Estimation and Calibration Algorithms for Distributed Sampling Systems

by

Vijay Divi

S.B., Massachusetts Institute of Technology (2002)
S.B., Massachusetts Institute of Technology (2003)
M.Eng., Massachusetts Institute of Technology (2004)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author . . . .
Department of Electrical Engineering and Computer Science
August 12, 2008

Certified by . . . .
Gregory W. Wornell
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Terry P. Orlando
Chair, Department Committee on Graduate Students

Estimation and Calibration Algorithms for Distributed Sampling Systems

by

Vijay Divi

Thesis Draft: August 12, 2008

Abstract

Traditionally, the sampling of a signal is performed using a single component such as an analog-to-digital converter. However, many new technologies are motivating the use of multiple sampling components to capture a signal. In some cases such as sensor networks, multiple components are naturally found in the physical layout; while in other cases like time-interleaved analog-to-digital converters, additional components are added to increase the sampling rate. Although distributing the sampling load across multiple channels can provide large benefits in terms of speed, power, and resolution, a variety mismatch errors arise that require calibration in order to prevent a degradation in system performance.

In this thesis, we develop low-complexity, blind algorithms for the calibration of distributed sampling systems. In particular, we focus on recovery from timing skews that cause deviations from uniform timing. Methods for bandlimited input reconstruction from nonuniform recurrent samples are presented for both the small-mismatch and the low-SNR domains. Alternate iterative reconstruction methods are developed to give insight into the geometry of the problem.

From these reconstruction methods, we develop time-skew estimation algorithms that have high performance and low complexity even for large numbers of components. We also extend these algorithms to compensate for gain mismatch between sampling components. To understand the feasibility of implementation, analysis is also presented for a sequential implementation of the estimation algorithm.

In distributed sampling systems, the minimum input reconstruction error is dependent upon the number of sampling components as well as the sample times of the components. We develop bounds on the expected reconstruction error when the time-skews are distributed uniformly. Performance is compared to systems where input measurements are made via projections onto random bases, an alternative to the sinc basis of time-domain sampling. From these results, we provide a framework on which to compare the effectiveness of any calibration algorithm.

Finally, we address the topic of extreme oversampling, which pertains to systems with large amounts of oversampling due to redundant sampling components. Calibration algorithms are developed for ordering the components and for estimating the input from ordered components. The algorithms exploit the extra samples in the system to increase estimation performance and decrease computational complexity.

Thesis Supervisor: Gregory W. Wornell
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First, I would like to thank my advisor, Gregory Wornell, for his guidance throughout my graduate program. After undergrad, I was armed with mathematical tools and Greg showed me how to use them in order to conduct research and build my knowledge further. He also encouraged research outside of my primary thesis focus, which allowed me to develop a broad understanding of many other topics.

I would also like to thank my thesis committee, Professor Vivek Goyal and Professor Hae-Seung Lee, whose feedback has been extremely useful. Their input has strengthened my thesis and brought insightful perspectives of other fields into my research. Also, many thanks to Julius Kusuma for reviewing my thesis and to Shashi Borade, Ashish Khisti, and Charles Swannack for their input on my defense presentation. I would also like to thank National Defense Science and Engineering Graduate program (NDSEG) and Lincoln Laboratories for funding my research.

Throughout my graduate program, I have been surrounded by an amazing group of colleagues at MIT. My countless conversations with them have been invaluable to my research and to learning about other fields. From both the Signals Information and Algorithms lab as well as the Digital Signal Processing lab, I especially thank Anthony Accardi, Lane Brooks, Albert Chan, Venkat Chandar, Sourav Dey, Professor Uri Erez, Everest Huang, Ying-Zong Huang, Hiroyuki Ishii, Ashish Khisti, Emin Martinian, Urs Niesen, Maryam Shanechi, Charles Swannack, Aslan Tchamkerten, and Professor Rami Zamir. A number of other professors, including Professors Roger Mark, Muriel Medard, Alan Oppenheim, George Verghese, and John Wyatt, have guided me throughout the years and been key to my academic development.

My graduate school experience was enhanced by multiple summer internships. Positions at Mitsubishi Electric Research Labs under Dr. Bhiksha Raj and Dr. Peter Wolf and at Lincoln Laboratories gave me perspectives into other fields of research. My desire to travel and learn other cultures led me internationally for other summers. I was able to experience research outside MIT through a summer at EPFL in Switzerland under Professor Michael Unser. Working at Samsung Electronics in Korea under Dr. Soon Young Yoon taught me a great deal about wireless systems and opened my eyes to the amazing engineering being done outside the US. Lastly, I got the opportunity to apply my background to other fields while working with friend and colleague Akshay Naheta at Deutsche Bank in Hong Kong.

I have greatly appreciated the SIA environment, whose upbeat atmosphere is due in large part to Tricia Mulcahy. Aside from providing me with help anytime I needed it, any conversation we had was bound to put me in a better mood, even on the most stressful days. My gratitude is also extended to Cindy LeBlanc, whose support and political conversations I have always appreciated. One giant tool utilized during the thesis was the extensive computer network by Giovanni Aliberti. I will miss his help and wisdom when I leave MIT.

Outside of work, my friends have always been there for me throughout the way, including Anishee, Rahul, Rupa, Sandeep, Nihar, Pat, the C.R.C.'s of Grant, Josh, Karl, and Miguel, the Burton 2 crew, Maiya, Amy, Eric, Amita, Alice, Sonal, Krishna, Kiran and many others.

Lastly, and most importantly, I would like to thank my family for their love and support. Without the encouragement of my parents and my brothers Venu and Vasu, the research trek would have been infinitely times more arduous. I owe all of my achievements to them.

To my family

Contents

List of Figures

Chapter 1

Introduction

Many systems require the estimation of a signal from measurements of it. However, the system may not be able to directly observe the signal; instead, the input may first be processed by an arbitrary channel before it is observed as shown in Figure 1-1. The goal of the system is to invert the effects of the channel as best as possible in order to form an estimate of the input signal.

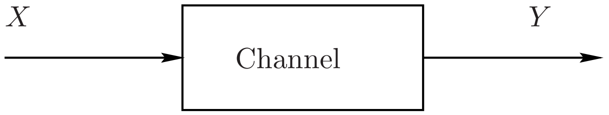

Figure 1-1: General System Channel

By allowing the input and channel to be arbitrary, it is clear to see that the estimation of input X from output Y is not well defined. Thus when analyzing such problems, we incorporate all knowledge about the class of input signals and type of channel into the problem. For example, the channel may be known or unknown. It may have certain properties such as being memoryless, noisy, linear, time-invariant, continuous, or stable. Similarly, the input may be restricted to a class of signals whose properties may include being bandlimited, range-limited, $\mathcal{L}_1$, $\mathcal{L}_2$, continuous, discrete, or differentiable. For each different scenario, a new approach for signal estimation can be developed with the goal of minimizing some error criterion. For the input class of continuous-time bandlimited signals and channel of distortion-free periodic sampling, the estimation of the input $x(t)$ from the output $y[n]$ is the well-known sinc interpolation.

Often, a system may observe the input signal through the output of multiple distinct channels, as seen in Figure 1-2. Using the output of both channels, a system can improve its estimation performance.

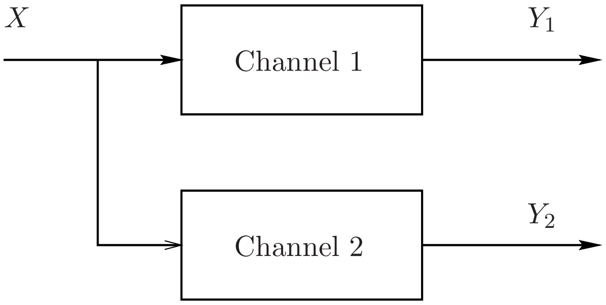

Figure 1-2: 2-Channel Distributed System

In the MIMO (Multiple Input Multiple Output) wireless setup, multiple antennas are used at the receiver to gather more information about the input. Because the channels have independent gain and noise realizations, the system can leverage these variations to increase the data rate and decrease the error. A similar situation arises in speaker localization using microphone arrays. Single channels many only provide a speaker direction, whereas three channels provides a 3-D position estimate, while even higher numbers of microphones produce increasingly accurate estimates.

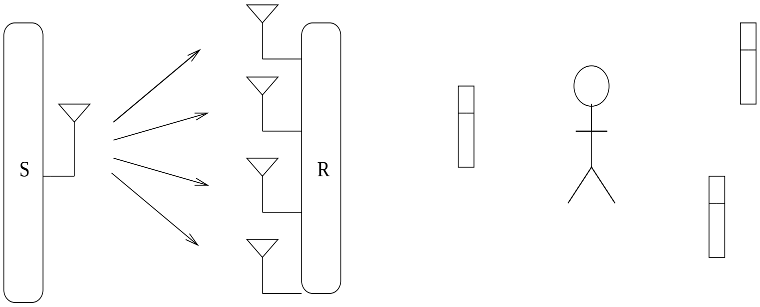

Figure 1-3: Example systems with multiple channels to capture signal data: MIMO antennas and microphone arrays

Another example of input reconstruction from multiple outputs is studied in [31]. Here the channels are analog-to-digital converters uniformly spaced in time. Although reconstruction is well known in this case, the authors study the optimal bit-allocation among the converters. The tradeoff between number of converters and converter resolution is studied in order to minimize the reconstruction noise.

When the channel is unknown, the first step can often be to estimate it. This can be performed with the aid of a training signal or blindly based on the channel output. One case of channel estimation is studied in [50] where the input is an unknown deterministic discrete signal and the channels are FIR filters. Under a certain set of necessary conditions, the channels and input signals can be identified from only the observed outputs.

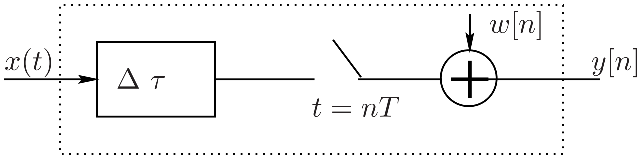

Figure 1-4: Distributed Sampling Channel

In this thesis, we examine methods for signal estimation for the class of channels found in distributed sampling systems. Observations of the input are attained through the outputs of multiple periodic sampling components. Because the components are not necessarily coordinated, each component may take samples starting at an arbitrary time with respect to the other components. Thus, each component is modeled as a channel that applies an arbitrary phase delay to the input and then periodically samples it, as seen in Figure 1-4. In this manner, the sampling is distributed across multiple channels that may be located on a single device or spread across multiple systems.

In order to reconstruct the input signal, it is necessary to calibrate the system. We develop blind calibration methods for estimating the unknown delay parameters associated with the sampling channels by restricting the input class to bandlimited input signals. Using these parameter estimates, multiple methods for efficient reconstruction are developed. Bounds on system performance are also developed. We separately address the issue of calibration in systems with a large amount of oversampling due a redundant number of sampling channels. For this case, we develop alternative methods that exploit the sampling redundancy in order to estimate the input.

1.1 Single Sampling Channel

We begin by examining the properties of a single sampling channel. To start, we assume that the delays in the channel model above are zero and readdress this issue later in the multi-channel distributed model. In the absence of noise, a channel samples periodically with period T_s, i.e., $y[n] = x(nT_s)$, as seen in Fig. 1-5.

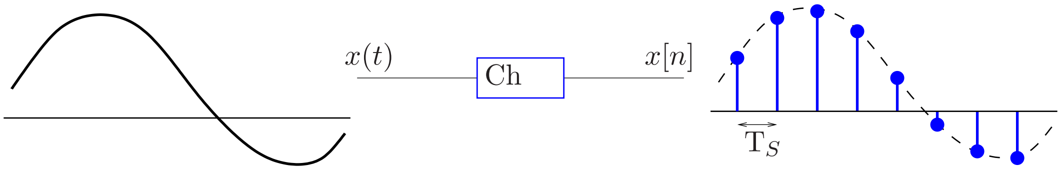

Figure 1-5: Single sampling component operating on analog input

Given these output samples, there are multiple possible interpolations that can generate the information content between samples, shown in Fig. 1-6. However, without more information on the input, it is unclear which choice is best.

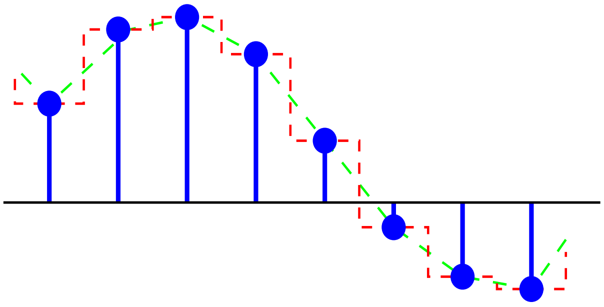

Figure 1-6: Input estimation from linear and step interpolations of periodic samples

For certain classes of inputs, this confusion does not arise. In these cases, periodic samples provide a complete representation and can be used to uniquely determine the input. The most widely discussed class is bandlimited signals, where sampling theory states that for an input with maximum frequency of B Hz, the input can be perfectly reconstructed by sampling at a rate higher than $f_s = 2B$. As we shall discuss later, additional benefit can be gained by sampling faster than the Nyquist rate in certain situations.

This model of the channel as a periodic noise-free component is a bit optimistic. In most setups, there are a variety of factors that cause errors in the sampling. Possible distortions include an additive noise source or random phase. These issues can severely limit the ability to reconstruct the input. In order to understand the non-idealities that arise, we examine the problems that occur in the most common realization of a sampling channel: the analog-to-digital converter.

1.1.1 Analog-To-Digital Converters

The most ubiquitous sampling component in today's hardware is the analog-to-digital converter (ADC). Just as its name implies, this piece of hardware converts an analog continuous-time signal into a digital discrete-time signal. In general, ADCs sample the analog input periodically. To ensure a fixed amount of time between samples, ADCs use the system clock and often sample at a certain fraction of its rate. Quantizing the signal while sampling allows the analog values to be approximated through a set of 0/1 digital bits.

