [.1, .25, .16, .08, .04]$



(a) (b) (c)

Figure 3.3: **(a)** Example of an error event with channel $\mathbf{h}_{B,1/2}$ at 25dB. Plot in gray is after noise addition. The small circles denote slicers. **(b)** Probability of error for different indices Eq. (3.21) **(c)** $g(\Omega, t)$ for the sequence shown in **(a)** at 25dB



(a) (b) (c)

Figure 3.4: The curves in gray depict the cost function (Eq. 3.25) **(a)** MLSE BER for $\mathbf{h}_{B,0} = [.23, .46, .69, .46, .23]$ **(b)** MLSE BER for $\mathbf{h}_{B,1/2} = [.1, .34, .61, .61, .34, .1]$ **(c)** MLSE BER for FR4 channel $\mathbf{h}_{A,0} = [.1, .25, .16, .08, .04]$

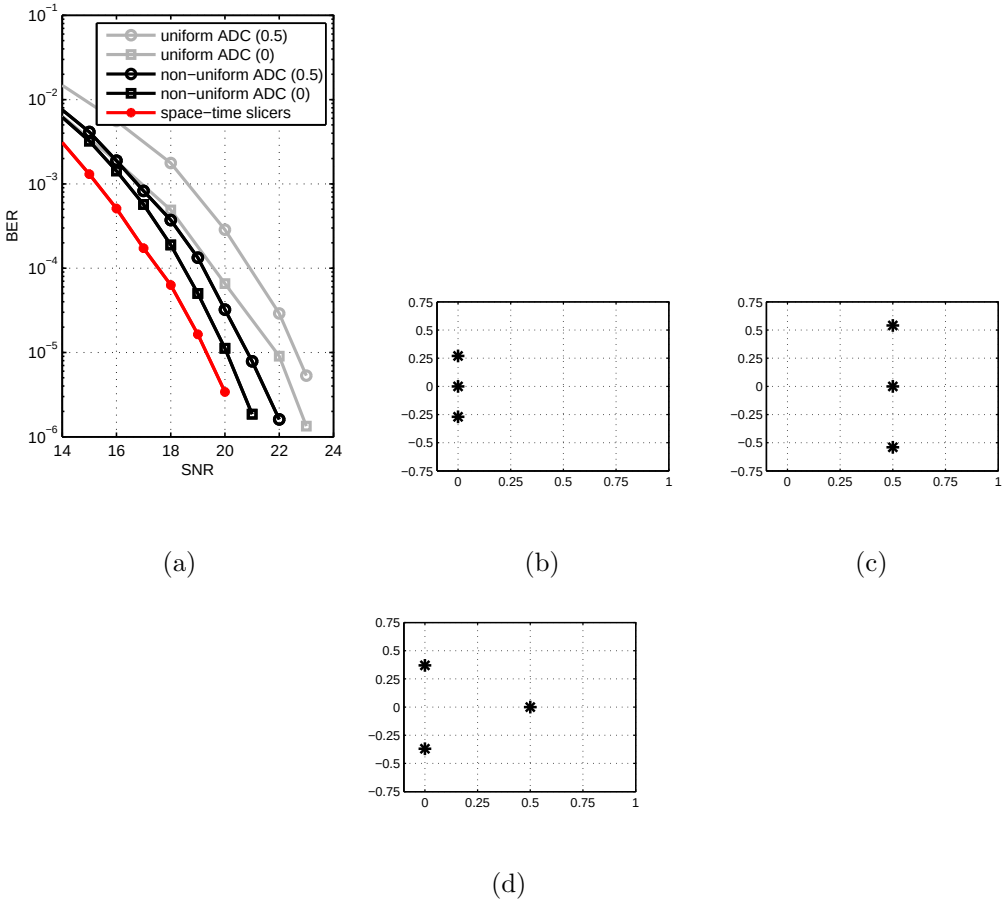


Figure 3.5: (a) Bit error rate curves for channel C with sampling phases 0 and 0.5 and a budget of 3 thresholds **(b)** Non-uniform ADC thresholds at $t = 0$ **(c)** Non-uniform ADC thresholds at $t = 0.5$ **(d)** Optimal space-time slicers configuration

Chapter 4

A Framework for Machine Vision based on Neuro-Mimetic Front End Processing and Clustering

In this chapter, we explore a front end design for machine vision that leverages the neuroscientific research about the visual pathway and combines it with unsupervised feature extraction using k-means clustering. This framework when combined with a final layer of supervised classification using support vector machine (SVM) yields excellent recognition performance with standard image databases of NORB (uniform-normalized) and MNIST.

Map of this Chapter: After a discussion of the related work in section 4.1 we present the neuro-mimetic front end design in section 4.2. We first describe the processing performed by Retinal Ganglion Cells (RGCs) and then discuss the operation of the V1 simple cells. Then we move on to the higher layer processing using clustering and show how features such as edges and combinations of edges

(corners, junctions etc) are extracted by the learned centroids. This is discussed in section 4.3. We conclude the chapter by presenting the experimental results in section 4.4.

The development of the RGC and V1 processing stages has been done by Emre Akbas, a student in the Department of Psychology and Brain Sciences, UCSB.

4.1 Related work

The relevant papers in experimental and computational neuroscience which our front end model is based on are mentioned in Section 4.2. The importance of carefully designing the pre-processing layer has been noted in the machine learning literature. It was shown in [18] that optimizing the various parameters of a single layer convolutional architecture, followed by simple non-linear clustering using k-means, results in performance even better than several deep architectures. In [15], it was found that adding a pre-processing *contrast-extraction* layer to the deep CNN architecture improves recognition performance with the NORB dataset.

There has also been recent interest in using center-surround processing in computer vision (e.g., [43]). Early modeling of simple cells was performed using Gabor functions [58], but a more neuro-plausible model was reported to yield superior edge detection performance in [5]. It is also worth mentioning "analog

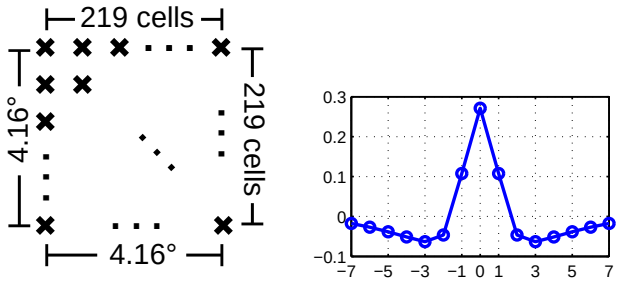
retina" hardware that uses loose neuro-inspiration to extract sparse features (with reduced power consumption) from image sensors for dynamic object tracking [25].

There are several references [71, 49, 53, 16, 89] that have employed layers of unsupervised feature extraction prior to supervised classification, an approach adopted in this work as well. Most of these papers use some form of reconstruction error combined with a sparsity constraint as the cost function for training the unsupervised layers. This differs from our use of k-means clustering to learn the weights of the unsupervised layers, an approach which is much simpler to implement computationally. A few references that have used k-means clustering for vision include [18, 17]. In these papers the clustering step is used directly on the raw images and their implementation of k-means differs significantly from ours, especially for the higher layers. We use much fewer number of centroids and get better error performance on the dataset common amongst their work and ours (NORB, [18]).

4.2 The Front End Model

Our model consists of two layers of neurons, the first corresponding to the RGC/LGN cascade, and the second to V1 simple cells, along the primate visual pathway. We model the fovea, the small part of the visual field around the center of gaze where the visual acuity is highest [86]. The fovea is responsible for tasks

that require high-resolution spatial detail such as reading. The diameter of the fovea is reported to be between 4.17° and 5.21° [45, 86]. The average of these estimates is 4.69°, and we model our "digital fovea" as a 4.16°-by-4.16° square patch having the same area as a disk with 4.69° diameter.



(a) (b)

Figure 4.1: **(a)** Cross marks show cell centers which are arranged on the vertices of a regular grid. In each row (or column) there are 219 RGCs. Each RGC cell applies a difference-of-Gaussian (DoG) filter, which defines the receptive field of the cell. Receptive fields of neighboring cells heavily overlap. **(b)** Difference of Gaussian filter along a single dimension. X-axis indices correspond to number of RGC cells.

4.2.1 RGC/LGN processing

The number of RGCs in the fovea is estimated around $120,000$ [29, 79]. Among many types of RGCs [28], midget RGCs (sustained response cells or P-cells) carry the high-acuity information [45] and comprise 80% of all the RGCs in the retina [23]. About half of these cells are ON-center-OFF-surround and the other half are OFF-center-ON-surround [86]. Based on this evidence, we create two parallel visual pathways, one for ON-center cells and the other for OFF-center cells. Each

pathway contains approximately 48000 cells. The cell centers are located on the vertices of a square regular grid (Fig. 4.1(a)). The front end also includes two mechanisms that are critical for operation over the wide dynamic range exhibited by natural stimuli: local luminance gain control (LGC) and contrast gain control (CGC) [10, 11].

We first apply LGC as described by Carandini and Heeger [11]. Denoting by $\boldsymbol{x}$ the input image, the luminance normalized image $\boldsymbol{c}$ is given as

$$\boldsymbol{c}_{i,j} = \frac{\boldsymbol{x}_{i,j} - \overline{\boldsymbol{x}_{i,j}}}{\overline{\boldsymbol{x}_{i,j}}} \tag{4.1}$$

where i, j denote a pixel and $\overline{\boldsymbol{x}_{i,j}}$ is a weighted average around pixel i, j,

$$\overline{\boldsymbol{x}_{i,j}} = \sum_p \sum_q \boldsymbol{w}_{p,q} \boldsymbol{x}_{i-p,j-q}. \tag{4.2}$$

where the weights $\boldsymbol{w}$ are given by the Gaussian surround filter suggested in [8], normalized to sum to 1.