There are many architectures for implementing analog-to-digital conversion in hardware [36]. The most popular design structures for ADCs are: pipelined, sigma-delta, successive approximation register (SAR), flash, and folding. Each of these methods have their benefits and drawbacks. For a given application, the type of converter is selected based on its:

- Rate ($f_s = 1/T_s$) - speed of sampling measured in terms of samples/second or Hertz
- Power dissipation (P_d) - power consumed by converter during operation typically measured in Watts
- Resolution (B) - stated number of quantization bits; another related measurement is the effective number of bits (ENOB)

Also taken into consideration is the converter cost, which is commonly a function of its size and thus power dissipation. One common figure-of-merit for comparing ADC performance is

$$F = \frac{P_d}{2^B f_s}, \tag{1.1}$$

typically expressed in pJ/bit or fJ/bit. By this metric, increasing resolution by one bit provides a similar performance increase as doubling the sampling speed or cutting the power in half. We briefly examine the various types of converter architectures in order to understand the errors that arise in sampling.

One commonly used architecture is the pipelined ADC due to its high resolution and fast rate of operation. The converter works by using sequential stages, each stage determining the next significant bit of the digital representation. The first stage determines the most significant bit through a simple comparison. The subsequent stage then subtracts this bit (after a digital-to-analog conversion) off of the input value. The new value is gained by a factor of two and an identical comparison stage can be used to determine the next significant bit. The method can also be generalized to compute multiple bits per stage.

Pipelined converters provide a high rate and moderate resolution, while keeping die area and power consumption low.

An alternative to the pipelined architecture is flash ADCs, which are useful due to their extremely fast rates. They operate by comparing the input voltage to a reference ladder of 2^B evenly spaced voltages, where each step requires a unique comparator. The exponential number of comparators limits the resolution of such designs due to the extra space and cost associated with more bits. Thus, current flash ADCs are low resolution (≤ 8 bits) but their fast comparison time allows for rates over 1 Gsps (gigasample per second).

Sigma-delta ADCs offer a significantly higher resolution than any other ADC on the market, reaching up to 24 bits. By highly oversampling the input signal, the noise can be shaped to appear in out-of-band frequencies. After shaping, the noise is then filtered out during the decimation process. This removal leads to higher resolutions, although overall decimated rates are limited to below 10 Msps.

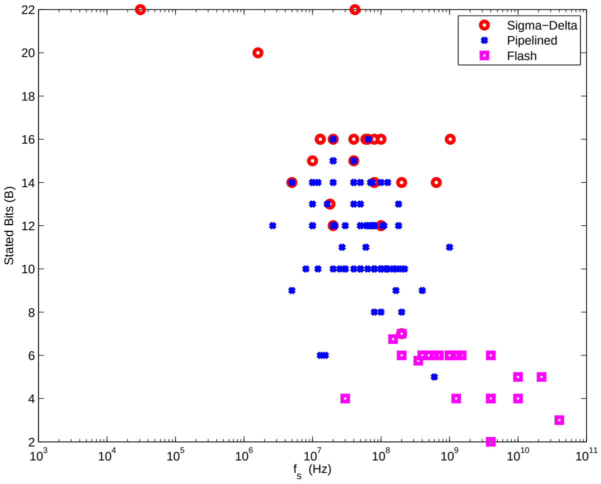

Figure 1-7: Performance plot for various types of ADC architectures measuring stated number of bits against the sampling rate.

The current performance capabilities of each type of converter is plotted in Fig. 1-7 (data courtesy of B. Ginsberg at MIT). The data points on the plot indicate the resolution versus sampling rate for existing converter designs. As we can see, the sigma-delta points dominate the high resolution regime and the flash dominate the high rate. The pipelined ADCs provide the middle range and lie along the current performance line bound.

For a converter resolution of B bits and full voltage range of E volts, the quantization steps have size $E/2^B$, giving an input-output function

$$x_{out} = \text{round}\left(\frac{2^B}{E}x_{in}\right)\frac{E}{2^B}. \tag{1.2}$$

However, due to design issues in the various architectures, this ideal linearity is never achieved. Problems such as capacitor mismatch or limited opamp gain lead to abnormalities in the curve [7]. In Fig. 1-8, we plot the transfer curve for an ideal ADC (dashed-dot) and compare it to the transfer curve for an example non-ideal ADC (solid dots). As we can see, the various distortion sources lead to decreased performance and a lower number of effective bits.

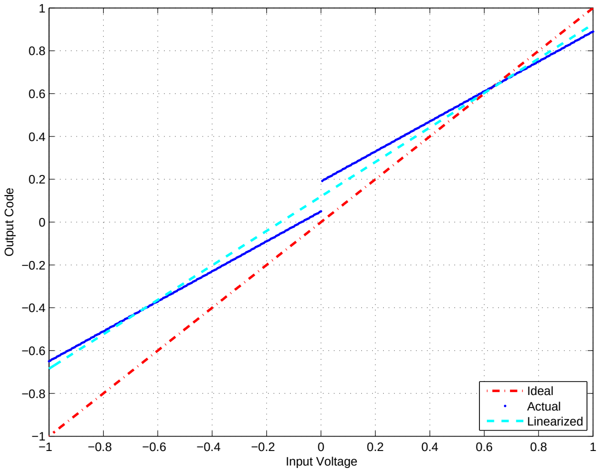

Figure 1-8: Single ADC input-output transfer curves for ideal ADC, example ADC, and linearized ADC.

To model the combination of distortion sources, we construct a first order approximation of the transfer curve, which yields the dashed line plotted in Fig. 1-8. Thus, the converter can be modeled through a gain g, which represents the line slope, and an amplitude offset A, which represents the line intercept, such that

$$x_{out} = A + gx_{in} + w \tag{1.3}$$

where w represents the remaining error, which we model to be Gaussian white noise. This linearization of the transfer curve conceals the higher-order nonlinear effects that occur in the ADC. Its general structure allows us to use it as a model for any converter architecture; and it also forms a robust model for general sampling channels.

1.1.2 Noise Sources

In (1.3), we model our noise as white Gaussian; however the noise is actually a combination from multiple sources. The most predominant sources being nonlinearities, thermal noise, shot noise, quantization noise, and noise from sampling jitter. Each noise has its own distribution, which we describe more carefully to justify the validity of the Gaussian approximation.

Quantization noise occurs due to the rounding necessary for digital representation. Assuming a wide range of input values, the quantization noise is modeled as having a uniform distribution in the range $[-E/2^{(B+1)}, E/2^{(B+1)}]$. The variance of this noise decreases as we increase the number of quantization bits

$$\sigma_Q^2 = \frac{1}{12}\left(\frac{E}{2^B}\right)^2. \tag{1.4}$$

Although quantization is the predominant source of error in some ADCs, we approximate it as white Gaussian noise for analytic tractability.

The linearization of the ADC transfer curve masks the curve's nonlinearities, which results in another source of error. Nonlinear effects arise for several reasons in each type of converter architecture and therefore have different characteristics. It is difficult to model this noise more precisely without specifying the converter type and specific design parameters. For this reason, we ignore the details of these nonlinear effects by modeling it as Gaussian noise and develop calibration schemes that can broadly be applied to any ADC.

Thermal noise is generated by the effects of thermal motions on the electrons of a circuit. The noise generally has uniform power across all frequencies, thus it is modeled as white Gaussian noise. Another electronic noise is known as shot noise. The noise arises due to the discrete nature of electron charge. These fluctuations are modeled as Poisson, which approaches the Gaussian distribution for large values. Our ADC noise model fits this Gaussian model; however, we consider the effects of thermal and shot noise to be negligible in comparison to quantization noise. Clock jitter leads to another source of noise in sampling. Because the ADC clock signal cannot be generated perfectly, each sample time has a random perturbation from its ideal location. These small timing errors lead to a noise whose power is proportional to frequency. Again, we ignore the specific treatment of this noise because the timing skews we model in the distributed setup are significantly larger than the jitter.

1.1.3 High-speed ADC Applications

ADCs are found in a multitude of electronic devices from digital cameras to electrocardiogram machines. These devices have a wide range of sampling requirements in terms of rate, resolution, and power. Although the current on-the-market ADCs are sufficient for many applications, there are an increasing number of products that require faster sampling due to very high bandwidth analog inputs. One common place is communication systems, where data is being transmitted in the high frequency spectrum. Another recent area of interest is imaging technologies that use millimeter wavelength signals. We briefly explore some motiving examples.

Ultra-wideband (UWB) communication systems operate by spreading information content across multiple orthogonal frequency bands. By using low power in each band, UWB can exploit the extra bandwidth to transmit while not interfering with other communication in the spectrum. Transmission often occurs through the use of signal pulses, which occupy bandwidths in the range of 500MHz to 10GHz. High speed ADCs are necessary in the channel receiver in order to ensure the signal pulses are detected. UWB technology is proving useful in high-rate, short distance communication and continues to show promise for other applications.

Software defined radio is another application which typically uses high-speed ADCs. In this case, the transmitter and receiver use general purpose processors, rather than an application specific integrated circuit (ASIC). The software on the processors allow for a large amount of variability in the transmission scheme. Code rate and frequency band are among the possible variables that can be dynamically changed in the system. In classic software defined radio, the receivers directly digitize the high frequency RF signal. However, to alleviate some of the burden, the signal may be demodulated to an intermediate band before sampling with an ADC and performing the remaining computation in hardware. Because the intermediate frequency is still large, a high-speed converter is still necessary for sampling.

These systems have sampling requirements that are starting to exceed the limits of the current state-of-the-art ADC performance. Although it may be possible for a single-channel ADC to service the needs of all these applications in the future, the need for speed will always exist. Newer technologies will require faster sampling components. To address this issue, multiple ADC systems can be implemented to increase the sampling rate of any single ADC technology. The multi-ADC scheme not only applies to converter technologies but also generalizes to any distributed sampling system.

1.2 Distributed Sampling Systems

By using multiple sampling channels on a common input, one can often increase the amount of information known about the input and produce a better estimate. There are many ways in which to combine the information from the individual channels. For example, in bandlimited systems if the channels all have the same sampling times, their samples can first be averaged to reduce noise effects.

In other systems, multiple sampling channels be used to distribute the sampling load across channels. By staggering the timings of the channels, samples are attained at a larger number of timing locations and the rate per channel can be decreased while still meeting the Nyquist criterion. We describe the multi-channel system in the context of time-interleaved ADCs, although this model generalizes to any distributed sampling system with multiple channels.

1.2.1 Time-Interleaved ADCs

One widely explored application with distributed sampling is the time-interleaved analog-to-digital converter (TIADC), where signal sampling is managed by multiple converters. TIADCs operate in a round-robin manner. In a system of M converters, to realize a system sampling period of T_s, each converter operates with sampling period MT_s and a spacing of T_s between consecutive converters. Thus, the sampling rate required by the ADCs in the system is reduced by a factor of M.

There are many reasons to use distributed sampling on a chip. The use of a single analog-to-digital converter (ADC) is often insufficient to handle the requirements of current applications. The single converter may not be able sample fast enough or may consume a large amount of power in order to do so. At high sampling rates, TIADCs offer an attractive method of sampling by sharing the load across many converters, allowing for a lower amount of overall power consumption and a greater control over sampling accuracy.

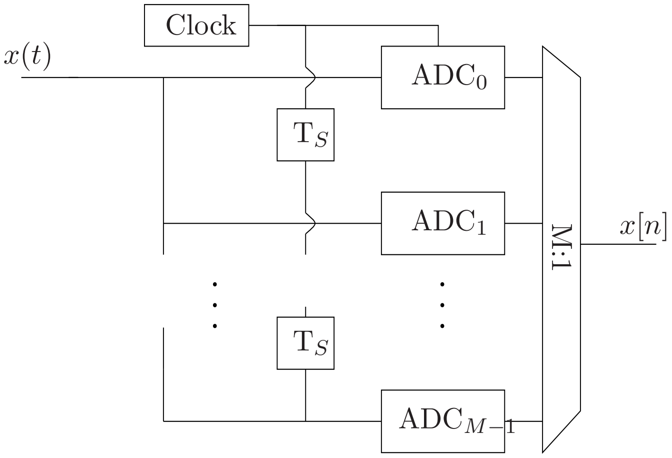

Figure 1-9: Ideal time-interleaved ADC system with M converters

Although TIADCs may avoid some of the problems presented by using a single high speed ADC, they also introduce a new set of problems. In particular, variations among the individual ADCs in a time-interleaved system lead to inaccurate sampling [5, 41, 32, 48]. The primary source of error in TIADCs is timing skew, which is caused by signal path length differences. Although system designers try to ensure that the clock signal and input signal travel uniform distances to each converter, restrictions in the physical layout introduce unavoidable errors. Even when clock and signal paths are precisely matched, variations in gate delays also result in timing skew. The skew is often modeled as a fixed parameter of the system, but may change after long periods of time.

Other sources of error in the individual converters are modeled as a gain mismatch and amplitude offset as in (1.3). Because the converters are not identical, the gain and offsets vary among the converters [14]. Although these gain and offset nonuniformities can degrade performance, a variety of circuit based matching and calibration techniques exist to reduce their effects. Therefore, in our development of calibration methods, we focus on the timing mismatch.

These sources of error in TIADCs also provide a general model for the distributed sampling system, where channels may have independent gains and time delays. In this thesis, we develop calibration methods that are designed for the minimization of such errors. We focus on methods which use digital post-sampling processing to increase the performance in the TIADC system.

1.2.2 Applications

With efficient calibration, the multi-channel TIADC can be used to replace the single channel ADC in any application. Although its power versus performance tradeoff has yet to prove beneficial in the low-rate regime, it is being implemented in future high-speed applications such as UWB and millimeter imaging.

The TIADC channel model generalizes to other distributed sampling systems. Rather than assuming all channels are on a common chip with a common clock signal, there are many other natural realizations of the multi-channel model. Sensor networks that have independent nodes sampling a common signal also face the same issues. Timing skew can occur due to the lack of synchronization and physical locations can cause gain variations.

In the following section, we present the problem formulation and detail the possible input signal classes which can be used for calibration.

1.3 Calibration Setup

We now describe the general problem setup for the multi-channel distributed sampling system. The input to the system $x(t)$ is modeled as a bandlimited signal with cutoff frequency Ω_c, i.e. the continuous time Fourier transform $X(j\Omega) = 0$ for $\Omega_c < |\Omega|$. The overall sampling period T_s of the system is chosen to ensure that the sampling rate strictly exceeds the Nyquist frequency, i.e., $T_s < \pi/\Omega_c$, thus creating some amount of excess bandwidth. The signal recovery problem is to estimate

$$x[n] = x(nT_s), \tag{1.5}$$

which is bandlimited to $\omega_c = \Omega_c T_s < \pi$, as accurately as possible from the component outputs.

We model the output of the ith constituent component as

$$y_i[n] = x(nMT_s + \tau_i') + w_i[n] \tag{1.6}$$

where the τ_i' model the unknown timing shifts. Without loss of generality, we can choose an arbitrary time reference, thus we let $\tau_0' = 0$. In ideal systems which contain uniformly spaced components, the timing shifts are spaced T_s time apart, i.e. $\tau_i' = iT_s$.

In systems with small amounts of error, the timing shifts can be modeled through their deviation from uniform timing,

$$\tau_i' = iT_s + \tau_i \tag{1.7}$$

where the τ_i represent the skew from uniform timing. The gains g_i and offsets A_i are assumed to be pre-calibrated although their treatment is considered in subsequent sections. The $w_i[n]$ in (1.6) represents the aggregate noise, modeled as white Gaussian whose variance primarily depends on the number of bits to which the input is quantized.

We use the following notation to represent the signal obtained by multiplexing the component outputs

$$y[n] \quad = \quad y_i\left[\frac{n-i}{M}\right] \qquad n \pmod{M} = i. \tag{1.8}$$

This received signal is also referred to as the uncalibrated signal.

In Figure 1-10, an example sampling pattern is shown for $M = 4$ components. The multiplexed output (1.8) is nonuniform periodic (recurrent) samples of the input.

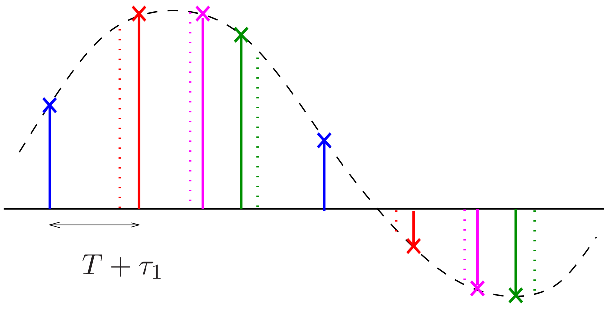

Figure 1-10: Signal sampled with $M = 4$ sets of periodic samples. Dotted lines show uniform samples. Solid lines show nonuniform samples.

1.3.1 Input Signal Classes

Within this thesis, we examine a variety of different input signal classes. The classes are generated from models of real-world problems. We begin with arbitrary bandlimited signals and then examine narrower and broader classes of signals. In general, as we add more restrictions to the input class, the complexity increases however there are often more characteristics that can be used for blind calibration. We now introduce multiple situations that motivate these classes.

Bandlimited Input Signals

A continuous-time bandlimited signal is a signal whose power goes to zero outside a finite frequency range. This class has long been used to model a variety of signals in the real world. There are both mathematical and physical reasons to do so. Mathematically, it provides a separate basis on which to represent and process the signal. And physically, it can represent signals that are continuous and differentiable and where the rate at which the variable changes is limited by a physical restriction. As we stated, this class is only a model; thus, it is possible that the actual signal deviates from the class. We model this deviation as error and ensure its effects are negligible as long as it stays small.

Stationary Signals

In many situations, the statistics of the input signal are constant over time, a property known as stationarity. The statistics commonly dealt with are the mean and variance of the signal, as well as the correlation between samples spaced by fixed distances apart. Fixed mean and autocorrelation is known as wide-sense stationary.

Stationarity is a common property for models of signals in engineering systems. When a communications channel is not changing rapidly, the signals transmitted and received have a fixed power spectral density (implying a fixed autocorrelation) for a period of time. This period is known as the coherence time for which the statistics are constant. Longer coherence times for the spectrum are possible in systems like sensor networks and digital subscriber line (DSL). For other channels, such as cell phone communications, the coherence interval is shorter due to mobility and interference variations.

Arbitrary Signals

Although not built into the current model, arbitrary signals can also be sampled in a distributed sampling system. The channel outputs provides a partial representation of input but does not guarantee the possibility of perfect reconstruction. For example, in the case of continuous-time Brownian motion as an input, the variance can be estimated but the signal itself cannot be reconstructed from periodic samples. Also, a signal with sharp jumps cannot be fully represented by its samples. In some systems, this partial representation is

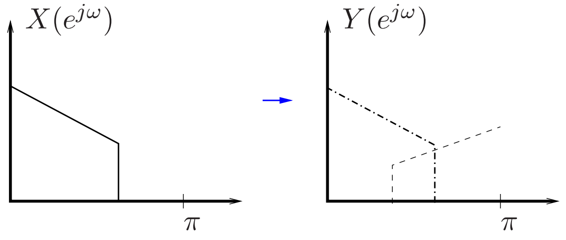

Figure 1-11: Example of aliasing due to periodic nonuniform sampling with $M = 2$ components

sufficient; however, many systems desire the ability to uniquely determine the input. As described in [2], unique reconstruction from nonuniform sampling is possible for other input classes, such as shift-invariant spaces.

1.3.2 Bandlimited Class Restrictions

Although we specify that $x[n]$ must be bandlimited, it is necessary to add additional requirements on the input class in order to ensure that accurate blind reconstruction can be obtained in a time-interleaved system. In particular, we restrict ourselves to the class of nontrivial input signals for which nonuniform periodic sampling (with non-zero skew) yields aliased content in the frequency band $\omega_c < \omega < \pi$, i.e.

$$Y(e^{j\omega}) \neq 0 \quad \text{for some } \omega_c < \omega < \pi. \tag{1.9}$$

This requirement on the input is equivalent to saying that it must be possible to detect that the received signal $y[n]$ is not bandlimited to ω_c. Without this requirement, the problem becomes ill-posed as the input signal $x(t)$ which generates output $y[n]$ may not be unique. It will be straightforward to establish that for our class of signals, the out-of-band energy in $\omega_c < \omega < \pi$ for the estimated signal $\hat{x}[n]$ is zero if and only if the parameter values so determined are the correct ones.

The calibration methods presented can also work for a larger class of signals, where the aliased content due to nonuniform sampling appears in other bands of the spectrum; in this case, the algorithms can be redefined with small modifications made to the filters, as discussed in subsequent chapters.

To understand this better, we briefly examine some example subclasses of bandlimited input signals:

- Inputs $x(t)$ which are spectrally 'full'

$$X(e^{j\omega}) \begin{cases} \neq 0 & |\omega| \leq \omega_c \\ = 0 & \omega_c < |\omega| \leq \pi \end{cases} \tag{1.10}$$

  Aliasing occurs in the band $\omega_c < |\omega| \leq \pi$ and the algorithm can be used to accurately estimate the unknown timing skews.

- Passband signals with aliasing in other bands, e.g.,

$$X(e^{j\omega}) \begin{cases} \neq 0 & \frac{\pi}{2} \leq |\omega| \leq \frac{2\pi}{3} \\ = 0 & \text{otherwise} \end{cases} \tag{1.11}$$

  For $M = 2$ converters, the aliasing in $y[n]$ appears in the band $\pi/3 \leq |\omega| \leq \pi/2$. Thus, the signal is still bandlimited to $2\pi/3$. The calibration algorithms which are presented can be modified to handle such signals as long as the input signal band $\pi/2 \leq |\omega| \leq 2\pi/3$ is known to the system.

- Passband signals whose aliasing occurs in the passband, e.g.,

$$X(e^{j\omega}) \begin{cases} \neq 0 & \frac{\pi}{3} \leq |\omega| \leq \frac{2\pi}{3} \\ = 0 & \text{otherwise} \end{cases} \tag{1.12}$$

  For $M = 2$ converters, the aliasing also appears in the band $\pi/3 \leq |\omega| \leq 2\pi/3$. Any estimate of the timing skews will yield an estimate $\hat{x}[n]$ which lies in this passband; thus, the signal cannot be accurately reconstructed.

In systems with $M > 2$ ADCs, the spectral content is aliased to $M-1$ locations and there may be more locations in which the aliasing in $y[n]$ is detectable. As shown in subsequent chapters, knowledge of all aliasing spectrum locations with zero signal content will increase calibration accuracy.

1.4 Calibration Background

Digital calibration of analog circuitry has been present for many technological generations. As new technologies require the use of higher performance analog circuitry, often the analog components cannot realize the necessary system specifications in a power- and cost-efficient manner. Also, process variations in chip production can require larger design margins to keep chip yields high.

With digital performance scaling aggressively, large benefits can be achieved by finding digital methods to compensate for inadequate analog circuit performance and allowing the system designer to relax analog circuit constraints. Advantages can include a reduction in cost, power consumption, and size, along with providing an increase in speed, accuracy, testability, and system robustness. In the TIADC system, multiple ADCs can be used to reduce the load on individual ADCs and digital compensation techniques can further broaden the range of acceptable circuit designs.

The calibration of timing-skews requires a careful approach. Because the output of an individual channel is under the Nyquist rate, it only provides partial representation of the input. The traditional techniques of time-delay estimation are ineffective due to the aliasing; thus multiple channels must be combined to estimate the input. This multiplexing introduces timing skew between samples, whose effects are nonlinear.

There are many different possible approaches to system calibration (input reconstruction) in such systems. In general, the estimation of the mismatch parameters can be decoupled from the TIADC and reconstruction, as shown in Fig. 1-12. After the estimates are formed, they can be used to modify the analog sampling circuit [26] or to modify the reconstruction filter [11]. Also, some systems assume knowledge of the input at the estimation box, usually with the use of a training signal. These two characteristics: whether the system uses skew estimates in the analog or digital domain and whether the system uses a training signal, can be used to differentiate the calibration algorithms. We briefly review the literature background on calibration techniques that have been proposed in the past.

A training signal can be used in calibration by either pausing the normal input or by injecting the training signal into the normal input. Because pausing the input leads to system delays, this method is usually not preferred in practical implementation. Instead, a known background signal can be added to the input, which facilitates the estimation of

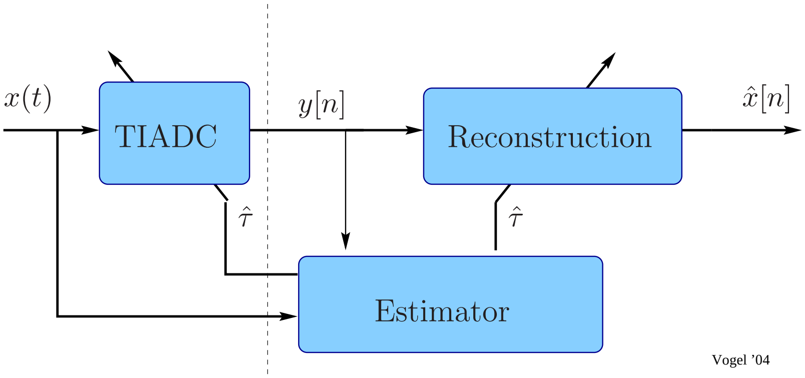

Figure 1-12: System layout of calibration decouples mismatch estimation and signal reconstruction

the timing-skews. After skew estimation, this known signal is then subtracted away in the digital domain. In [27], a basic method for training signal based estimation is presented by inputting a reference function to the TIADC and measuring the difference between the sampled value and the known value. Multiple types of training signals are analyzed, including the ramp, triangular, and sine functions. The various implementations of such a method require extra hardware, decrease the sampling resolution, and can require long calibration times if performed in the background.

An alternate approach to calibration is to perform blind recovery using only the ADC outputs. This approach does not require any dedicated time for calibration. Such methods may use oversampling and take advantage of the excess bandwidth in the system to enable parameter estimation. Or other properties of the input like stationary statistics can also be exploited for estimation The primary focus is to perform calibration without placing significant restrictions on the input class and while keeping computation complexity low for large numbers of converters; however, it is often the case that one of these goals is not achieved.

There has been growing interest in such blind methods for time-interleaved ADCs [10, 39, 16, 25, 11, 44, 45, 23, 24, 49, 47]. Because computing the optimal parameter estimates involves the minimization of a complicated nonlinear function of the timing-skews, the solution is not analytically tractable and therefore difficult to implement in hardware. Multiple calibration algorithms have been developed to approximate this solution.

In [39], a basic method for skew-estimation is presented by performing an exhaustive search over the parameter space. By quantizing the M dimensional space of skews, each possible set of skew values can be tested to minimize a certain heuristic. Assuming there are no local minima, the method can converge upon true values. Although some reduction of the space is possible, the solution is exponential in terms of the number of timing skews and requires are large amount of computation.

Other approaches impose a stationarity property on the input. By doing so, the system can take advantage of the correlation between samples to produce a low complexity calibration scheme. Only the computation of the difference between channel outputs is necessary in [16]; while [44] focuses on designing a TIADC reconstruction filter whose output matches the statistics of the input signal. For $M = 2$ converters, an efficient hardware implementation is possible [45]; however estimation is difficult for systems operating close to the Nyquist rate. An alternate 2-ADC blind calibration method that does not require input stationarity is proposed in [25]. This method uses a modified phase detector to estimate timing skew. Although the calibration algorithm has low complexity, it requires large amounts of oversampling and does not generalize to $M > 2$ converters.

In [24, 24], a frequency based approach is presented that takes advantage of the fact that only $M - 1$ copies of the aliasing are present at the frequency band around zero. Time-skews are picked to invert the aliasing effect and simplifications that reduce complexity are also presented. Calibration shows promising performance for systems with few converters ($M = 2$ in [23], $M = 4$ in [24]). For larger numbers of converters, the algorithm becomes computationally inefficient because the complexity-reducing steps are no longer valid.