Computation of center-surround contrast is classically modeled using the difference-of-Gaussian (DoG) model [73, 27, 20] consisting of two components, center and surround, each of which is a 2D Gaussian function. We set the parameters of the center and surround Gaussian filters based on the values given for the macaque retina [20] (details in the appendix A.7). Taking the difference between these gives a DoG filter (Fig. 4.1(b)) whose radius covers about 7 cell centers along a row. Convolving the luminance-normalized image with the DoG filter, the ON-center cell responses are governed by the positive part of the output, and the OFF-center

by the negative part (Fig. 4.2). We apply CGC as follows. The output (spike rate) of a cell whose center is at i, j set to [11] is given by

$$\boldsymbol{r}_{i,j} = \frac{\sum_p \sum_q \boldsymbol{v}_{p,q} \boldsymbol{c}_{i-p,j-q}}{\beta + \sqrt{\sum_p \sum_q \boldsymbol{w}_{p,q} \boldsymbol{c}^2_{i-p,j-q}}} \tag{4.3}$$

where $\boldsymbol{v}$ are the difference-of-Gaussian weights. The square-root term in the denominator, called the local contrast, is the weighted root mean square of the luminance normalized intensity values within the whole receptive field. The area defined by $\boldsymbol{w}$ is called the suppressive field. The parameter β has been fit to neural data by Bonin *et al* [8], but this value is for cells outside of the fovea, and hence is not directly usable for our model. We therefore choose a value of β (= 0.1) so that the cells in our model qualitatively match various effects (step change in luminance, step change in contrast, size and contrast tuning) described by Bonin et al. [8].

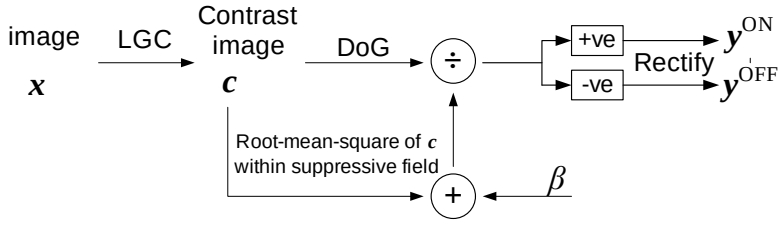


Figure 4.2: RGC processing pipeline for a single RGC cell

Finally, the non-negative spike rate of the cell is obtained via a rectification non-linearity [10]:

$$\boldsymbol{y}_{i,j}^{ON} = \max(0, \boldsymbol{r}_{i,j} - T_{RGC}) \tag{4.4}$$

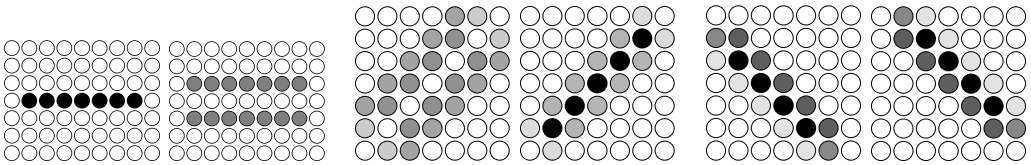
$$\boldsymbol{y}_{i,j}^{OFF} = \max(0, -\boldsymbol{r}_{i,j} - T_{RGC}) \tag{4.5}$$

where T_{RGC} is the rectification threshold: we set $T_{RGC} = 0$, which corresponds to simply splitting responses into positive and negative components. Such "polarity splitting" has been used in several machine learning algorithms (e.g., [16]), and preserves more information than absolute value rectification. The overall flow of RGC processing is illustrated for a single cell in Fig. 4.2.

While both luminance and contrast gain control are thought to start at the retina, lateral geniculate nucleus (LGN) cells strengthen CGC [10]. For this reason, we refer to this layer as the RGC/LGN layer.

4.2.2 V1 simple cells

The V1 layer consists of two populations of neurons: simple cells and complex cells. While there is a strong consensus on the computation performed by V1 simple cells – they extract oriented edges – the picture is less clear about the complex and hypercomplex cells. Hubel and Wiesel [38] suggest that some complex cells are implementing an OR-like (or MAX-like) operation, while there are recent



(a) (b) (c)

Figure 4.3: A simple cell sums the output of RGC/LGN cells according to its incoming weights, these are represented here in terms of the colors of the circles. The darker the color of a cell, the more weight it has. Transparent cells have zero weight. Weights of each simple cell are normalized to sum to 1. For each simple cell, the weight connections to the midget-ON and OFF RGCs are shown on the left and right sides respectively. (a) orientation 0°, OFF-ON-OFF type connection to midget ON. (b) orientation 45°, ON-OFF-ON type connection to midget ON. (c) orientation 135°, ON-OFF type connection to midget ON.

studies [30, 41] which suggest significant computational diversity among complex cells. We therefore only include simple cells in our front end model.

Simple cells have incoming connections from the RGC/LGN layer. We create simple cell receptive fields based on the size (0.25°x0.25° [39]) and the shapes ([38, Fig. 2]) reported by Hubel and Wiesel for foveal simple cells. While this seminal work that we draw upon is almost five decades old, there are only a few other studies [35, 67] of primate foveal V1 cells, and the detail they present are insufficient to implement a complete simple cell population. Other models for parafoveal neurons ($5° - 6°$ degrees off-center) [58, 72] are similar in concept, but different in size, from the Hubel/Wiesel foveal model.

There are a total of 48 different types of simple cells in our model. There are 8 orientations, starting at 0° (horizontal edge) and increasing in increments

of 22.5°. For each orientation, there are 6 kinds of simple cells: two ON-OFF-ON, two OFF-ON-OFF and one each of the type ON-OFF and OFF-ON. To understand the differences between these types we illustrate three different simple cells in Figure 4.3. Each simple cell is connected to both midget-ON and midget-OFF RGCs (and thus obtains information from both the positive and negative parts of the DoG outputs), and its shape is characterized by the set of nonzero weights. Each simple cell has a receptive field size of 7x7 RGC cells, but depending on its shape and type (equivalently, the set of nonzero weights), the number of incoming connections vary from 14 to 39 RGC/LGN cells. The unnormalized output of the simple cell at location (i, j) with orientation θ and shape γ is the sum of its afferent inputs:

$$\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma} = \sum_{p,q} \boldsymbol{\ell}^{ON}_{p,q} \boldsymbol{y}^{ON}_{i-p,j-q} + \sum_{p,q} \boldsymbol{\ell}^{OFF}_{p,q} \boldsymbol{y}^{OFF}_{i-p,j-q} \tag{4.6}$$

where $\boldsymbol{\ell}$ are the weights (e.g. as shown in Fig. 4.3) of the incoming RGC/LGN cells. The superscripts ON and OFF refer to the midget-ON and midget-OFF pathways. Similar to the contrast gain control occurring at the previous layer, cortical neurons are also locally normalized [10]. Carandini and Heeger [11] propose several variations of the normalization model. (Normalization has also been successfully used in bio-inspired methods [40, 55, 68].) In our experiments, we use a normalization similar to (4.3) used at the RGC/LGN layer: local demeaning, followed by a divisive normalization with root-mean-square of nearby outputs, a

measure of local contrast.

$$\boldsymbol{s}^{(norm)}_{i,j,\theta,\gamma} = \frac{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma} - \overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}}{\max\left(\epsilon, \sqrt{\sum_{p,q,\theta,\gamma} \boldsymbol{w}_{p,q}\left(\boldsymbol{s}^{(raw)}_{i-p,j-q,\theta,\gamma} - \overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}\right)^2}\right)} \tag{4.7}$$

where the summation is taken over the suppressive field $\boldsymbol{w}$ across orientations and shapes, $\overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}$ is a weighted local average (using $\boldsymbol{w}$ as weights) of unnormalized V1 outputs for θ, γ around i, j, and ϵ is a small positive constant to prevent division by zero (we set it to 0.001). Finally, the normalized simple cell output is rectified to yield a non-negative spike rate

$$\boldsymbol{s}_{i,j,\theta,\gamma} = \max(0, \boldsymbol{s}^{(norm)}_{i,j,\theta,\gamma}). \tag{4.8}$$

4.2.3 Viewing distance and foveal image resolution

Our model has a 4.16°x4.16° visual field. For a typical viewing distance of 50 cm, this field corresponds to a 3.6x3.6 cm^2 patch. The smaller the viewing distance, the smaller the image patch covered by the fovea, and vice versa.

In order to implement our model digitally, one has to assume a size for the foveal image. One possibility is to assume that the resolution is limited by the number of photo receptor cells. In the fovea, there are almost exclusively cone photo receptors. Based on the cone density at the fovea [45], there are about $3 \cdot 10^5$ cells which would mean a 550x550 pixel resolution. Considering the typical viewing distance example given above, 3.6 cm would correspond to 550 pixels

resulting in a 152.8 pixels/cm density which is too high compared to pixel densities of available displays ($\approx 40 - 100$ pixels/cm). To close this gap, one either has to increase the resolution of the input image or scale down the foveal image size. We choose the latter for simplicity and assume that the foveal image resolution is equivalent to the RGC resolution, i.e. 219x219 (61 pixel/cm). That is, at every pixel there is a RGC cell center. With these settings, the radius of the center component for a midget-ON cell is 1.27 pixels and the radius of the surround component is 5.53 pixels. An image from the MNIST dataset [50], which is 28x28 pixels, would be seen by 28x28 midget-ON RGC cells (and by the same number of midget-OFF cells); and would cover approximately 0.5x0.5 cm^2 area on a display with 60 pixel/cm viewed at 50 cm distance. An image from the NORB dataset [51] (96x96 pixels) would cover 1.6x1.6 cm^2.

While one RGC center per pixel is a sensible design choice, it is possible to tune the viewing distance parameter in our model. For example, larger values would increase the number of RGC centers per pixel, and require sub-pixel computations. We do not experiment with the viewing distance parameter in this paper, but note that it could be of interest, for example, when comparing the performance of our model with human performance on the same task in psychophysics experiments.

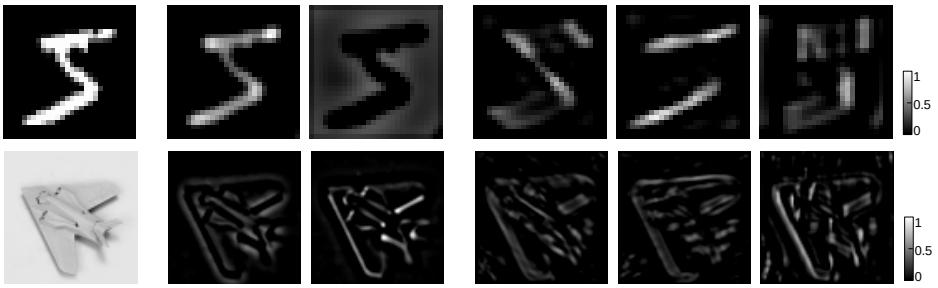


Figure 4.4: Sample RGC and V1 output. First row is for an image from MNIST, the second row is for NORB. The first column has the original images. The second and third columns are midget-ON and midget-OFF outputs. The last three columns are outputs of 4 simple cells at different orientations. The midget-ON and OFF responses seem to light up the *relevant* regions containing activity.

4.3 Higher Layer Processing

Our front end implements 48 types of simple cells centered around each input pixel, so that our front end outputs, for each pixel, an f-dimensional feature ($f = 48$ for monocular images as in MNIST, and $f = 96$ for NORB, which consists of a set of binocular images). We employ k-means clustering on this f-dimensional data, as a natural proxy for complex cell modeling. Thus, the feature map for an $N \times N$ image at our front end output is $N \times N \times f$, while that after the first layer of clustering is $N \times N \times k$ (to be cut down by pooling). We consider two implementations: a single layer of clustering followed by pooling and supervised classification, or two layers of clustering (and pooling), and then using a concatenation of layers 1 and 2 features for classification. The second

implementation is consistent with visual models for higher layers, which predict connections from both layers V1 and V2 into V3.

4.3.1 Layer 1 of clustering

We have denoted by $\boldsymbol{s}_{i,j}$ the activations of simple cells centered at a particular spatial location i, j. To represent a response in general, we drop spatial coordinates from the notation and denote the activations by $\boldsymbol{a} = \boldsymbol{s}_{i,j}$, an f-dimensional vector. We implement spherical k-means clustering [93] using an *inner product* similarity metric $\boldsymbol{a}^T\boldsymbol{c}$, where $\boldsymbol{c}$ denotes a cluster center. This is equivalent to clustering using a standard Euclidean distance metric with a unit norm constraint on the cluster centers. In our implementation, we use the online clustering algorithm in [93], which has the advantage of being less sensitive to initialization. We speed up the algorithm by using mini-batches instead of iterating over single data points.

Note that computation of the inner product of a data vector with a cluster center is identical to weighted summations in classical neural networks, hence we may interpret each cluster center as a neuron. The subsequent nonlinearity, however, is different from the sigmoidal nonlinearity in standard neural networks. As described shortly, we use soft assignments, which may be interpreted in terms of local competition between the neurons.

In addition to using the *standard* inner product as a similarity metric, we also consider a modified version that takes into account the correlations in simple cell activations. Given the weights connecting LGNs to the simple cells, represented by $L = [\boldsymbol{\ell}_1, .., \boldsymbol{\ell}_{48}]$, we compute the 48×48 correlation matrix as $C_l = L^T L$ and use the metric $\boldsymbol{a}^T C_l^{-1} \boldsymbol{c}$ or $(C_l^{-\frac{1}{2}} \boldsymbol{a})^T (C_l^{-\frac{1}{2}} \boldsymbol{c})$ for k-means. This can be viewed as doing *whitening* before clustering. For NORB, where $f = 96$ and simple cell outputs are concatenation of the left and right channels, we do not have prior information about the correlations among the two channels, and model them as independent.

Given the centroids, the soft activations are evaluated by

$$f\left([\boldsymbol{a}^T C^{-1} \boldsymbol{c_1}, ..., \boldsymbol{a}^T C^{-1} \boldsymbol{c_{K_1}}]^T\right)$$

where $C = C_l$ or $C = I$ and K_1 are the number of layer 1 cluster centers learned. We use the *soft threshold* as the encoding function, i.e. $f(x) := \max(0, x - T)$. It is known that neurons fire only when active above a certain threshold hence rectification for the non-linearity is a natural choice. For choosing the value of T we take the simple approach of setting it to maintain a certain level of *sparsity* on average. For instance, we can choose T for 80% sparsity (i.e., only 20% of the neurons have non-zero activations on average). This design rule gives us a direct and intuitive handle on controlling the level of sparsity, as opposed to the regularization parameter generally used in cost functions containing a sparsity term [89, 71, 49]. The resulting design conforms to the intuition that neural

activity on average is expected to be low. The final activation vector generated is of length $K_1 + 1$: the last coordinate is set to a non-zero value when all the K_1 responses corresponding to the centroids turn out to be zero after thresholding. This typically corresponds to patches with no or negligible activity.

Features extracted by layer 1, as expected, correspond to different kinds of edges, blobs etc. In order to visualize a centroid, we back project its receptive fields to the raw image level and plot the patches closest to it. Since layer 1 centroids are connected directly to the simple cell responses, their receptive field size is same as that of the simple cells: 7×7 RGCs or pixels in the image domain. In Figure 4.5, for the MNIST dataset, we show visualizations for four centroids.

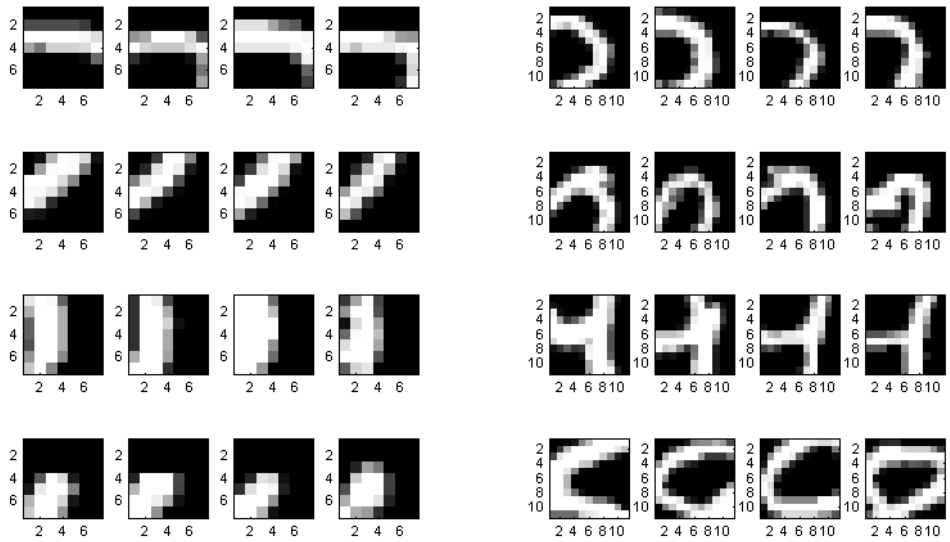


Figure 4.5: Left side: layer 1 centroids. Right side: layer 2 centroids. Each row plots patches closest to that centroid.

4.3.2 Layer 2 of clustering

The idea with the second layer of clustering is to extract more complex features: combination of simple edges like corners, L-junctions etc. The expansion of receptive field size or zooming out is achieved via local spatial pooling and concatenation. Max-pooling over a small neighborhood also results in local translation invariance. Pooling is generally followed by subsampling, hence it results in reducing the resolution of the feature maps. Denoting the max-pooled activations at the spatial location i, j by $\boldsymbol{b}_{i,j}$, these are then concatenated over a 2×2 neighborhood to generate $4(K_1+1)$-dimensional input for the second layer of clustering, given by $[\boldsymbol{b}_{i,j}; \boldsymbol{b}_{i,j+1}; \boldsymbol{b}_{i+1,j}, \boldsymbol{b}_{i+1,j+1}]$. These activations now correspond to larger patches of the raw image. Clustering is performed using the similarity metric:

$$\sum_{ii=0}^{1} \sum_{jj=0}^{1} \frac{\boldsymbol{b}_{i+ii,j+jj}^{T} \boldsymbol{w}_{ii,jj}}{\|\boldsymbol{b}_{i+ii,j+jj}\| \, \|\boldsymbol{w}_{ii,jj}\|} \tag{4.9}$$

where a second layer centroid is represented by $\boldsymbol{c}^{(2)} = [\boldsymbol{w}_{0,0}; \boldsymbol{w}_{0,1}; \boldsymbol{w}_{1,0}; \boldsymbol{w}_{1,1}]$. Using this metric can be interpreted as individually comparing the four quadrants of the larger patch and computing an averaged matching score. This is expected to group together shapes with similar arrangement of edges, with the metric interpreted as *stitching* the edges together. The soft assignment encoding function is as in layer 1.