Calibration of systems with larger numbers of converters has been examined in [11], which is presented in this thesis, and by other authors [49, 47]. By using Taylor series approximations to linearize the problem, vast advances have been made that reduce the complexity while providing accurate parameter estimates. These methods address the problem in both the time and frequency domain.

In general, these calibration schemes are designed to perform time-skew parameter estimation. From these estimates, it is necessary to provide a reconstruction method to estimate the input. While the original method of nonuniform periodic reconstruction was presented in [52], multiple methods have been developed that focus on efficient signal reconstruction [34, 17, 28, 29, 46, 12]. These works describe methods to reduce complexity such as filter approximations and parallelization techniques In this thesis, we develop new methods for both parameter estimation and signal reconstruction as well as bounds for their performance. We now detail an outline of the thesis.

1.5 Thesis Outline

The thesis is organized as follows. In Chapter 2, we present multiple methods of input signal recovery in systems when the time-skews of the channels are known. Starting from basic nonuniform reconstruction methods, we develop a reconstruction approximation that can be efficiently implemented. We also develop a method that is optimal in the presence of additive noise on the samples. Using the reconstruction approximation, we construct and solve a least-squares formulation for estimating the unknown mismatch parameters in Chapter 3. The estimation algorithm is developed for the small mismatch case, where the time-skews τ are small in magnitude; however, it is also applicable to systems with larger skews.

The performance bounds of nonuniform sampling with noise are computed in Chapter 4 through an examination of the eigenvalues of sampling matrices. The performance gain of adding extra sampling components is also presented. This case of 'extreme oversampling' is discussed in Chapter 5. By greatly increasing the number of channels, it is possible to take advantage of the oversampling to reduce the calibration complexity. Algorithms are presented for ordering samples and for efficiently estimating the input. We conclude with a summarization of results and remarks about future work in Chapter 6.

Chapter 2

Signal Reconstruction

In this chapter, we explore methods of signal reconstruction in distributed sampling systems, as seen in Fig. 2-1. Distributed systems are able to sample an analog signal at a fast rate by allowing nonuniform sampling. However, there is added complexity and noise sensitivity when reconstructing uniform samples. We develop computationally efficient and noise-resistant methods for signal reconstruction from periodic nonuniform samples when the timings are known.

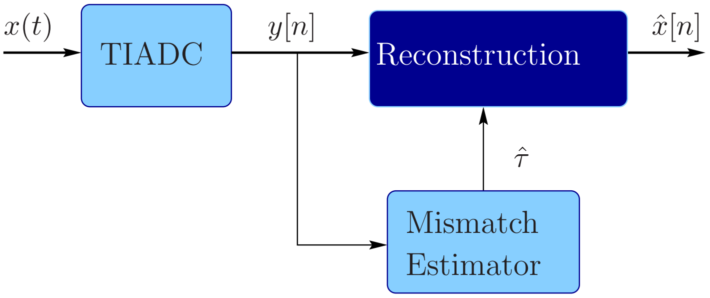

Figure 2-1: Signal Reconstruction

Methods for reconstruction of uniform samples from nonuniform samples have been widely explored [4, 51, 2]. The case of periodic nonuniform sampling has also been studied in detail since it allows for practical implementations. In [17], the added structure of the periodic nonuniform sampling is exploited to develop a filterbank that reconstructs bandlimited inputs perfectly. More computationally efficient methods for approximate reconstructions are given in [29, 46].

Reconstruction methods also operate in the presence of noise, which can be introduced through a variety of sources including quantization and thermal noise. This noise can seriously degrade performance when reconstructing uniform samples from the nonuniform samples [43]. This issue is addressed in the general setting for nonuniform sampling [2, 38]; however, it is often cursorily treated in efficient periodic nonuniform reconstruction.

We start by presenting a method for perfect reconstruction in the absence of noise. We then construct an approximation of this reconstruction method. By re-writing the problem in matrix form, we develop a low-SNR reconstruction scheme. Finally, we examine iterative reconstruction solutions and present simulation results. The benefits of each method are analyzed to determine the useful regime. The methods presented will also motivate the mismatch estimation methods that are proposed in later sections.

2.1 Perfect Reconstruction

We first examine the relationship between output $y[n]$ and uniform samples of the input $x[n]$ and then present methods of recovering $x[n]$ from $y[n]$. We develop our reconstruction for the high SNR (signal-to-noise ratio) regime, where $\sigma^2 \to 0$, and later introduce the effects of noise on our signal estimation.

Because the input is bandlimited to Ω_c and the sampling frequency is higher than the Nyquist frequency, the input can be written as

$$x(t) \quad = \quad \sum_m x[m] \operatorname{sinc}(t/T_s - m). \tag{2.1}$$

In the absense of noise, for $n \pmod{M} = i$, the output

$$\begin{aligned} y[n] \quad &= \quad x(nT_s + \tau_i) && (2.2) \\ &= \quad \sum_m x[m] \operatorname{sinc}((n-m) + \tau_i/T_s). && (2.3) \end{aligned}$$

This received signal can be viewed as the output of a linear time-varying filter

$$y[n] \quad = \quad (f_i * x)[n] \qquad n \pmod{M} = i \tag{2.4}$$

where the $f_i[n]$ filters represent the fractional sample delays in (2.3).

The general approach for input estimation is to invert the effects of the time-varying filters in (2.4). Typically, the inverse of a time-varying filter is difficult to compute; however, due to the added structure of our formulation, there exists a closed form solution. In the absence of noise, the input $x[n]$ can be reconstructed from $y[n]$ through another set of time-varying filters $g_i[n]$ such that

$$x[n] = (g_i * y)[n] \qquad n \ (\mathrm{mod}\ M) = i. \tag{2.5}$$

To give insight into the methods used to construct these filters, we briefly introduce some results from sampling theory.

A bandlimited analog signal can be reconstructed from its samples if the average sampling rate is faster than the Nyquist rate. One general method of nonuniform reconstruction is presented by Yao and Thomas [51]. Suppose $x(t)$ is a finite energy, bandlimited signal, $X(j\Omega) = 0$ for $|\Omega| > \Omega_c - \epsilon$ for some $0 < \epsilon < \Omega_c$. If the following properties are satisfied by the sample timings,

$$|t_n - n\frac{\pi}{\Omega_c}| < L < \infty \tag{2.6}$$

$$|t_n - t_m| > \gamma > 0 \quad n \neq m, \tag{2.7}$$

where t_n is the time of the nth sample, then $x(t)$ can be reconstructed from its samples $x(t_n)$. The reconstructed signal $x(t)$ is:

$$x(t) = \sum_{n=-\infty}^{\infty} x(t_n) \frac{G(t)}{G'(t_n)(t - t_n)} \tag{2.8}$$

where

$$G(t) = (t - t_0) \prod_{n=-\infty, n\neq 0}^{\infty} \left(1 - \frac{t}{t_n}\right). \tag{2.9}$$

By plugging the periodic nonuniform sampling pattern into t_n, the reconstruction (2.8) can be simplified. In [52], this simplification is developed for the M component distributed sampling system in the absense of noise

$$x(t) \quad = \quad \gamma(t) \sum_{\alpha=-\infty}^{\infty} \sum_{i=0}^{M-1} y[M\alpha + i] \frac{a_i(-1)^{\alpha M}}{\pi(t - \alpha T_A - \tau_i')/T_A} \tag{2.10}$$

where

$$a_i = \frac{1}{\prod_{k=0,k\neq i}^{M-1} \sin(\pi(\tau_i' - \tau_k')/T_A)}, \tag{2.11}$$

$$\gamma(t) = \prod_{k=0}^{M-1} \sin(\pi(t - \tau_k')/T_A) \tag{2.12}$$

with $T_A = MT_s$ denoting the sampling period of a single component and $\tau_k' = kT_s + \tau_k$. The discrete time equivalent filter can be derived by sampling at times $t = nT_s$.

Equation (2.10) performs perfect reconstruction when no noise is present in the system and sample timings are known precisely. However, a large amount of computational power is necessary to compute and implement the time-varying filters. Also, the reconstruction is suboptimal in the presence of noise. In the following sections, we develop alternative methods that address these issues more carefully.

2.2 Reconstruction Approximation

The reconstruction (2.10) is nonlinear in terms of the timing skews. To simplify this relationship, we turn to an approximation of the reconstruction equation where we assume that the timing skews are small in magnitude ($|\tau_i| \ll T_s$). This approximation is valid within the high-resolution time-interleaved ADCs, as discussed later. Aside from decreasing the complexity of the reconstruction, the approximation presented will also allow for easier estimation of the unknown mismatch parameters.

In Appendix A, we derive an approximation to (2.10) by the first-order Taylor series around the point $\boldsymbol{\tau}/T_s = 0$, where vector $\boldsymbol{\tau}$ represents the timing skews

$$\boldsymbol{\tau} = [\tau_1 \;\; \tau_2 \;\; \ldots \;\; \tau_{M-1}]^T . \tag{2.13}$$

At times $t = nT$, the approximation yields

$$\hat{x}[n] \;\; \approx \;\; y[n] - \frac{\tau_i}{T_s}(h * y)[n] \qquad n \ (\mathrm{mod}\ M) = i \tag{2.14}$$

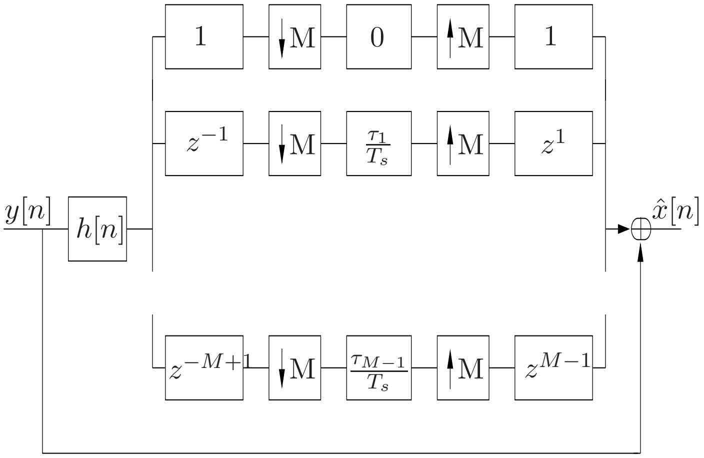

Figure 2-2: Recovery filterbank for an M component distributed sampling system using time-skew values $\boldsymbol{\tau}$

where

$$h[n] = \begin{cases} 0 & n = 0 \\ \frac{(-1)^n}{n} & \text{otherwise} \end{cases} \tag{2.15}$$

is a discrete-time derivative filter and $\tau_0 = 0$ as specified previously. The linearity of the τ_i in (2.14) will be useful for our blind mismatch estimation.

A filterbank implementation of (2.14) is shown in Figure 2-2. Only one filter $h[n]$ is used and each sample of the filter output is multiplied by its corresponding timing skew. The signals are multiplexed and added to $y[n]$ to attain a reconstruction of the input. The complexity is greatly reduced from the general equation but performance is sacrificed for large skews. In Section 2.6, we plot the accuracy of this approximation.

2.3 Matrix Formulation

In order to ease future analysis, we rewrite our sampling and reconstruction equations in a matrix form. Vectors are used to represent signals truncated to a block length of size N and matrices are used to represent filters. This simplification of notation will motivate the low-SNR reconstruction method, which is introduced in the next section. We reintroduce noise into the problem in order to accurately understand the sampling equations and reconstruction approximation.

2.3.1 Output Generation

To start, we write the distributed sampling output signal (2.3) with noise as

$$\boldsymbol{y} \quad = \quad \mathbf{F}\boldsymbol{x} + \boldsymbol{w} \tag{2.16}$$

with vectors representing the received signal, the uniformly sampled input signal, and the noise signal

$$\boldsymbol{y} \quad = \quad \begin{bmatrix} y[0] & y[1] & \dots & y[N-1] \end{bmatrix}^T \tag{2.17}$$

$$\boldsymbol{x} \quad = \quad \begin{bmatrix} x[0] & x[1] & \dots & x[N-1] \end{bmatrix}^T \tag{2.18}$$

$$\boldsymbol{w} \quad = \quad \begin{bmatrix} w[0] & w[1] & \dots & w[N-1] \end{bmatrix}^T \tag{2.19}$$

and matrix

$$\mathbf{F}_{k,l} \quad = \quad \mathrm{sinc}((k-l) + \tau_i/T_s) \quad i = k \pmod{M} \tag{2.20}$$

where $0 \leq k, l \leq N-1$. By using matrix notation to describe the sampling, we are performing an approximation by truncating the signal into blocks and approximating filter tails. We assume that these edge effects can be ignored by choosing a sufficiently large block size, although their treatment is more carefully handled during simulation.

Similar to the approximation (2.14), we can also compute the first order expansion of the sinc terms in the f_i from (2.4) around the point $\tau_i/T_s = 0$. This yields the approximation

$$y[n] \quad \approx \quad x[n] + \frac{\tau_i}{T_s}(h * x)[n] + w[n] \quad n \pmod{M} = i \tag{2.21}$$

where $h[n]$ is given in (2.15). In matrix form, (2.21) becomes

$$\boldsymbol{y} \quad = \quad \boldsymbol{x} + \sum_{i=1}^{M-1} \frac{\tau_i}{T_s}\mathbf{D}_i\mathbf{H}\boldsymbol{x} + \boldsymbol{w} \tag{2.22}$$

with $N \times N$ Toeplitz matrix $\mathbf{H}$ representing the h filter, i.e., $\mathbf{H}_{k,l} = h[k-l]$. The $\mathbf{D}_i$ are

$N \times N$ diagonal matrices that select the entries from the ith channel,

$$[\mathbf{D}_i]_{k,l} = \begin{cases} 1 & k = l, \quad i = k(\text{mod } M) \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

where $0 \leq k, l \leq N - 1$.