In order to understand how pooling, subsampling and concatenation enlarges the receptive field size, consider a simple 1D example. Suppose that layer 1

centroids/neurons have a receptive field of size 7 (i.e. a neuron at location i in layer 1 gets its inputs from layer 0 neurons indexed at $[i-3, i+3]$). Now suppose we do pooling and subsampling, both by a factor of 2. For pooling by a factor of 2, layer 1 neurons at i and $i+1$ are pooled together to generate a layer 2 neuron, so that the effective receptive field (with respect to layer 0) for this new neuron is 8: $[i-3, i+4]$. Since we subsample by a factor of 2, the neighbor of this new neuron is based on pooling layer 1 neurons at $i+2$ and $i+3$. Now, when these two neighboring layer 2 neurons are concatenated, their resulting receptive field size is 10 in terms of layer 0: $[i-3, i+4] + [i-1, i+6] = [i-3, i+6]$.

In our experiments with MNIST, after layer 1 clustering, we perform 2×2 pooling, subsampling by 3 and 2×2 concatenation, followed by layer 2 clustering: hence layer 2 centroids correspond to 11×11 sized raw image patches. Figure 4.5 shows visualizations of a few layer 2 centroids using these 11×11 patches.

4.4 Experiments

In this section, we evaluate our model on two standard image classification benchmarks, MNIST [50] and NORB [51]. The only free parameter for the neuro-mimetic front end is the viewing distance which we set to 50cm. For the higher layers we experiment with number of centers $K_1 = 200$ or 600 for layer 1, and $K_1 = 200$ and $K_2 = 600$ when employing both layers 1 and 2. Thresholds are

	Sparsity level= 80%			Sparsity level= 95%		
	(Layer 1) (K_1=200)	(Layer 1) (K_1=600)	(Layer 1+2) (K_1=200) (K_2=600)	(Layer 1) (K_1=200)	(Layer 1) (K_1=600)	(Layer 1+2) (K_1=200) (K_2=600)
MNIST	0.73	0.72	**0.66**	0.78	0.78	0.68
NORB	3.96	3.71	2.94	2.58	**2.52**	2.90

Table 4.1: MNIST and NORB results: error rate (%) on the test set.

chosen to keep the sparsity level at either 80% or 95% for both layers. We use non-linear SVM with the radial basis function (RBF) kernel [12] for supervised classification. RBF SVM has two parameters: the cost parameter, which we fix to 100 as that seemed to be a robust choice in our experiments, and the scale parameter for the kernel, γ, which is set via a grid search using cross-validation on a subset of the training set. Several references have used data augmentation (via affine distortions) to enlarge the training set in order to boost classification performance, but we do not employ it here.

MNIST: MNIST consists of 28×28 images of handwritten digits. The dataset contains 60K training and 10K testing images. The front end produces feature maps of size $28 \times 28 \times (K_1+1)$. If only layer 1 is used for classification, spatial average pooling over a 4×4 grid followed by concatenation provides a 1D vector of dimension $4^2 \times (K_1+1)$ to be fed into the RBF SVM. When layer 2 is also used, we fix $K_1 = 200$ and max-pool layer 1 activations over a 2×2 local neighborhood. This is subsampled by a factor of 3, and edges are cropped, giving feature maps of size $8 \times 8 \times 201$. We then concatenate neighboring responses over a 2×2 grid,

which leads to a feature map size $7 \times 7 \times 804$. The 804-length feature vectors are clustered in layer 2 using $K_2 = 600$ centroids, producing feature maps of size $7 \times 7 \times 601$. Finally, layer 2 features for classification are generated by pooling over a 3×3 grid, coarser than layer 1 since the activations now correspond to larger image patches (11×11, layer 1 centroids represent 7×7). Concatenating layer 1 and 2 features results in a total of $4^2 \cdot 201 + 3^2 \cdot 601 = 8625$ features per image, which is comparable to the length of layer 1 features alone with $K_1 = 600$ (9616). For MNIST, we find that using *whitening* prior to layer 1 clustering, as discussed in subsection 4.3.1, yields better results, hence we only report those error rates (Table 4.1). We see that the best error rate 0.66% is achieved using both layer 1 and 2 features and a sparsity level of 80%. Increasing the sparsity appears to degrade the performance, especially when using just layer 1. The state of the art on MNIST (without distortions) is 0.39% [52], which is achieved using a purely supervised net. Although the error rate we get is higher than that, it is comparable to the rates reported by several other references, 0.64% [71], 0.82% [53], 0.59% [49], that use a combination of unsupervised and supervised learning.

NORB: We use the normalized-uniform variant [51] of the NORB dataset. Each of the training and test sets have 24300 binocular images of 5 classes of toys placed on a uniform background. Each monocular image is 96×96. We pre-process the images by cropping 8 pixels from all sides reducing the image size to 80×80, in order to speed up the processing of the dataset. This cropping discards some

of the uniform background and it does not affect the final performance. The operations are mostly identical to those for MNIST, hence we only mention the differences here. Due to the larger image sizes, the final spatial pooling before classification is done over a finer grid: 5×5 for layer 1 and 4×4 for layer 2. Another difference is that max pooling is performed over 3×3 neighborhoods after layer 1 clustering, the layer 2 centroids represent 12×12 patches. As with MNIST, the size of concatenated layer 1 and 2 features is comparable to layer 1 features with $K_1 = 600$ centers. For NORB, unlike MNIST, omitting whitening at layer 1 clustering results in better performance. We believe this could be due to the inability of the correlation matrix (C_l) to model correlations between the left and right channels. The current best result on the normalized-uniform NORB, to the best of our knowledge, is the one reported in [15] and is 2.87% without data augmentation and 2.53% with translation distortions. The best result obtained by us of 2.52% thus improves upon the state of the art; it is even marginally better than the previous best with distortions, even though we do not employ distortions.

Discussion: While these classification results are encouraging, there are several unanswered questions. Design choices such as whitening and sparsity level appear to be dataset dependent for optimizing the classification performance. It might be the case that the optimal sparsity levels depend on the noisiness of the dataset or hierarchy of the layer. The impact of whitening before clustering is also not clear.

In [18], whitening using the empirical covariance matrix has been found to improve performance, but it did not improve our results. We generally expect higher layer features to improve recognition performance, but in the NORB experiments with 95% sparsity, we were surprised to find performance degrading with the inclusion of layer 2 features. Clearly, our understanding of how best to combine information generated from different layers is far from complete. While our focus has been on feature design via clustering, it is important to explore multiple options for the supervised classification layered on top of it (e.g., comparing multilayer neural nets to the nonlinear SVM used here).

Chapter 5

Conclusions

In this dissertation we have provided three distinct examples and in each case recognized the importance of carefully designing a front end that is more suited to the requirements of the system. Although the designs are specific to the problems at hand, such endeavors provide useful guidelines and insight into system design. For instance, our work in this dissertation has highlighted the importance of using Bayesian principles as means of efficiently extracting information and the promise of neural inspiration in scenarios when precise modeling is not possible.

We conclude by summarizing our contributions and pointing out some future directions in each of the three case studies taken up in this work.

5.1 Blind Phase/frequency Synchronization

The framework for ADC-constrained receiver design illustrated in this work has two core components:

(a) digitally controlled analog preprocessing: this provides the *dither* required for estimation with coarsely quantized observations in the acquisition step, and the *correction* required for coherent demodulation in the tracking step;

(b) Bayesian algorithms for estimation and feedback generation: this involves propagation of posterior probabilities in a manner that accounts for the quantization nonlinearity while probabilistically modeling unknown data and channel parameters. These posteriors are used to compute both the feedback for the analog preprocessor and the ultimate estimates of interest.

Our numerical results indicate that such architectures provide a promising approach for DSP-centric designs that exploit Moore's law despite the ADC bottleneck encountered at high communication bandwidths.

The success of a Bayesian approach for the simplified model considered here motivates future research on a comprehensive framework for receiver design subject to severe quantization constraints. Although we have also analyzed channel equalization separately, it is of interest to jointly address the problem of carrier synchronization with timing synchronization and dispersion. It is important to consider extensions to larger amplitude/phase constellations. It is also of interest to develop a deeper theoretical understanding of fundamental performance limits under quantization constraints.

5.2 Slicer Architectures for Analog-to-Information Conversion in Channel Equalizers

We show in this work that, for communication over dispersive channels with low-precision quantization, there is significant scope for improving on generic ADC designs by focusing on *analog-to-information* in which slicer thresholds are chosen so as to effectively separate out waveforms corresponding to different bit sequences. In addition to choosing slicer thresholds as a function of the channel, spreading slicers out over time can improve upon Nyquist rate sampling. We have shown that there are no error floors when we take this concept to an extreme, with one-bit comparators dispersed uniformly over time. We have also provided an algorithm for choosing slicer thresholds for TSE and FSE (sampled at twice the symbol rate), which yields designs that significantly outperform the standard Nyquist-sampled uniform ADC. In summary, our results show that, despite the increased dynamic range due to channel dispersion, it is possible to significantly reduce the number of slicers (and hence the power consumption of the analog front end), while recovering the information encoded in the received signal.

There are a number of open issues for future research. We have used the BCJR algorithm to benchmark performance in this paper, but it is of interest to reduce the complexity of the digital equalizer, and to design the analog-to-information converter accordingly. In particular, it is of interest to explore if we can improve

performance relative to prior attempts along these lines based on linear transmit filters and DFE [13, 63, 64], possibly using a judicious combination of the simplicity of the DFE with the more comprehensive exploration of sequence space obtained using more complex MLSE/BCJR algorithms. Extending our framework to larger constellations is also an important topic for future work. Another interesting issue relates to noise correlations for fractionally spaced sampling, which we have accounted for in simulations but ignored in our current designs. The effect of such correlations is expected to be minor at high SNR, but it is certainly of interest to explore, especially in low SNR settings, if it is possible to develop elegant approaches for handling correlations for quantized observations, both in terms of analysis and design. While our focus here has been on communication-theoretic considerations, from a circuit designer's point of view, it is essential to trade off complexity and power consumption of the analog front end and the associated digital backend (e.g., using fewer slicers in the analog front end may require complex digital processing). Further effort is also needed to account for, and design around, effects such as slicer metastability (i.e., uncertainty in digital output when the sample value is close to the threshold) and errors in sampling phases. Finally, while our starting point here is the flash ADC architecture, it is of interest to explore whether the concept of analog-to-information conversion can be effectively applied to obtain more power-efficient designs starting from the pipelined or successive approximation register architectures, for example.

5.3 Neuro-Mimetic Front End Processing and Clustering

We have shown that an architecture based on neuro-mimetic front end processing and clustering offers a promising approach for "universal" feature extraction for machine vision. Layering a generic (but powerful) supervised classifier on top is shown to provide performance close to, or exceeding, the state of the art for two well known image databases. Key advantages of our approach are its simplicity, the small number of tunable parameters, and the ability to easily interpret the features being extracted at each layer.

We view this work as a first step towards bridging the gap between computational neuroscience and machine learning: machine vision algorithms are often neuro-inspired but rarely implement computations that strictly follow neuroscientific findings, while psychophysical models that try to follow physiological visual processing more closely are typically applied to restricted problems with artificial inputs[32, 87]. The results in this paper show that leveraging neuroscientific findings more carefully can pay off in terms of machine vision performance.

An obvious disadvantage of our approach, from the point of view of machine learning, is that we are limited in our front end design by the state of knowledge in neuroscience, instead of learning purely from data. For example, our model here is

restricted here to grayscale images, because more work is needed to put together the available experimental evidence regarding color processing at the RGC/LGN layers, which exhibits features such as red-green and blue-yellow opponency [28]. However, we believe that this additional effort in faithful modeling is well worth it because of the potential benefits from leveraging evolution. In particular, we would like to extend our approach (both in terms of neuro-mimetic front end and layered clustering) to other kinds of data, such as audio and video.

A fundamental challenge, as we aim to build additional layers using clustering, is to develop a quantitative understanding of whether all of the relevant information is being captured by our feature extractor. The only available metric at present to evaluate the efficacy of our architecture is classification performance after inserting a supervised layer, which is sensitive to the dataset and perhaps to the complexity of the supervised layer. An important open question, therefore, is if there are alternative metrics for evaluating the quality of information being extracted by unsupervised learning models such as ours. Of course, in parallel with this line of inquiry, we would like to continue optimizing our architecture so that it meets or surpasses classification performance on standard databases.

Appendices

Appendix A

A.1 Derivation of the Phase Distribution

The expression for the unquantized phase is given by Eq. (2.2) as follows

$$u = \arg\left(e^{jp\frac{\pi}{4}}e^{j\beta} + w\right) = arg(v)$$

p is uniformly distributed over $\{1, 3, 5, 7\}$ and w is complex WGN with variance σ^2 per dimension. Let us denote coordinates of the random complex variable v by $X = \mathrm{Re}(v)$ and $Y = \mathrm{Im}(v)$. Conditioned on p, $X \sim \mathcal{N}\left(\cos(p\frac{\pi}{4} + \beta), \sigma^2\right)$ and $Y \sim \mathcal{N}\left(\sin(p\frac{\pi}{4} + \beta), \sigma^2\right)$. To evaluate the distribution of the argument of v, we transform from Cartesian to polar coordinates ($x = r\cos(\alpha)$, $y = r\sin(\alpha)$) which gives the following joint distribution

$$\begin{aligned} f(r,\alpha) &= r^2 f(x,y) \\ f(r,\alpha) &= \frac{r^2}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}\left(x-\cos(p\frac{\pi}{4}+\beta)\right)^2} e^{-\frac{1}{2\sigma^2}\left(y-\sin(p\frac{\pi}{4}+\beta)\right)^2} \\ f(r,\alpha) &= \frac{r}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}\left(r^2+1-2r\cos(p\frac{\pi}{4}+\beta-\alpha)\right)} \end{aligned} \tag{A.1}$$

where (A.1) follows from the independence of X and Y. We can now marginalize out r to get the distribution of u

$$f_u(a) = \int_0^\infty \frac{r}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}\left(r^2+1-2ra\right)} dr \tag{A.2}$$

$$a = \cos(p\frac{\pi}{4} + \beta - \alpha)$$

where dependence on α has being expressed through a. Integral (A.2) can be computed by observing that $f(a)$ (dropping subscript u) is the derivative of another integral $g(a)$ defined below, which in turn can be easily evaluated by completing squares in the exponent and expressing in terms of the standard Q function.

$$\begin{aligned} g(a) &= \frac{1}{2\pi}\int_0^\infty e^{-\frac{1}{2\sigma^2}\left(r^2+1-2ra\right)} dr \\ &= \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(1-a)^2}{2\sigma^2}} \left(1 - Q\left(a/\sigma\right)\right) \end{aligned}$$

$$f(a) = g'(a) = \frac{a(1-Q(a/\sigma))e^{\frac{a^2-1}{2\sigma^2}}}{\sigma\sqrt{2\pi}} + \frac{e^{-\frac{1}{2\sigma^2}}}{2\pi} \tag{A.3}$$

Averaging out p we get Eq. (2.3).

A.2 Proof of Theorem 1

Consider the Taylor series expansion of the KL divergence (Eq. 2.14) centered at ϕ_0 (note that $\phi_0 = \phi_{MAP}$ since $f(\phi) \sim \mathcal{N}(\phi_0, v^2)$)

$$D^\theta(\phi) = D^\theta(\phi_0) + (\phi - \phi_0)D'^\theta(\phi_0) + \frac{(\phi - \phi_0)^2}{2}D''^\theta(\phi_0) + \ ... \tag{A.4}$$

the superscripts $'$ and $''$ denote derivatives with respect to ϕ. Substituting this in Eq. (2.13) gives

$$IU^\theta = D^\theta(\phi_0)\int f(\phi)d\phi \ + \ D'^\theta(\phi_0)\int f(\phi)(\phi - \phi_0)d\phi + D''^\theta(\phi_0)\int f(\phi)\frac{(\phi - \phi_0)^2}{2}d\phi + \ ... \tag{A.5}$$

since $f(\phi)$ is normally distributed, this simplifies to

$$IU^\theta = D^\theta(\phi_0) + \frac{v^2}{2}D''^\theta(\phi_0) + O(v^4) \tag{A.6}$$

or

$$\lim_{v\to 0}\frac{IU^\theta}{v^2} = \lim_{v\to 0}\frac{D^\theta(\phi_0)}{v^2} + \lim_{v\to 0}\frac{1}{2}D''^\theta(\phi_0) \tag{A.7}$$

Consider the first term in the equation above

$$\frac{D^\theta(\phi_0)}{v^2} = \sum_i \frac{p^\theta_{\phi_0}(z_i)}{v^2}\log\left(\frac{p^\theta_{\phi_0}(z_i)}{\int p^\theta_\phi(z_i)f(\phi)d\phi}\right)$$

$$= \sum_i \frac{p^\theta_{\phi_0}(z_i)}{v^2}\log\left(\frac{p^\theta_{\phi_0}(z_i)}{p^\theta_{\phi_0}(z_i) + \frac{v^2}{2}h^\theta_{\phi_0}(z_i) + O(v^4)}\right) \tag{A.8}$$

$$\text{where} \ \ h^\theta_\phi(z) = \frac{\partial^2 p^\theta_\phi(z)}{\partial\phi^2}$$

where we have used the Taylor series expansion for $p^{\theta}_{\phi}(z_i)$ around ϕ_0 to get Eq. (A.8). Applying the limit $v \to 0$ using the L'Hospital's rule (and using the fact that $p^{\theta}_{\phi}(z)$ is strictly positive for any finite SNR), the expression above simplifies to

$$\begin{aligned}\lim_{v\to 0} \frac{D^{\theta}(\phi_0)}{v^2} &= \frac{-1}{2}\sum_i h^{\theta}_{\phi_0}(z_i) \\ &= \frac{-1}{2}\sum_i \frac{\partial^2 p^{\theta}_{\phi_0}(z_i)}{\partial \phi^2} = \frac{-1}{2}\frac{\partial^2}{\partial \phi^2}\left(\sum_i p^{\theta}_{\phi_0}(z_i)\right) \\ &= \frac{-1}{2}\frac{\partial^2}{\partial \phi^2}(1) = 0\end{aligned}$$

where we use the fact that $p^{\theta}_{\phi}(z)$ is the observation density and hence sums to 1. The first term in Eq. (A.7) is thus 0. For the second term, evaluating the double derivative of the KL divergence and using simple arithmetic simplifications (that we skip) gives

$$\frac{1}{2}D^{''\theta}(\phi_0) = \frac{1}{2}\sum_i h^{\theta}_{\phi_0}(z_i)\log\left(\frac{p^{\theta}_{\phi_0}(z_i)}{\int p^{\theta}_{\phi}(z_i) f(\phi) d\phi}\right) + \frac{1}{2}\sum_i \left(\frac{\partial p^{\theta}_{\phi}(z_i)}{\partial \phi}\right)^2_{\phi=\phi_0} \frac{1}{p^{\theta}_{\phi_0}(z_i)} \quad \text{(A.9)}$$

which is a summation of two terms, the second one is the fisher information evaluated at ϕ_0

$$\frac{1}{2}D^{''\theta}(\phi_0) = \frac{1}{2}T_1 + \frac{1}{2}FI^{\theta}(\phi_0) \quad \text{(A.10)}$$

Fisher information is independent of v. The proof of the theorem is complete by observing that the first terms goes to 0 as $v \to 0$. This is because the argument

of the log term approaches 1.

$$\lim_{v\to 0} \frac{p^{\theta}_{\phi_0}(z_i)}{\int p^{\theta}_{\phi}(z_i) f(\phi) d\phi} = 1 \tag{A.11}$$

This can be easily derived by using the Taylor series expansion of $p^{\theta}_{\phi}(z_i)$ around ϕ_0.