Simplifying (2.22) further,

$$\boldsymbol{y} \quad = \quad (\mathbf{I} + \mathbf{TH})\, \boldsymbol{x} + \boldsymbol{w} \tag{2.24}$$

where $\mathbf{T}$ is a diagonal matrix containing the skews

$$\mathbf{T}_{k,l} \quad = \quad \begin{cases} \tau_i & k = l, \ k - 1 \ (\text{mod } M) = i \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

and $\tau_0 = 0$ as stated previously. The matrix equations (2.16) and (2.24) yield the actual and approximate relationship, respectively, between $\boldsymbol{x}$ and $\boldsymbol{y}$.

2.3.2 Input Reconstruction

In a similar fashion, we can rewrite the reconstruction equation (2.10) as

$$\hat{\boldsymbol{x}} = \mathbf{G}\boldsymbol{y} \tag{2.26}$$

where for $l = r(\text{mod } M)$

$$\mathbf{G}_{k,l} = \gamma(kT_s) \frac{a_r(-1)^{(l-r)M}}{\pi(kT_s - (l-r)T_A - \tau_i')/T_A} \tag{2.27}$$

as defined in (2.10). Vector $\hat{\boldsymbol{x}}$ represents the estimated signal

$$\hat{\boldsymbol{x}} \quad = \quad \begin{bmatrix} \hat{x}[0] & \hat{x}[1] & \ldots & \hat{x}[N-1] \end{bmatrix}^T . \tag{2.28}$$

Note that in the absence of noise, the reconstruction equation will fully invert the nonuniform samples. Thus, ignoring the matrix edge effects, we find that $\mathbf{G} \approx \mathbf{F}^{-1}$.

The small timing skew approximation for reconstruction (2.14) yields matrix equations

$$\begin{aligned} \hat{\boldsymbol{x}} &= \boldsymbol{y} - \sum_{i=1}^{M-1} \frac{\tau_i}{T_s} \mathbf{D}_i \mathbf{H} \boldsymbol{y} && (2.29) \\ &= (\mathbf{I} - \mathbf{T}\mathbf{H})\, \boldsymbol{y}, && (2.30) \end{aligned}$$

where $\mathbf{D}_i$, $\mathbf{H}$, $\mathbf{T}$ are defined previously. The matrix equations (2.26) and (2.30) yield the actual and approximate relationship, respectively, between $\hat{\boldsymbol{x}}$ and $\boldsymbol{y}$.

By combining the sampling and reconstruction approximations (2.24) and (2.30), we find that

$$\begin{aligned} \hat{\boldsymbol{x}} &= (\mathbf{I} - \mathbf{T}\mathbf{H})\, (\mathbf{I} + \mathbf{T}\mathbf{H})\, \boldsymbol{x} + (\mathbf{I} - \mathbf{T}\mathbf{H})\, \boldsymbol{w} && (2.31) \\ &= (\mathbf{I} - \mathbf{T}\mathbf{H}\mathbf{T}\mathbf{H})\, \boldsymbol{x} + (\mathbf{I} - \mathbf{T}\mathbf{H})\, \boldsymbol{w}. && (2.32) \end{aligned}$$

As expected, in the absence of noise, the reconstruction approximation is accurate up to the second order in τ_i.

2.4 Noisy Signal Reconstruction

Until now, we have ignored the treatment of noise within the reconstruction. In cases where the input is oversampled, the reconstruction filter must be modified in order to reduce the noise effect on the estimate. Designing new filters to optimally handle the noise is intractable in the signal domain since it requires the inverse of more complicated time-varying filters. Instead, we use the matrix representation of the system to aid in the development of our algorithm. We now present a method for signal reconstruction from noisy periodic nonuniform samples when timing-skews are known. We begin by introducing a naive estimation method and then introduce a constrained least-squares formulation that allows for increased accuracy in the low-SNR setting.

2.4.1 Naive Least-Squares Estimation

The least-squares (LS) estimate for the input signal is given by

$$\hat{\boldsymbol{x}}_{\text{LS}} = \arg\min_{\boldsymbol{x}} \|hxb - \mathbf{F}(\boldsymbol{\tau})\boldsymbol{x}\|^2 \qquad (2.33)$$

where we now explicitly indicate the dependence of $\mathbf{F}$ on the vector of timing skews $\boldsymbol{\tau}$. Because the noise $\boldsymbol{w}$ is modeled as white Gaussian, the LS estimate is equivalent to the maximum-likelihood (ML) estimate. The solution is equal to

$$\hat{\boldsymbol{x}}_{\text{LS}} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\boldsymbol{y} = \mathbf{F}^{-1}\boldsymbol{y} \tag{2.34}$$

since $\mathbf{F}$ is invertible when the τ_i are sufficiently small.

This estimation method is equivalent to the inverse filterbank (2.10). The estimate performs well in high SNR situations. However, as noise power increases, it is clear to see that this estimate is suboptimal since it is not guaranteed to be bandlimited. A more accurate estimate of $\boldsymbol{x}$ can be produced by enforcing the bandlimited constraint on the estimator $\hat{\boldsymbol{x}}$. We label this as the filtered least-squares (FLS) estimator:

$$\hat{\boldsymbol{x}}_{\text{FLS}} = \mathbf{L}\mathbf{F}^{-1}\boldsymbol{y}. \tag{2.35}$$

where $\mathbf{L}$ is a matrix implementing a lowpass filter bandlimited to ω_c.

By plugging in for $\boldsymbol{y}$ in the estimator, we find

$$\begin{aligned} \hat{\boldsymbol{x}}_{\text{FLS}} &= \mathbf{L}\mathbf{F}^{-1}(\mathbf{F}\boldsymbol{x} + \boldsymbol{w}) && (2.36) \\ &= \boldsymbol{x} + \mathbf{L}\mathbf{F}^{-1}\boldsymbol{w}. && (2.37) \end{aligned}$$

Thus the error term equals $\mathbf{e}_{\text{FLS}} = \mathbf{L}\mathbf{F}^{-1}\boldsymbol{w}$.

2.4.2 Constrained Least-Squares Estimation

We develop a more accurate estimator by imposing the bandlimited requirement directly into the least-squares estimation. The constrained least-squares (CLS) optimization problem is given by

$$\hat{\boldsymbol{x}}_{\text{CLS}} = \arg\min_{\boldsymbol{x}\in\mathcal{S}} \|\boldsymbol{y} - \mathbf{F}(\boldsymbol{\tau})\boldsymbol{x}\|^2 \tag{2.38}$$

where $\mathcal{S} = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{R}^N,\ \mathbf{L}\boldsymbol{x} = \boldsymbol{x}\}$.

By introducing a secondary variable $\mathbf{z}$, where $\boldsymbol{x} = \mathbf{L}\mathbf{z}$, we can remove the constraint in the LS formulation

$$\begin{aligned}\hat{\boldsymbol{z}}_{\text{CLS}} &= \arg\min_{\mathbf{z}} \|\boldsymbol{y} - \mathbf{F}\mathbf{L}\mathbf{z}\|^2 && (2.39)\\ \hat{\boldsymbol{x}}_{\text{CLS}} &= \mathbf{L}\hat{\boldsymbol{z}}_{\text{CLS}}. && (2.40)\end{aligned}$$

First, we must compute $\hat{\boldsymbol{z}}_{\text{CLS}}$. Because the matrix $\mathbf{L}$ is singular (high frequency vectors lie in the nullspace), the product matrix $\mathbf{FL}$ is also singular; therefore it is not possible to take the inverse of this matrix in order to compute the maximum likelihood estimate $\hat{\boldsymbol{z}}_{\text{CLS}}$. Instead, we use the pseudoinverse

$$\hat{\boldsymbol{z}}_{\text{CLS}} = (\mathbf{FL})^{\dagger}\boldsymbol{y} \tag{2.41}$$

where $(\cdot)^{\dagger}$ is given by the Moore-Penrose pseudoinverse [40]

$$(\mathbf{FL})^{\dagger} = \lim_{\delta\to 0}((\mathbf{FL})^T(\mathbf{FL}) + \delta\mathbf{I})^{-1}(\mathbf{FL})^T. \tag{2.42}$$

The overall solution is given as

$$\hat{\boldsymbol{x}}_{\text{CLS}} = \mathbf{L}(\mathbf{FL})^{\dagger}\boldsymbol{y}. \tag{2.43}$$

To compute the CLS estimator error, we plug in for $\boldsymbol{y}$ and find

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{CLS}} &= \mathbf{L}(\mathbf{FL})^{\dagger}(\mathbf{F}\boldsymbol{x} + \boldsymbol{w}) && (2.44)\\ &= \boldsymbol{x} + (\mathbf{FL})^{\dagger}\boldsymbol{w}. && (2.45)\end{aligned}$$

The error term equals $\mathbf{e}_{\text{CLS}} = (\mathbf{FL})^{\dagger}\boldsymbol{w}$.

2.4.3 Reconstruction matrix operations

We now analyze the matrix operations in this estimator. The lowpass filter matrix can be implemented by a frequency sampling filter matrix, i.e.

$$\mathbf{L} = \mathbf{D}^{-1}\boldsymbol{\Sigma}_L\mathbf{D} \tag{2.46}$$

where $\mathbf{D}$ is the $N \times N$ DFT matrix, and $\mathbf{\Sigma}_L$ is the frequency response of the filter. Since $\mathbf{L}$ represents a lowpass filter with cutoff frequency ω_c, it is equivalent to having the $(N - k)$ eigenvalues that correspond to the low frequency eigenvectors equal to one and the k eigenvalues that correspond to high frequency eigenvectors equal to zero, where $k = \lfloor N(\pi - \omega_c)/\pi \rfloor$. The $N \times N$ eigenvalue matrix is given by

$$\mathbf{\Sigma}_L = \begin{bmatrix} \mathbf{I}_{N-k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2.47}$$

It is clear to see that the nullspace of $\mathbf{L}$ corresponds to linear combinations of high frequency vectors.

The pseudoinverse of $\mathbf{L}$ is equivalent to inverting all the non-zero eigenvalues. Because each of these eigenvalues is equal to one, their inverses are equal to one (and the zero eigenvalues remain zero). Thus we find that $\mathbf{L}^\dagger = \mathbf{L}$. Note that without the frequency sampling approximation (2.46), the matrix $\mathbf{L}$ may have very small non-zero eigenvalues. When the matrix is inverted, these eigenvalues become very large but their effects can be negated with a power constraint on $\hat{\boldsymbol{x}}$, [19].

Now, we analyze the properties of the product $\mathbf{FL}$. The singular value decomposition is given by

$$\mathbf{FL} = \mathbf{U}\mathbf{\Sigma}_{FL}\mathbf{V}^T \tag{2.48}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $N \times N$ orthonormal matrices. It is easy to see that because $\mathbf{F}$ is full rank, the nullspace $\mathcal{N}(\mathbf{FL}) = \mathcal{N}(\mathbf{L})$ and has rank $N - k$. Thus $\mathbf{\Sigma}_{FL}$ can be decomposed as

$$\mathbf{\Sigma}_{FL} = \begin{bmatrix} \mathbf{\Sigma}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.49}$$

where $\mathbf{\Sigma}_S$ is an $(N - k) \times (N - k)$ diagonal matrix.

The bottom k rows of $\mathbf{V}^T$ span the high frequency nullspace of $\mathbf{L}$, and the top rows span the low frequency space. Again, since $(\mathbf{FL})^\dagger$ inverts only the non-zero eigenvalues, we find that

$$(\mathbf{FL})^\dagger\mathbf{FL} = \mathbf{V}\mathbf{\Sigma}_L\mathbf{V}^T = \mathbf{L} \tag{2.50}$$

where the second equality holds because the nullspaces of $\mathbf{FL}$ and $\mathbf{L}$ are equal and the low frequency subspace has unity eigenvalues. This simplification does not imply anything about the product $\mathbf{FL}(\mathbf{FL})^{\dagger}$. For similar reasons, it is easy to see that $\mathbf{L}(\mathbf{FL})^{\dagger} = (\mathbf{FL})^{\dagger}$.

From the analysis above, we can better interpret the CLS estimator. First, the received signal is projected into the space of $\boldsymbol{\tau}$-spaced periodic nonuniform samples of any possible bandlimited signal, i.e. the space spanned by $\mathbf{FL}$. This produces the intermediate estimate $\mathbf{FL}(\mathbf{FL})^{\dagger}\boldsymbol{y}$. The nonuniform sampling is then inverted by applying $\mathbf{F}^{-1}$. Thus, the noise reduction occurs on the nonuniform samples before the signal reconstruction.

Fig. 2-3 shows a graphical representation of the CLS and FLS estimators. Convex set $\mathcal{B}$ represents signals bandlimited to ω_c and convex set $\mathcal{F}_{\boldsymbol{\tau}}$ represents signals that can be written as $\mathbf{F}s$, where $s \in \mathcal{B}$. The input $\boldsymbol{x}$ begins in $\mathcal{B}$. The periodic nonuniform sampling produces the signal $\mathbf{F}\boldsymbol{x}$ in $\mathcal{F}_{\boldsymbol{\tau}}$ and the noise perturbs the signal away from this set to $\mathbf{y} = \mathbf{F}\boldsymbol{x} + \boldsymbol{w}$.

As described above, the CLS estimator removes noise before reconstruction by projecting $\mathbf{y}$ into $\mathcal{F}_{\boldsymbol{\tau}}$ and then producing its compliment in $\mathcal{B}$. On the other hand, the FLS estimator removes noise after reconstruction by first inverting the nonuniform sampling (producing $\mathbf{F}^{-1}\mathbf{y}$) and then projecting into $\mathcal{B}$.

The CLS approach provides a more efficient method for noise reduction than the equivalent frequency-domain techniques for signal estimation, which are not as easily realizable in hardware. It also provides insight into optimal methods for treating noise when developing practical reconstruction filters. In the following section, we will show the performance improvement of the CLS estimator over the naive FLS method.

2.4.4 Analysis of Estimators

To compare the estimators, we analyze their bias, variance, and efficiency. From (2.37) and (2.45), it is clear that estimators FLS and CLS are unbiased, i.e., $E[\hat{\boldsymbol{x}}] = \boldsymbol{x}$. The unbiased property is due to the fact that the expectation of any linear combination of the noise $w[n]$ is zero.