A.3 Proof of Lemma 1

In the absence of noise, it is straightforward to see that the unnormalized single step phase density, $p^{\theta}_{\phi}(z)$, is uniformly distributed in ϕ for any given value of θ and z. Moreover, its support has the same size as the bin size which is $\frac{2\pi}{M}$ (30° or 45° for $M = 12$ and $M = 8$ respectively). Starting from a uniform prior, the phase posterior after k steps is given by

$$f_k(\phi) \propto \prod_{j=1}^{k} p^{\theta_j}_{\phi}(z_j) \tag{A.12}$$

this follows from the recursive update rule given by Eq. (2.5). The first part of the lemma follows directly from the fact that the product of uniform densities is also a uniform density, with a support that is the intersection of the individual support intervals.

Since $f_k(\phi) = \frac{1}{S_k}$, its entropy is given by

$$h(k) = -\int f_k(\phi) \log(f_k(\phi)) d\phi = \log(S_k) \tag{A.13}$$

We see that the entropy of a uniform density is equal to the logarithm of the length of the support interval. Hence minimizing entropy corresponds to minimizing the support. Let us denote the support interval of $f_k(\phi)$ by $[\phi_k^1, \phi_k^2]$; $0 \leq \phi_k^1 \leq \phi_k^2$ (we can assume it to be of this particular form if we do not wrap around to force the phase to lie in the interval $[0, \frac{\pi}{2})$, something that we do in practice for a simpler implementation). Note that $\phi_k^2 - \phi_k^1 = S_k$ and $S_k \leq \frac{2\pi}{M}$. Now, conditioned on the action θ_{k+1} and the QPSK symbol $p_k \frac{\pi}{4}$; $p_k \in \{1, 3, 5, 7\}$, the net final phase in the next step, Ω_{k+1}, lies uniformly in the interval $\Omega_{k+1} \in \left[\Omega_{k+1}^1, \Omega_{k+1}^2\right] = \left[\phi_k^1 - \theta_{k+1} + p_k \frac{\pi}{4}, \phi_k^2 - \theta_{k+1} + p_k \frac{\pi}{4}\right]$. Since this interval is less than $\frac{2\pi}{M}$, the bin size, there are only two quantized phase measurements possible at $k+1$; let us denote them by indices $i-1$ and i.

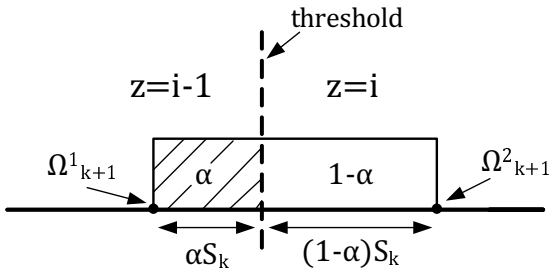


Figure A.1: Distribution of the net phase Ω_{k+1}. Dotted line denotes the phase threshold. Note that $\Omega_{k+1}^2 - \Omega_{k+1}^1 = S_k$

$$p_\phi^{\theta_k}(z_{k+1}) = \begin{cases} \alpha\ ; & z_{k+1} = i-1 \\ 1-\alpha\ ; & z_{k+1} = i \end{cases} \tag{A.14}$$

$$\alpha = Pr(\Omega_{k+1} \leq \text{threshold}) \in [0, 1]$$

The relative probabilities of getting these two measurements, denoted by $\{\alpha, 1-\alpha\}$, is determined by the action θ_{k+1} through which we can control the location of the uniform Ω_{k+1} density relative to the closest threshold. It can be easily seen that if we get the measurement $z_{k+1} = i - 1$, the uncertainty in phase will be reduced to an interval of size αS_k. This means that the conditional entropy $h(k+1|z = i-1) = \log(\alpha S_k)$. Similarly $h(k+1|z = i) = \log((1-\alpha)S_k)$. Hence the average entropy is given by

$$E[h(k+1)] = \alpha \log(\alpha S_k) + (1-\alpha)\log((1-\alpha)S_k) \tag{A.15}$$

this is minimized when $\alpha = \frac{1}{2}$. This means that irrespective of the measurement, the support of the new posterior is half of the earlier support, i.e. $S_{k+1} = \frac{S_k}{2}$. Since $S_0 = \frac{\pi}{2}$ we get an exponentially decreasing support $S_k = \frac{\pi}{2^{k+1}}$. GE strives to make $\alpha = \frac{1}{2}$ by choosing an action θ that places the net phase distribution symmetrically around one of the thresholds. This is equivalent to saying that the *expected* value of the net final phase is equal to one of the "boundaries" (phase thresholds). Note that this strategy is optimal as choosing any value of α other than $\frac{1}{2}$ results in a support size that on average is greater than half of the previous support. Also note that even though MFI is not well defined because of the flat posterior, if we choose ϕ_{MAP} as the mean of the posterior, it is same as GE since fisher information is maximized when the net phase is placed at the boundary at high SNR.

A.4 Proof of Lemma 2

The key observation to see why the lemma holds is this: it can be easily inferred from equations (2.3) and (2.4) that the set of phase offset rotations $\beta = \phi - \theta = \left\{\alpha, \frac{\pi}{4} - \alpha + k\frac{\pi}{2}\right\};\ k \in \mathbb{I};\ \forall\ \alpha$ result in identical conditional densities $P(z|\beta)$ when M=8. For fixed derotation, these different values correspond to different phase offsets. Setting $k = 0$ we can write:

$$\alpha = \phi - \theta \ \text{ and } \ \frac{\pi}{4} - \alpha = \phi' - \theta \tag{A.16}$$

$$\Rightarrow \phi' = \frac{\pi}{4} - \alpha + \theta = \frac{\pi}{4} - \phi + 2\theta \tag{A.17}$$

It suffices to consider $k = 0$ if ϕ' is wrapped around to lie in the interval $[0, \frac{\pi}{2})$.

A.5 BCJR Algorithm

The BCJR algorithm relies on a Markov structure [6], and applies directly to quantized observations with Nyquist sampling. For faster sampling, the noise correlation can still be handled by state extension if the observations are unquantized [42], but the Markov structure is destroyed by quantization. Thus, for FSE/space-time architectures, we simply ignore the noise correlations, so that the BER attained is an upper bound on the minimum possible BER.

For TSE, the state at time k is $S_k = \{b_k, b_{k-1}, ..., b_{k-L+2}\}$. From (3.4), the observation $x(k)$ is a function of S_{k-1}, S_k and the noise $w(k)$. The standard BCJR

equations for the posterior probability of the state are given by

$$p(S_k|\mathbf{x}_0^N) \propto p(S_k|\mathbf{x}_0^k)p(\mathbf{x}_{k+1}^N|S_k,\mathbf{x}_0^k) = p(S_k|\mathbf{x}_0^k)p(\mathbf{x}_{k+1}^N|S_k) = \alpha_k\beta_k \tag{A.18}$$

Forward Recursion

$$\alpha_k = p(S_k|\mathbf{x}_0^k) = \sum_{S_{k-1}} p(x_k|S_k,S_{k-1})p(S_k|S_{k-1})\alpha_{k-1} \tag{A.19}$$

Backward Recursion

$$\beta_k = p(\mathbf{x}_{k+1}^N|S_k) = \sum_{S_{k+1}} \beta_{k+1}\; p(x_{k+1}|S_k,S_{k+1})p(S_{k+1}|S_k) \tag{A.20}$$

Note that, for i.i.d. binary signaling, the only computation required is of $p(x_k|S_k,S_{k-1})$, since $p(S_k|S_{k-1}) = 0.5$. From (3.4), (3.5), the likelihood of the observation given the states is given by

Continuous Observations

$$p(x(k)|S_k,S_{k-1}) \propto \exp\left(\frac{-1}{2\sigma^2}\left\|x(k)-\mu\right\|^2\right) \tag{A.21}$$

Quantized Observations

$$p(x_q(k)|S_k,S_{k-1}) = Q\left(\frac{l-\mu}{\sigma}\right) - Q\left(\frac{u-\mu}{\sigma}\right) \;\;;\;\; l \le x(k) \le u \tag{A.22}$$

where $\mu = \langle \mathbf{h}, \mathbf{b}_k^{k-L+1} \rangle$. The quantized observation $x_q(k)$ is specified via the interval $[l,u]$. $Q(\cdot)$ denotes the standard normal Q-function. Note that $\mathbf{b}_{k-L+1}^k$ is specified completely via S_k and S_{k-1}. Note that MLSE using the Viterbi algorithm [31] can be run in similar fashion, since it also involves the same core computation of the observation likelihoods (A.22). Since we are ignoring noise correlations, the preceding approach extends directly to FSE with quantization.

A.6 Proof of Lemma 3

To prove the lemma, we utilize bounds on information rate derived by Zeitler [92], which are valid for both unquantized and quantized measurements, assuming i.i.d. bits and symbol spaced sampling (independent noise samples).

Lower Bound

$$I(\mathbf{b},\mathbf{z}) \geq \lim_{N\to\infty}\frac{1}{N}\sum_i I\left(b_i, \mathbf{z}_i^{i+L-1}|\mathbf{b}_{i-L+1}^{i-1}\right) \overset{\text{stationarity}}{=} I\left(b_i, \mathbf{z}_i^{i+L-1}|\mathbf{b}_{\text{past}}\right) \quad \text{(A.23)}$$

$$= H(b_i) - H\left(b_i|\mathbf{z}_i^{i+L-1}, \mathbf{b}_{\text{past}}\right) = 1 - H\left(b_i|\mathbf{z}_i^{i+L-1}, \mathbf{b}_{\text{past}}\right) \quad \text{(A.24)}$$

Upper Bound

$$I(\mathbf{b},\mathbf{z}) \leq \lim_{N\to\infty}\frac{1}{N}\sum_i I\left(b_i, \mathbf{z}_i^{i+L-1}|\mathbf{b}_{i-L+1}^{i-1}, \mathbf{b}_{i+1}^{i+L-1}\right)$$

$$\overset{\text{stationarity}}{=} I\left(b_i, \mathbf{z}_i^{i+L-1}|\mathbf{b}_{\text{past}}, \mathbf{b}_{\text{future}}\right) \quad \text{(A.25)}$$

$$= H(b_i) - H\left(b_i|\mathbf{z}_i^{i+L-1}, \mathbf{b}_{\text{past}}, \mathbf{b}_{\text{future}}\right)$$

$$= 1 - H\left(b_i|\mathbf{z}_i^{i+L-1}, \mathbf{b}_{\text{past}}, \mathbf{b}_{\text{future}}\right) \quad \text{(A.26)}$$

Here $\mathbf{z}$ denotes measurements at the symbol rate: $\mathbf{z} = \mathbf{x}$ (unquantized), $\mathbf{z} = \mathbf{x}_q$ (quantized). The lower bound is the average mutual information between a bit (b_i) and the set of observations it affects (which are $\mathbf{z}_i^{i+L-1}$), conditioned on the *past* bits ($\mathbf{b}_{\text{past}} = \mathbf{b}_{i-L+1}^{i-1}$). If we further condition on the *future* bits ($\mathbf{b}_{\text{future}} = \mathbf{b}_{i+1}^{i+L-1}$) we get the upper bound.

We set the noise variance to zero, and consider the normalized channel $\mathbf{g} = (g_1, ..., g_L)^T$ with $g_j \geq 0$ for all j. Setting $i = 0$ without loss of generality, let $\mathbf{y} = \mathbf{z}_0^{L-1} = \mathbf{x}_0^{L-1}$ denote the portion of the continuous-valued output containing contributions from b_0:

$$y(j) = ... + g_{j-1}b_1 + g_j b_0 + g_{j+1}b_{-1} + ... \quad \text{or} \quad \mathbf{y} = G_p \mathbf{b}_{\text{past}} + G_f \mathbf{b}_{\text{future}} + b_0 \mathbf{g} \quad \text{(A.27)}$$

where G_p and G_f are appropriately defined matrices of size $L \times (L-1)$.

In order to derive the lower bound N_l, consider the upper bound (A.26) on information rate. Let $\mathbf{y}_{+1}$ denote the value of $\mathbf{y}$ conditioned on $b_0 = +1$ and $\mathbf{y}_{-1}$ denote the corresponding value for $b_0 = -1$. Conditioned on the past and future bits, $\Delta\mathbf{y} = \mathbf{y}_{+1} - \mathbf{y}_{-1} = 2\mathbf{g}$. Since $\|\mathbf{g}\|_1 = 1$, each output sample $y(j)$ is confined to $[-1, 1]$ (since the input bits are from ± 1). For a uniform ADC with N thresholds covering this range, the size of each quantization bin is $\frac{2}{N+1}$. If the thresholds *separate* even one component of $\Delta\mathbf{y}$, we can distinguish between $b_0 = +1$ and $b_0 = -1$, and the conditional entropy term in (A.26) is zero. This happens if N is large enough that the bin size is smaller than the biggest separation, given by $\max_k 2g_k$:

$$\frac{2}{N+1} \leq \max(2\mathbf{g}) \Rightarrow N \geq \frac{1}{\max(\mathbf{g})} - 1 \quad \text{(A.28)}$$

If N is smaller than the preceding value, it is easy to see that there is at least one set of values for the past and future bits (e.g., set them all to one) for which $b_0 = +1$ and $b_0 = -1$ cannot be distinguished.

For deriving N_u, we consider the lower bound (A.24) on the information rate. Conditioned on the past bits, the possible values of the components of $\mathbf{y}_{+1}$ and $\mathbf{y}_{-1}$ are given by

$$y_{+1}(j) = .. + g_{j-2}b_2^l + g_{j-1}b_1^l + g_j + g_{j+1}b_{-1} + ..$$

$$y_{-1}(j) = .. + g_{j-2}b_2^k + g_{j-1}b_1^k - g_j + g_{j+1}b_{-1} + ..$$

where the superscripts l and k are used to denote that the future bits $b_1, b_2, ...$ need not be the same. The *minimum* value of $y_{+1}(j)$ and the *maximum* value of $y_{-1}(j)$ are given by

$$y_1^*(i) := \min_{\mathbf{b}^j_{\text{future}}} y_1(i) = -\sum_{t=1}^{i-1} g_t + g_i + g_{i+1}b_{-1} + ..$$

$$y_{-1}^*(i) := \max_{\mathbf{b}^k_{\text{future}}} y_{-1}(i) = \sum_{t=1}^{i-1} g_t - g_i + g_{i+1}b_{-1} + ..$$

We have an open eye at sample j if $y_{+1}^*(j) - y_{-1}^*(j) > 0$, which happens if $2\left(g_j - \sum_{t=1}^{j-1} g_t\right) \geq 0$. If there is a threshold between $y_{+1}^*(j)$ and $y_{-1}^*(j)$, then we can separate $b_0 = +1$ and $b_0 = -1$ irrespective of the value of the future bits. This corresponds to the following condition on N:

$$\frac{2}{N+1} \leq 2\left(g_i - \sum_{t=1}^{j-1} g_t\right) \Rightarrow N \geq \frac{1}{g_j - \sum_{t=1}^{i-1} g_t} - 1 \tag{A.29}$$

We get a set of upper bounds on N for each $j = 1, ..., L$, along with a corresponding set of bounds for the time-reversed channel. Minimizing across these gives the bound N_u stated in the lemma.

A.7 Difference of Gaussian parameters

We use the classical difference-of-Gaussians (DoG) model ([73, 27, 20]):

$$R(x,y) = K_c e^{-\frac{(x^2+y^2)}{r_c^2}} - K_s e^{-\frac{(x^2+y^2)}{r_s^2}} \tag{A.30}$$

where K_c and r_c are the contrast gain and radius of the center component, respectively, and K_s, r_s are the same for the surround component. DoG parameter values for the foveal RGCs are not directly available in published data. Croner and Kaplan [20] report

- median values of $r_c = 0.03°$ and $r_s = 0.18°$, for cells at $0° - 5°$ eccentricity (The eccentricity of a point A on the retina is the angle between the center of the fovea and A); and
- median values of $r_c = 0.05°$ and $r_s = 0.43°$ for cells at $5° - 10°$ eccentricity.

r_c, r_s increase linearly with eccentricity [20]. Hence, we fit a line to the values above (e.g. for r_c, two points on the line are $(2.5°, 0.03)$ and $(7.5°, 0.05)$ where we took $2.5°$ as the representative eccentricity for the $0° - 5°$ interval, and $7.5°$ for the $5° - 10°$). We choose $1°$ as the representative eccentricity for foveal RGCs, where the lines yield $r_c = 0.024°$ and $r_s = 0.105°$. The degree/pixel ratio for our model is $4.16°/219$ pixels $= 0.019$ degree/pixel. Therefore, $r_c = 0.024/0.019 = 1.27$ pixels and $r_s = 0.105/0.019 = 5.53$ pixels. The values of K_c and K_s are inversely proportional to the center and surround areas, respectively [20].

Bibliography

[1] D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001.

[2] E. Akbas, A. Wadhwa, M. Eckstein, and U. Madhow. A framework for machine vision based on neuro-mimetic front end processing and clustering. In *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE, 2014.

[3] D.-M. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng. Simulation-based computation of information rates for channels with memory. *Information Theory, IEEE Transactions on*, 52(8):3498–3508, 2006.

[4] G. Atia and S. Aeron. Asymptotic optimality results for controlled sequential estimation. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*.

[5] G. Azzopardi and N. Petkov. A corf computational model of a simple cell that relies on lgn input outperforms the gabor function model. *Biological Cybernetics*, 106(3):177–189, 2012.

[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *Information Theory, IEEE Transactions on*, 20(2):284–287, 1974.

[7] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. Wiley-Interscience, 2001.

[8] V. Bonin, V. Mante, and M. Carandini. The suppressive field of neurons in lateral geniculate nucleus. *The Journal of neuroscience*, 25(47):10844–56, Nov. 2005.

[9] A. G. Busetto, A. Hauser, G. Krummenacher, M. Sunnåker, S. Dimopoulos, C. S. Ong, J. Stelling, and J. M. Buhmann. Near-optimal experimental design for model selection in systems biology. *Bioinformatics*, 29(20):2625–2632, 2013.

[10] M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. a. Olshausen, J. L. Gallant, and N. C. Rust. Do we know what the early visual system does? *The Journal of Neuroscience*, 25(46):10577–97, 2005.

[11] M. Carandini and D. J. Heeger. Normalization as a canonical neural computation. *Nature reviews. Neuroscience*, 13(1):51–62, Jan. 2012.