The covariance matrices of the noise are given by

$$\begin{aligned}\Lambda_{\mathrm{FLS}} &= E[\mathbf{e}_{\mathrm{FLS}}\,\mathbf{e}_{\mathrm{FLS}}^T] = \sigma^2\mathbf{L}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{L}^T && (2.51)\\ \Lambda_{\mathrm{CLS}} &= E[\mathbf{e}_{\mathrm{CLS}}\,\mathbf{e}_{\mathrm{CLS}}^T] = \sigma^2(\mathbf{FL})^{\dagger}(\mathbf{FL})^{\dagger^T}. && (2.52)\end{aligned}$$

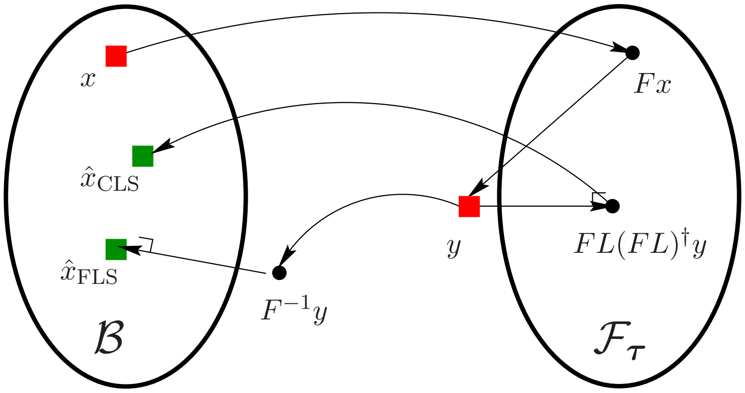

Figure 2-3: Graphical representation of the estimator operations. Set $\mathcal{B}$ represents the convex set of bandlimited signals. Set $\mathcal{F}_{\boldsymbol{\tau}}$ represents the convex set of signals that can be written as $\mathbf{F}s$, where $s \in \mathcal{B}$.

From these matrices, we can calculate the average variances

$$\sigma^2_{\text{FLS}} = \frac{1}{N}\sigma^2 \operatorname{tr}(\mathbf{L}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{L}^T) \quad (2.53)$$

$$\sigma^2_{\text{CLS}} = \frac{1}{N}\sigma^2 \operatorname{tr}((\mathbf{FL})^{\dagger}(\mathbf{FL})^{\dagger^T}). \quad (2.54)$$

Because comparing the errors is not analytically tractable, we numerically calculate them in Section 2.6 and verify that $\sigma^2_{\text{CLS}} \leq \sigma^2_{\text{FLS}}$, where equality holds if no noise is present in the system.

An estimator is efficient if its error variance achieves the Cramer-Rao bound (CRB). In [20], the CRB is redefined for constrained parameter estimation problems. For the problem of

$$\hat{\boldsymbol{x}}_{\text{ML}} = \underset{g(\boldsymbol{x})=0}{\arg\max}\, p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x}) \quad (2.55)$$

where the $g(\cdot)$ function may contain multiple nonlinear equations, the constrained CRB bound is given by

$$\boldsymbol{\Lambda}_{\hat{\mathbf{x}}} \geq \mathbf{J}^{-1} - \mathbf{J}^{-1}\mathbf{G}(\mathbf{G}^T\mathbf{J}^{-1}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{J}^{-1} \quad (2.56)$$

where

$$\mathbf{J} \quad = \quad -E\left[\nabla_{\boldsymbol{x}} \ln p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x}) \cdot \nabla_{\boldsymbol{x}}^{T} \ln p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})\right] \tag{2.57}$$

is the Fisher information matrix and

$$\mathbf{G} \quad = \quad \nabla_{\boldsymbol{x}} g^{T}(\boldsymbol{x}) \tag{2.58}$$

is the $N \times N$ gradient matrix of the constraints. In the time interleaved component setup, the constraints enforce that the estimate is bandlimited, $g(\boldsymbol{x}) = (\mathbf{I} - \mathbf{L})\boldsymbol{x}$. We find

$$\mathbf{J} = \mathbf{F}^{T}\mathbf{F}, \qquad \mathbf{G} = (\mathbf{I} - \mathbf{L})^{T} \tag{2.59}$$

yielding the bound on $\Lambda_{\hat{\boldsymbol{x}}}$ given in (2.56). Thus, we have the individual variance bound as:

$$\mathrm{var}[\hat{\boldsymbol{x}}_i(\boldsymbol{y})] \quad \geq \quad [\boldsymbol{\Lambda}_{\hat{\mathbf{x}}}]_{i,i}. \tag{2.60}$$

It is clear that the CLS estimate will achieve the CRB for the constrained ML problem because the constraints are linear. This property can also be seen because the CLS estimate is directly minimizing the Gaussian noise, whereas the FLS estimate indirectly minimizes the noise through a colored function of it. Therefore, by incorporating the bandlimited constraint into the LS estimation, we create an unbiased estimator $\hat{\boldsymbol{x}}_{\mathrm{CLS}}$ that performs better than the naive 'reconstruct and filter' estimator $\hat{\boldsymbol{x}}_{\mathrm{FLS}}$.

2.5 Iterative Methods of Reconstruction

Alternative techniques for reconstruction are now developed by using iterative methods. The use of iterations can allow for a deeper understanding of the signal geometries; however these methods are often too complex for practical implementation.

2.5.1 Frame Method

In [2], a reconstruction method is developed for an arbitrary nonuniform sampling pattern in the presence of noise. The method applies for any shift-invariant space, which is spanned by a generator ϕ,

$$x(t) = \sum_k x[k]\phi(t-k). \tag{2.61}$$

We apply this technique to the periodic nonuniform sampling of distributed sampling, where $\phi(t) = \text{sinc}(t/T_s)$. For the moment, we will ignore the system noise and later show stability when it is present.

In a general shift-invariant space, we can think of each sample $y[n]$ as a projection of the input $x(t)$ onto a shifted copy of a function $K(t)$,

$$y[n] = x(t_n) = \langle x, K_{t_n} \rangle = \int x(t) K_{t_n}(t) dt. \tag{2.62}$$

where $K_{t_n}(t) = K(t - t_n)$. For distributed periodic sampling,

$$K_{t_n}(t) = \text{sinc}((t - t_n)/Ts), \tag{2.63}$$

where $t_n = nT_s + \tau_n$ and $\tau_n = \tau_i$ for $i = n(\text{mod } M)$. As long as the Nyquist condition is satisfied, the K_{t_n} functions form a Hilbert frame for the bandlimited space $\mathcal{B}$, implying that there exists $\alpha > 0$ and $\beta < \infty$ such that

$$\alpha\|x\|^2 \leq \sum_n |\langle x, K_{t_n} \rangle|^2 \leq \beta\|x\|^2 \tag{2.64}$$

for all $x \in \mathcal{B}$. For any frame, there exists a dual frame $\tilde{K}$ such that the input $x(t)$ can be uniquely determined [9] by

$$x(t) = \sum_n \langle x, K_{t_n} \rangle \tilde{K}_{t_n}(t) = \sum_n y[n] \tilde{K}_{t_n}(t). \tag{2.65}$$

Thus, the dual frame $\tilde{K}$ acts as an interpolating function for the nonuniform samples. For our distributed sampling setup, both equations (2.8) and (2.10) provide solutions for the dual frame. However, without knowledge of the dual frame, there also exists an iterative method that provides an equivalent solution.

Starting from the initial samples $y^0[n]$, we can form a first estimate of the input $x(t)$ as

$$\hat{x}^1(t) = \sum_n \langle x, K_{t_n} \rangle K_{t_n}(t) = \sum_n y^0[n] K_{t_n}(t). \tag{2.66}$$

From this estimate of the input, we can resample at times $t = t_n$, forming the signal $y^1[n] = \hat{x}^1(t_n)$. At the next time step, we can form a new estimate based upon our initial estimate and our updated samples,

$$\hat{x}^2(t) = \hat{x}^1(t) - \lambda \sum_n y^1[n] K_{t_n}(t) + \lambda \hat{x}^0(t) \tag{2.67}$$

where λ is a step-size equal to $2/(\alpha + \beta)$ from (2.64). Recursively, we can iterate between forming a new input estimate and sampling at times t_n

$$\begin{aligned} \hat{x}^{(i+1)}(t) &= \hat{x}^i(t) - \lambda \sum_n y^i[n] K_{t_n}(t) + \lambda \hat{x}^0(t) & (2.68) \\ y^{(i+1)}[n] &= \hat{x}^{(i+1)}(t_n). & (2.69) \end{aligned}$$

The method converges to the input $\hat{x}^\infty(t) = x(t)$ and $y_\infty[n] = y[n]$. This iterative algorithm works by continually re-influencing the estimate of the input by the known values at times t_n. In the presence of noise, the method is stable and converges to a point within a noise ball around the true value [2].

2.5.2 Projection onto Convex Sets Method

An alternate iterative procedure can be derived in discrete-time from the projection onto convex sets (POCS) method. To setup this reconstruction method, we first quantize our timing skews by a factor of T_s/v, where v is the time quantization size. By doing so, we can rewrite the entire problem in the discrete domain. The input to the distributed sampling system is the oversampled signal $x_{ov}[n] = x(nT_s/v)$ and the ith component selects samples

$$y_i[n] = x_{ov}[nvM + (i-1)v + \delta_i], \tag{2.70}$$

where δ_i is τ_i quantized by T_s/v, i.e.,

$$\delta_i = \text{round}\left(\frac{\tau_i}{T_s/v}\right). \tag{2.71}$$

The output of all the components can now be written as a signal on this finely spaced grid by filling in the known samples and substituting zero for all unknown samples

$$y_{ov}[n] = \begin{cases} x_{ov}[n] & \text{for } n \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{2.72}$$

where $\mathcal{T}$ is the set of times of the component samples, i.e. $\mathcal{T} = \{kvM + iv + \delta_i\}$ for integer k and $0 \leq i < M$. In vector form, we take blocks of vN samples from the signals $x_{ov}[n]$ and $y_{ov}[n]$ to form

$$\mathbf{y}_{ov} = \mathbf{S}\boldsymbol{x}_{ov} \tag{2.73}$$

where $\mathbf{S}$ is a 0/1 diagonal matrix

$$\mathbf{S}_{j,} \quad = \begin{cases} 1 & j \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} . \tag{2.74}$$

Thus, the vector $\mathbf{y}_{ov}$ is zero except in the locations where the samples are known from the components. This received vector is the effective multiplexing of the outputs.

To estimate the input in this discretized time-skew setup, we use the method of POCS, which provides an iterative algorithm for finding the unique common point between two convex sets if one exists [37],[6]. Beginning at a point in one of the sets, the common point can be achieved by alternating orthogonal projections between sets. In the distributed system, the first convex set, $\mathcal{C}$, represents bandlimited vectors in $\mathcal{R}^{vN}$ with cutoff ω_c/v. The second set, $\mathcal{D}$, represents all vectors where the value of the vector in positions $\mathcal{T}$ are equal to value of the component outputs at these times. It is easy to verify that both of these sets are convex and $\boldsymbol{x}_{ov}$ is the unique common point.

The orthogonal projection from set $\mathcal{C}$ to $\mathcal{D}$ occurs by replacing the vector values at times $\mathcal{T}$ by the value of the component outputs at these times. The projection from $\mathcal{D}$ to $\mathcal{C}$ can be implemented by applying a lowpass filter $\mathbf{L}$ with cutoff ω_c/v. Our input $\mathbf{y}_{ov}$ begins in $\mathcal{D}$ and iterating between the projections will lead us to $\boldsymbol{x}_{ov}$ as follows:

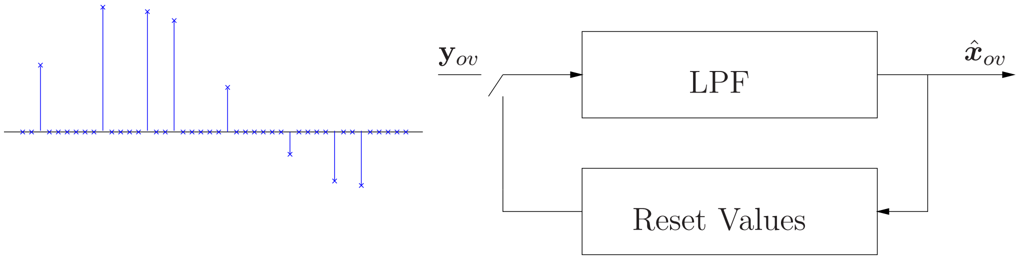

Figure 2-4: POCS Signal Reconstruction

Initialization:

$$\hat{\boldsymbol{x}}_D^0 = \mathbf{y}_{ov} \tag{2.75}$$

$$\tag{2.76}$$

Iterative Step:

$$\begin{aligned} \hat{\boldsymbol{x}}_C^i &= \mathbf{L}\hat{\boldsymbol{x}}_D^{i-1} && (2.77) \\ \hat{\boldsymbol{x}}_D^i &= (\mathbf{I}-\mathbf{S})\hat{\boldsymbol{x}}_C^i + \mathbf{y}_{ov}. && (2.78) \end{aligned}$$

The projections converge to

$$\begin{aligned} \hat{\boldsymbol{x}}_{ov}^{\infty} &= \sum_{i=0}^{\infty}((\mathbf{I}-\mathbf{S})\mathbf{L})^i \mathbf{y}_{ov} && (2.79) \\ &= (\mathbf{I}-(\mathbf{I}-\mathbf{S})\mathbf{L})^{-1}\mathbf{y}_{ov} && (2.80) \\ &= \boldsymbol{x}_{ov} && (2.81) \end{aligned}$$

because $\mathbf{L}\boldsymbol{x} = \mathbf{0}$ and $\mathbf{y}_{ov}$ can be written as $(\mathbf{I}-(\mathbf{I}-\mathbf{S})\mathbf{L})\boldsymbol{x}_{ov}$. Figure 2-4 shows the input and system diagram for the iterative process. The convergence speed is inversely proportional to the value of v. In this iterative process, the noise is removed within the lowpass filter projections.

2.6 Reconstruction Simulation Performance

In this section, we numerically evaluate the performance characteristics of the reconstruction methods. We present our system performance in the context of time-interleaved ADCs, where noise occurs due to quantization and performance is measured in terms of effective number of bits. We compare the effective number of bits for the reconstruction without calibration $y[n]$, to the reconstruction with calibration, i.e., $\hat{x}[n]$.

2.6.1 Effective Number of Bits (ENOB)

To measure effective bits, we first compute the signal-to-noise ratio (SNR) of the recovered signal $\hat{x}[n]$

$$\mathrm{SNR}_{\hat{x}} = 10 \log_{10} \frac{\sum x[n]^2}{\sum (x[n] - \hat{x}[n])^2}. \tag{2.82}$$

The uncalibrated signal SNR can be calculated in a similar fashion. The effective SNR of a signal is then related to the number of effective bits via $B = (\mathrm{SNR} - 1.76)/6.02$. This formula is based on the common assumption of a uniform distribution of input signal values.

In the tests, the converters quantize the input at 12-bit resolution, which generates the noise $w_i[n]$; the performance is measured through the increase in effective bits between the uncalibrated and calibrated signals. We will present the performance of each of the non-iterative reconstruction methods and discuss the tradeoffs between the amount of excess bandwidth, block size, number of converters and input SNR (effective bits).

For each simulation, we randomly select the $M-1$ converter timing skews independently using a uniform distribution. Increasing the range of this distribution yields a lower number of effective bits in the uncalibrated signal $y[n]$. Performance is measured for both small and large magnitude skews. The tests are performed using bandlimited Gaussian noise as input on 12-bit converters, with block sizes of 2^{15} samples and 33% extra oversampling. The gains are uniform among all converters.

Figures 2-5 and 2-6 show the relationship between the timing skew size and effective number of uncalibrated bits. Each point on the graph represents a different random selection of timing skews. The x-axis shows the total amount of skew ($\sum |\tau_i|/T$). In the case of a $M = 2$ converter system, the x-axis represents the magnitude of the skew $|\tau_1|/T$ and the y-axis represents the resulting number of effective bits in the uncalibrated signal $y[n]$. For $M = 16$, we compute the summation of the skew magnitudes and use this as measurement for the sampling inaccuracy of the entire system. As expected, for increasing amounts of timing skew, the error in the uncalibrated signal increases, thus decreasing the effective number of bits.

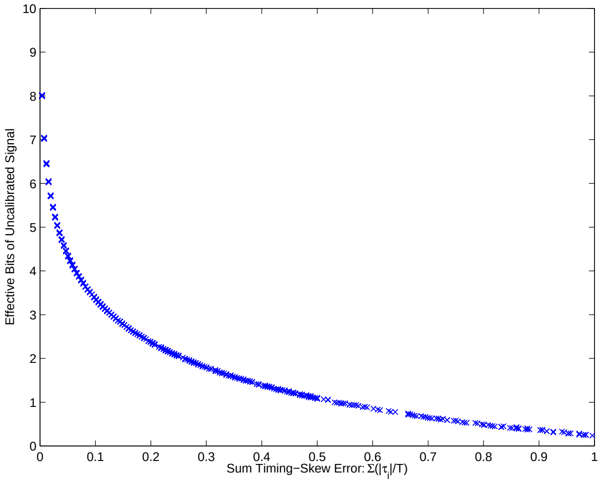

Figure 2-5: Effective number of uncalibrated bits vs. sum absolute timing skew $(\sum |\boldsymbol{\tau}|/T_s)$ for a 12-bit 2-ADC system. Each 'x' on the curve represents a unique test where the timing skew and input signal are chosen at random.

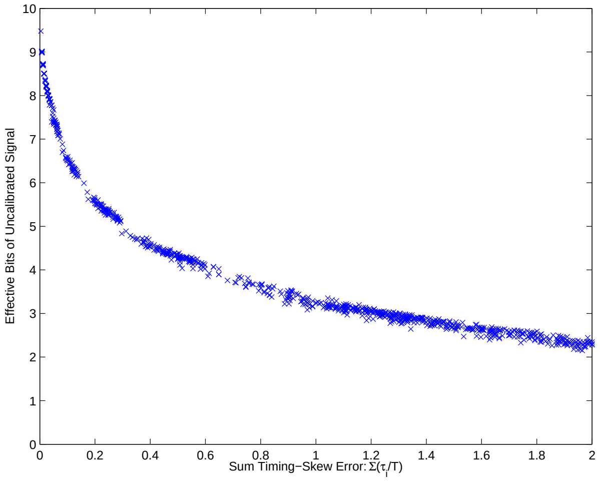

Figure 2-6: Effective number of uncalibrated bits vs. sum absolute timing skew $(\sum |\boldsymbol{\tau}|/T_s)$ for a 12-bit 16-ADC system. Each 'x' on the curve represents a unique test where the timing skews and input signal are chosen at random.

We can also calculate these curves analytically by computing the SNR

$$\text{Output Bits} = \frac{1}{6.02} 10 \log \left(\frac{\sigma_x^2}{\frac{1}{M} \sum \left(\frac{\tau_i}{T_s} \right)^2 \sigma_d^2 + \sigma_w^2} \right) \tag{2.83}$$

$$\approx k - \frac{20}{6.02} \log \left(\sum \frac{|\tau_i|}{T_s} \right) \tag{2.84}$$

where σ_x^2 is the power of the input signal and σ_d^2 is the power of the derivative signal $(\partial/\partial t)x(t)|_{t=nT_s}$.

2.6.2 Reconstruction Approximations

To reduce the complexity of reconstruction, (2.14) is used as a linearization of (2.8). Assuming knowledge of the timing skews, we compare the reconstruction accuracy of the these methods in Fig. 2-7 for the $M = 2$ system and Fig. 2-8 for the $M = 16$ system. Again, we simulate with random bandlimited Gaussian inputs.

In the figures, we first show the full reconstruction (2.8) and the no reconstruction (uncalibrated) curves. Due to out-of-band noise and filter truncation effects, the full calibration (dots) is limited below 12 bits for larger skews. The first order reconstruction approximation (x's) matches the full performance above six effective bits, and drops off linearly in calibrated bits for low numbers of uncalibrated bits.

By using higher order terms in the Taylor series, it is possible to generate increasingly accurate approximations. In Appendix A, we derive the second-order approximation and present a simple way to construct other orders. In Figs. 2-7 and 2-8, the second-order approximation (circles) performs better than the first-order by matching the ideal performance for greater than four uncalibrated bits. Similarly, the third-order (squares) matches for greater than two bits.

Note that these performance curves are signal dependent. As shown in the previous section, they are based upon the power of the input signal as well as the power of the derivative of the input. The class of bandlimited Gaussian inputs was chosen due to its robust properties of representing signals. Through simulations, we verified that the performance curves for other signals have similar behavior.

As we shall see, the performance of the mismatch estimation algorithms are limited by the reconstruction equations that are used. Therefore, it is important to understand

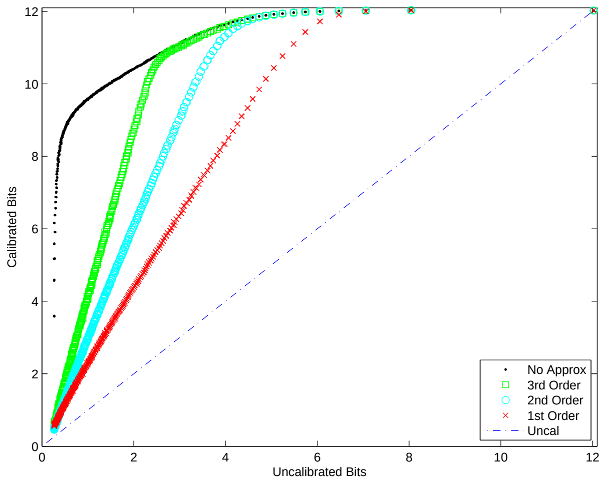

Figure 2-7: Comparison of ideal reconstruction equation to Taylor series approximations for a 12-Bit 2-ADC system with bandlimited Gaussian inputs. First-order (squares), second-order (circles), and third-order (x's) reconstruction performance curves are compared with full reconstruction and no reconstruction performance curves.

which approximation is sufficient in order to achieve the target performance for the signal estimate.

2.6.3 Noisy Estimators

We now analyze the performance of the FLS and CLS estimators. To do this, we compute the traces of the covariance matrices of the estimators. As expected, in all cases, the noise power is lower for the CLS estimator. The performance is measured in terms of the increase in effective bits of the CLS estimator over the FLS estimator. The results in Fig. 2-9 were simulated with a 12-bit 2-ADC system with block lengths of 512 samples. We present the performance for varying oversampling ratios and timing skew size.

The plot shows that for small amounts of timing skew, the CLS estimate only provides marginal improvements in effective bits. However, the benefits of the estimator are more visible for higher levels of timing skew and oversampling. This effective bit increase is independent of the starting number of effective bits because the signal power remains the same. Thus, the 0.1 bit increase obtained for the 2-ADC system with cutoff $\omega_c = 0.75\pi$ (33% oversampling) and 40% timing skew holds even for low-resolution converters.

Fig. 2-10 shows performance in a 16-ADC system for tests where the set of timing skews is chosen at random and the cutoff is $\omega_c = 0.75\pi$. The system timing error is measured

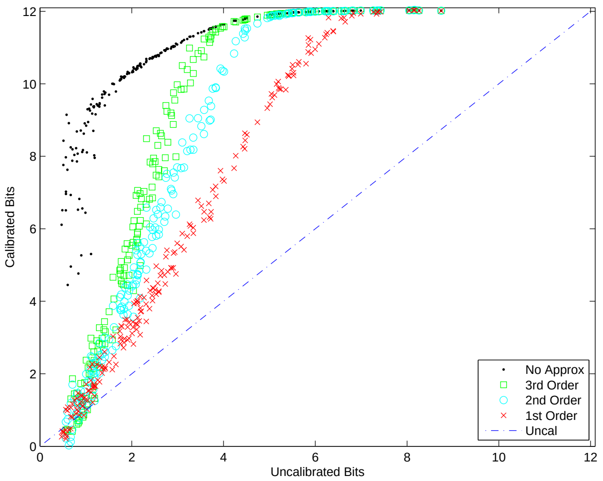

Figure 2-8: Comparison of ideal reconstruction equation to Taylor series approximations for a 12-Bit 16-ADC system with bandlimited Gaussian inputs. First-order (squares), second-order (circles), and third-order (x's) reconstruction performance curves are compared with full reconstruction and no reconstruction performance curves.

by the sum of the magnitudes of the timing skews $\sum |\tau_i|/T$. In this case, reconstruction for an average timing skew of $\sim$ 16% yields a 0.1 bit increase in resolution. As expected, the curves for the 2-ADC and 16-ADC system are approximately equal when the axis are normalized by the number of converters.

In this chapter, we have presented a variety of reconstruction schemes based upon the original nonuniform periodic reconstruction. First, we developed a first-order approximation and then developed a low-SNR method. From frame theory, we also presented iterative recovery methods. Finally, we demonstrated the performance difference between the methods through simulations.

We now turn to the problem of parameter estimation in systems where the timing skews are unknown. The reconstruction methods presented here will motivate efficient methods for system calibration.

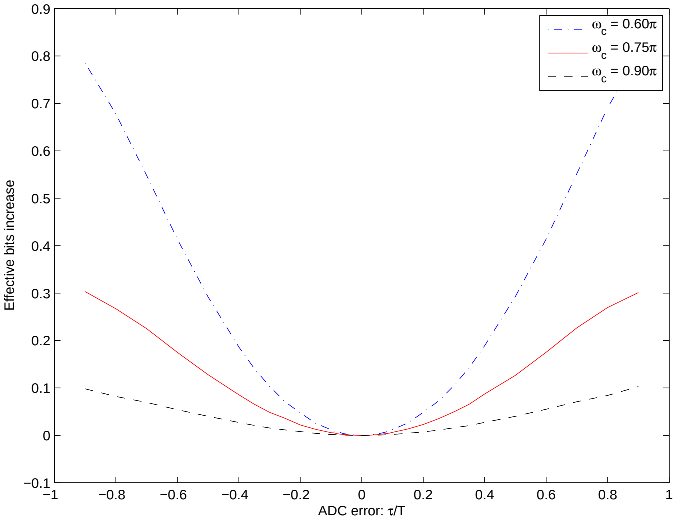

Figure 2-9: Effective bit increase of estimator CLS over FLS in a 2-ADC system for varying amounts of skew. Average performance is plotted for different cutoff frequencies (amounts of oversampling).

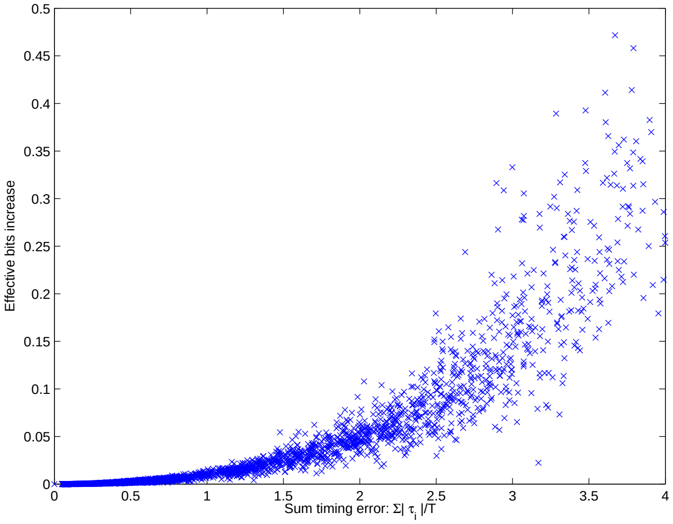

Figure 2-10: Effective bit increase of estimator CLS over FLS in a 16-ADC system with $\omega_c = 0.75\pi$. System timing skew is measured by $\sum |\tau_i|/T$. Each x on the plot represents a random set of skews.