[12] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] E.-H. Chen, R. Yousry, and C.-K. Yang. Power optimized adc-based serial link receiver. *Solid-State Circuits, IEEE Journal of*, 47(4):938–951, 2012.

[14] V. H.-C. Chen and L. Pileggi. An 8.5 mW 5GS/s 6b flash adc with dynamic offset calibration in 32nm CMOS SOI. In *VLSI Circuits (VLSIC), 2013 Symposium on*, pages C264–C265. IEEE, 2013.

[15] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. *arXiv preprint arXiv:1102.0183*, 2011.

[16] A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 921–928, 2011.

[17] A. Coates and A. Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[18] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[20] L. J. Croner and E. Kaplan. Receptive fields of P and M ganglion cells across the primate retina. *Vision Research*, 35(1):7–24, Jan. 1995.

[21] O. Dabeer and U. Madhow. Channel estimation with low-precision analog-to-digital conversion. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.

[22] O. Dabeer and E. Masry. Multivariate signal parameter estimation under dependent noise from 1-bit dithered quantized data. *Information Theory, IEEE Transactions on*, 54(4):1637–1654, 2008.

[23] D. M. Dacey and M. R. Petersen. Dendritic field size and morphology of midget and parasol ganglion cells of the human retina. *Proc. of the National Academy of Sciences (PNAS)*, 89(20):9666–70, Oct. 1992.

[24] S. Dasgupta and A. Gupta. An elementary proof of the johnson-lindenstrauss lemma. *International Computer Science Institute, Technical Report*, pages 99–006, 1999.

[25] T. Delbruck and P. Lichtsteiner. Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 845–848. IEEE, 2007.

[26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[27] C. Enroth-Cugell and J. G. Robson. The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology*, 187(3):517–52, Dec. 1966.

[28] G. D. Field and E. J. Chichilnisky. Information processing in the primate retina: circuitry and coding. *Annual review of neuroscience*, 30:1–30, Jan. 2007.

[29] S. Filipe and L. a. Alexandre. From the human visual system to the computational models of visual attention: a survey. *Artificial Intelligence Review*, Jan. 2013.

[30] I. M. Finn and D. Ferster. Computational diversity in complex cells of cat primary visual cortex. *The Journal of Neuroscience*, 27(36):9638–48, Sept. 2007.

[31] G. D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[32] W. S. Geisler. Sequential ideal-observer analysis of visual discriminations. *Psychological review*, 96(2):267, 1989.

[33] E. N. Gilbert. Increased information rate by oversampling. *Information Theory, IEEE Transactions on*, 39(6):1973–1976, 1993.

[34] R. Gitlin and S. Weinstein. Fractionally-spaced equalization: An improved digital transversal equalizer. *Bell System Technical Journal*, 60(2):275–296, 1981.

[35] M. J. Hawken and A. J. Parker. Spatial properties of neurons in the monkey striate cortex. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 231(1263):251–88, July 1987.

[36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[37] A. Host-Madsen and P. Handel. Effects of sampling and quantization on single-tone frequency estimation. *Signal Processing, IEEE Transactions on*, 48(3):650–662, 2000.

[38] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, pages 106–154, 1962.

[39] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, pages 215–243, 1968.

[40] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision (ICCV)*, pages 2146–2153, 2009.

[41] I. Kagan, M. Gur, and M. Snodderly. Modeling V1 complex cells in alert monkeys. In *CSH meeting on Computational and Systems Neuroscience (COSYNE)*, 2004.

[42] A. Kavcic and J. M. Moura. The viterbi algorithm and markov noise memory. *Information Theory, IEEE Transactions on*, 46(1):291–301, 2000.

[43] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2214–2219. IEEE, 2011.

[44] T. Koch and A. Lapidoth. Increased capacity per unit-cost by oversampling. *arXiv preprint arXiv:1008.5393*, 2010.

[45] H. Kolb, E. Fernandez, and R. Nelson, editors. *Webvision: The Organization of the Retina and Visual System*. Salt Lake City (UT): University of Utah Health Sciences Center, 1995. Available from http://www.ncbi.nlm.nih.gov/books/NBK11530/.

[46] A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. *arXiv preprint arXiv:1207.1394*, 2012.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, page 4, 2012.

[48] A. Kumar, P. Ishwar, and K. Ramchandran. High-resolution distributed sampling of bandlimited fields with low-precision sensors. *Information Theory, IEEE Transactions on*, 57(1):476–492, 2011.

[49] K. Labusch, E. Barth, and T. Martinetz. Simple method for high-performance digit recognition based on sparse coding. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 19(11):1985–9, Nov. 2008.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[51] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–97. IEEE, 2004.

[52] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *arXiv preprint arXiv:1409.5185*, 2014.

[53] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[54] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[55] S. Lyu and E. Simoncelli. Nonlinear image representation using divisive normalization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[56] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[57] U. Madhow. *Fundamentals of digital communication*, volume 518. Cambridge University Press New York, USA, 2008.

[58] S. Marčelja. Mathematical description of the responses of simple cortical cells. *JOSA*, 70(11):1297–1300, 1980.

[59] G. Middleton and A. Sabharwal. On the impact of finite receiver resolution in fading channels. In *Allerton Conf. on Communication, Control and Computing*, 2006.

[60] B. Murmann. ADC performance survey 1997-2013. `http://www.stanford.edu/~murmann/adcsurvey.html`.

[61] M. Naghshvar, T. Javidi, et al. Active sequential hypothesis testing. *The Annals of Statistics*, 41(6):2703–2738, 2013.

[62] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[63] R. Narasimha, M. Lu, N. R. Shanbhag, and A. C. Singer. Ber-optimal analog-to-digital converters for communication links. *Signal Processing, IEEE Transactions on*, 60(7):3683–3691, 2012.

[64] R. Narasimha, G. Zeitler, N. Shanbhag, A. C. Singer, and G. Kramer. System-driven metrics for the design and adaptation of analog to digital converters. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5281–5284. IEEE, 2012.

[65] S. Nitinawarat, G. K. Atia, and V. V. Veeravalli. Controlled sensing for multi-hypothesis testing. *Automatic Control, IEEE Transactions on*, 58(10):2451–2464, 2013.

[66] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision research*, 37(23):3311–25, Dec. 1997.

[67] A. J. Parker and M. J. Hawken. Two-dimenisonal spatial structure of receptive fields in monkey striate cortex. *Journal of the Optical Society of America. A*, 5:598–605, 1988.

[68] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, Jan. 2008.

[69] J. G. Proakis. *Digital communications*. McGraw-Hill Science/Engineering/Math, 4 edition, 2000.

[70] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow. Distributed beamforming with software-defined radios: frequency synchronization and digital feedback. In *Global Telecommunications Conference (GLOBECOM 2012), 2012 IEEE*, Anaheim, CA, December 2012. IEEE.

[71] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. Lecun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[72] D. L. Ringach. Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *Journal of neurophysiology*, 88(1):455–63, July 2002.

[73] R. Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5(12):583–601, Dec. 1965.

[74] S. Shamai. Information rates by oversampling the sign of a bandlimited process. *Information Theory, IEEE Transactions on*, 40(4):1230–1236, 1994.

[75] P. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[76] J. Singh, O. Dabeer, and U. Madhow. On the limits of communication with low-precision analog-to-digital conversion at the receiver. *Communications, IEEE Transactions on*, 57(12):3629–3639, 2009.

[77] J. Singh and U. Madhow. On block noncoherent communication with low-precision phase quantization at the receiver. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2199–2203. IEEE, 2009.

[78] J. Singh and U. Madhow. Phase-quantized block noncoherent communication. *CoRR*, abs/1112.4811, 2011.

[79] J. Sjöstrand, N. Conradi, and L. Klarén. How many ganglion cells are there to a foveal cone? *Graefe's Archive for Clinical and Experimental Ophthalmology*, 232(7):432437, 1994.

[80] D. A. Sobel and R. W. Brodersen. A 1 Gb/s mixed-signal baseband analog front-end for a 60 GHz wireless receiver. *Solid-State Circuits, IEEE Journal of*, 44(4):1281–1289, 2009.

[81] F. Sun, J. Singh, and U. Madhow. Automatic gain control for ADC-limited communication. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[82] A. Varzaghani, A. Kasapi, D. N. Loizos, S.-H. Paik, S. Verma, S. Zogopoulos, and S. Sidiropoulos. A 10.3-GS/s, 6-bit flash adc for 10G ethernet applications. 2013.

[83] A. Wadhwa and U. Madhow. Blind phase/frequency synchronization with low-precision adc: a bayesian approach. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*. IEEE, 2013.

[84] A. Wadhwa, U. Madhow, and N. Shanbhag. Space-time slicer architectures for analog-to-information conversion in channel equalizers. In *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014.

[85] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013.

[86] B. A. Wandell. *Foundations of vision*. Sinauer Associates, 1995. Available from https://foundationsofvision.stanford.edu/.

[87] A. B. Watson and A. J. Ahumada. A standard model for foveal detection of spatial contrast. *Journal of Vision*, 5(9):6, 2005.

[88] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[89] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

[90] G. Zeitler. Low-precision analog-to-digital conversion and mutual information in channels with memory. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010.

[91] G. Zeitler. *Low-precision quantizer design for communication problems*. PhD thesis, Technische Universität München, 2012.

[92] G. Zeitler, A. Singer, and G. Kramer. Low-precision A/D conversion for maximum information rate in channels with memory. *Communications, IEEE Transactions on*, 60(9):2511–2521, 2012.

[93] S. Zhong. Efficient online spherical k-means clustering. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 5, pages 3180–3185. IEEE, 2005.