Chapter 3

Small Mismatch Case

In many systems, a moderate level of timing control of the individual sampling components is present. Often system designers set the time spacing between components as accurately as possible. In TIADCs, the timing is controlled through a careful layout of components, while in other distributed systems the timing may be controlled by synchronization. Although this control may not be enough to provide perfectly uniform spacing between sampling components, it is often enough to ensure that the timing mismatch errors are small in magnitude. To ensure the overall system is within the performance specifications, it is necessary to calibrate and estimation the timing skews, as seen in Fig. 3-1.

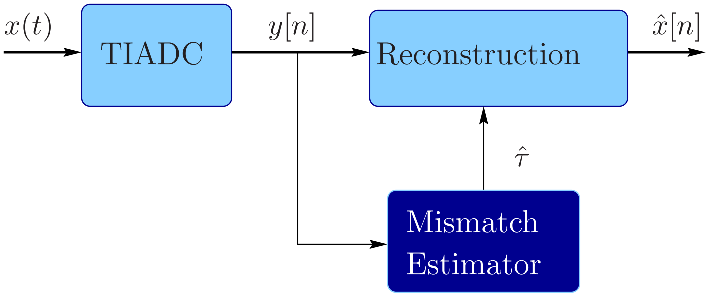

Figure 3-1: Parameter Mismatch Estimation

In this chapter, we focus on calibration methods for the small timing mismatch regime. To begin, we describe the small mismatch framework. Using the reconstruction approximation (2.30), we develop a least-squares method for estimating the unknown time-skews. We finish by generalizing to systems where nonuniform gains also exist and discuss practical adaptive implementations. Simulations are performed for various scenarios including systems where the skews lie outside the small mismatch range.

The system input $x(t)$ is modeled as a deterministic bandlimited signal as specified in Section 1.3. We model the output of the ith constituent channel as

$$y_i[n] = x(nMT_s + iT_s + \tau_i) + w_i[n] \tag{3.1}$$

where the τ_i model the unknown skews from uniform sampling. We also make the assumption that the timing skews are relatively small, e.g., not more than 20% of the overall sampling period. For ease of analysis, we assume the input is quantized with high-resolution, $w_i[n] \approx 0$; however the effects of quantization noise are considered within the subsequent simulations. The goal of the calibration algorithm is to estimate timing-skews τ_i from the output $y[n]$.

In our formulation, we develop an algorithm that seeks to estimate the unknown timing skews τ_i that provide a signal reconstruction with no out-of-band energy (in the absence of quantization noise and modeling error). Thus, we try to pick the bandlimited signal that best corresponds to the component outputs through nonuniform recurrent sampling.

3.1 Least-Squares Formulation

There are many methods and heuristics that we can use to estimate the input; one of the most natural is the maximum-likelihood estimate. Because the noise $\boldsymbol{w}$ is modeled as white Gaussian, the ML estimate of the nonrandom unknown parameters $x[n]$ and τ reduces to the least-squares problem

$$[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{\tau}}] \quad = \quad \arg\min_{\boldsymbol{x},\boldsymbol{\tau}} \|\boldsymbol{y} - \mathbf{F}_{\boldsymbol{\tau}}\boldsymbol{x}\|^2 \tag{3.2}$$

where we now explicitly indicate the dependence of $\mathbf{F}$ on $\boldsymbol{\tau}$. As it is stated, this optimization problem has multiple solutions that produce zero square-error, e.g. $\hat{\boldsymbol{\tau}} = \mathbf{0}$ and $\hat{\boldsymbol{x}} = \boldsymbol{y}$. Thus, we must specify a way to determine which solution is best. Rather than imposing any restriction on the timing skew estimate (such as a magnitude bound), we constrain the input signal to being bandlimited, as is assumed in the problem setup.

By introducing the bandlimited restriction on $\boldsymbol{x}$, we arrive at the constrained optimization problem

$$[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{\tau}}] \quad = \quad \arg\min_{\boldsymbol{x}\in\mathcal{B},\boldsymbol{\tau}} \|\boldsymbol{y} - \mathbf{F}_{\boldsymbol{\tau}}\boldsymbol{x}\|^2 \tag{3.3}$$

where

$$\mathcal{B} = \{\mathbf{z} \mid \mathbf{z} \in \mathcal{R}^N,\ \mathbf{Lz} = \mathbf{z}\}, \tag{3.4}$$

again $\mathbf{L}$ is the matrix implementing a lowpass filter bandlimited to ω_c. Because of noise issues, the actual values may not yield zero error. It is also possible that multiple minimizing solutions exist; however the noise affects the estimate smoothly and all solutions lie within a noise ball around the true values.

To solve this optimization problem, we can split the problem into sequential optimizations

$$[\hat{\boldsymbol{x}}\ \hat{\boldsymbol{\tau}}] \quad = \quad \arg\min_{\boldsymbol{\tau}} \arg\min_{\boldsymbol{x}\in\mathcal{B}} \|\boldsymbol{y} - \mathbf{F}_{\boldsymbol{\tau}}\boldsymbol{x}\|^2. \tag{3.5}$$

As shown in section 2.4.2, the solution to the inside optimization is the constrained least squares (CLS) estimate, thus

$$\hat{\boldsymbol{\tau}} \quad = \quad \arg\min_{\boldsymbol{\tau}} \|(\mathbf{I} - \mathbf{F}_{\boldsymbol{\tau}}\mathbf{L}(\mathbf{F}_{\boldsymbol{\tau}}\mathbf{L})^{\dagger})\boldsymbol{y}\|^2 \tag{3.6}$$

Because of the nonlinear dependence of $\boldsymbol{\tau}$ in $\mathbf{F}$ and in the pseudoinverse, the minimization is difficult to compute in closed form and an alternate approach must be taken in order to reduce the complexity.

3.2 Skew Mismatch Estimation via Least-Squares

Due to the difficult nature of the joint $\boldsymbol{\tau}$ and $\boldsymbol{x}$ optimization, we develop an indirect approach of estimating these parameters: first computing an estimate of the timings skews $\boldsymbol{\tau}$ and then using these estimates to estimate $\boldsymbol{x}$ through one of the methods presented in Chapter 2. It is clear to see that for accurate reconstruction, the estimated signal must be bandlimited,

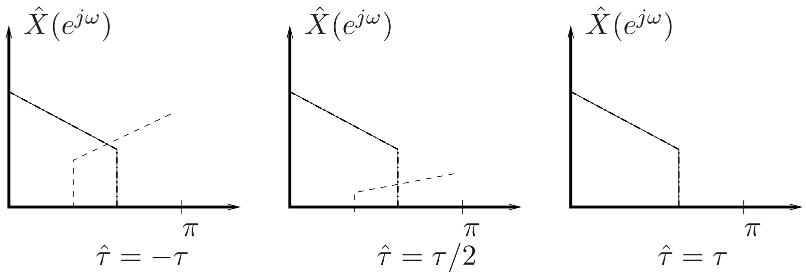

Figure 3-2: Reconstruction aliasing shown for different skew estimates. Out-of-band energy goes to zero iff the estimates are equal to the true skew values.

i.e.,

$$\hat{\boldsymbol{x}} = \mathbf{L}\hat{\boldsymbol{x}} \tag{3.7}$$

because any power in the high frequency directions is orthogonal to the input signal content.

For a given skew estimate $\hat{\boldsymbol{\tau}}$, the basic input estimator is given as

$$\hat{\boldsymbol{x}} = \mathbf{F}^{-1}(\hat{\boldsymbol{\tau}})\boldsymbol{y},. \tag{3.8}$$

The bandlimited condition implies that for an accurate of choice of timing skews $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}$

$$(\mathbf{L} - \mathbf{I})\mathbf{F}^{-1}(\hat{\boldsymbol{\tau}})\boldsymbol{y} = 0 \tag{3.9}$$

It can be shown that all other choices of $\hat{\boldsymbol{\tau}} \neq \boldsymbol{\tau}$ produce estimates $\hat{\boldsymbol{x}}$ that are not bandlimited, as seen in Fig. 3-2 and proved in Appendix B. In the absence of noise, the timing skews in equation (3.9) can be computed by the linear equations attained by using the approximation $\mathbf{F}^{-1}(\boldsymbol{\tau}) = (\mathbf{I} - \mathbf{TH})$.

In practice, no solution exists due to modeling error in the approximation (2.14) and quantization error; therefore, the optimization is formulated as a least-squares problem that computes the timing skews $\hat{\boldsymbol{\tau}}$ that minimize the out-of-band energy in the input estimate $\hat{\boldsymbol{x}}$

$$\hat{\boldsymbol{\tau}} = \arg\min_{\boldsymbol{\tau}} \|(\mathbf{L} - \mathbf{I})\mathbf{F}^{-1}(\hat{\boldsymbol{\tau}})\boldsymbol{y}\|^2. \tag{3.10}$$

We have reduced the parameter space of the joint minimization (3.3) by removing the need to estimate $\boldsymbol{x}$ directly. Using the approximation (2.30), our least-squares minimization becomes

$$\hat{\boldsymbol{\tau}} = \arg\min_{\boldsymbol{\tau}} \|(\mathbf{L}-\mathbf{I})(\mathbf{I}-\mathbf{T}\mathbf{H})\boldsymbol{y}\|^2 \tag{3.11}$$

where the $\mathbf{T}$ matrix is the only term depending on the timing skews $\boldsymbol{\tau}$. Rewriting this minimization directly in terms of $\boldsymbol{\tau}$, we find

$$\hat{\boldsymbol{\tau}} \quad = \quad \arg\min_{\boldsymbol{\tau}} \|\gamma - \mathbf{R}\boldsymbol{\tau}\| \tag{3.12}$$

where

$$\mathbf{R} = \begin{bmatrix} | & | & & | \\ \mathsf{r}_1 & \mathsf{r}_2 & \dots & \mathsf{r}_{M-1} \\ | & | & & | \end{bmatrix}, \quad \mathsf{r}_i = (\mathbf{L}-\mathbf{I})\mathbf{D}_i\mathbf{H}\boldsymbol{y} \tag{3.13}$$

and

$$\gamma = (\mathbf{L}-\mathbf{I})\boldsymbol{y}. \tag{3.14}$$

The solution to the over-constrained least-squares estimation problem is given by

$$\hat{\boldsymbol{\tau}} = \left(\mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{R}^T\gamma. \tag{3.15}$$

where the inversion of $\mathbf{R}^T\mathbf{R}$ is possible because for $N \gg M$ and nontrivial $\boldsymbol{y}$, the matrix $\mathbf{R}$ has full column rank. Thus, with $O(M^2N)$ complexity, the optimal solution $\hat{\boldsymbol{\tau}}$ can be computed. Uniform samples of the input signal can then be recovered from the timing skew estimates via any nonuniform periodic reconstruction method from Chapter 2.

3.2.1 Geometric Picture

There are multiple geometric ways to understand the operations of the LS estimation. One interpretation of the algorithm is that the least-squares method computes the signal in the convex set of signals $\hat{x}[n;\hat{\boldsymbol{\tau}}]$ spanned by $\hat{\boldsymbol{\tau}}$ that is closest to the convex set of signals bandlimited to ω_c. When we introduce back the noise in the system, the estimate deviates from the true value but the estimate error changes in a stable manner with respect to the

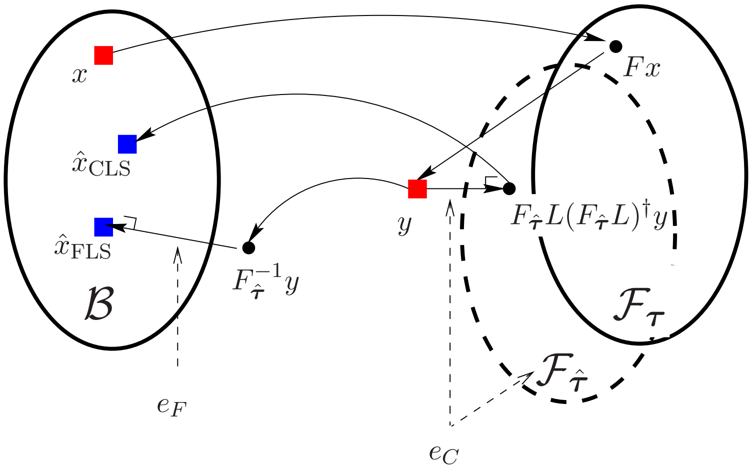
x
$\hat{x}_{\text{CLS}}$
$\hat{x}_{\text{FLS}}$
$\mathcal{B}$
$F_{\hat{\boldsymbol{\tau}}}^{-1}y$
y
$F_{\hat{\boldsymbol{\tau}}}L(F_{\hat{\boldsymbol{\tau}}}L)^{\dagger}y$
Fx
$\mathcal{F}_{\boldsymbol{\tau}}$
$\mathcal{F}_{\hat{\boldsymbol{\tau}}}$
e_F
e_C

Figure 3-3: Graphical representation of the timing skew and signal estimation operations. Set $\mathcal{B}$ represents the convex set of bandlimited signals. Set $\mathcal{F}_{\boldsymbol{\tau}}$ represents the convex set of signals that can be written as $\mathbf{F}s$, where $s \in \mathcal{B}$. Timing skew estimates can be derived from the values that minimize e_F and e_C respectively.

noise power. Similar to the POCS algorithm in Section 2.5, it is possible to implement this in a sequential projections manner.

In the second interpretation, we examine the calibration algorithm in the context of Fig. 3-3. The skew estimator developed here seeks to find the timing skews that minimize the amount of out-of-band energy in the reconstructed signal. This optimization is equivalent to choosing $\boldsymbol{\tau}$ to the minimize the distance between $\mathbf{F}_{\hat{\boldsymbol{\tau}}}^{-1}\mathbf{y}$ and $\mathbf{L}\mathbf{F}_{\hat{\boldsymbol{\tau}}}^{-1}\mathbf{y}$, shown as vector e_F in Fig. 3-3.

An alternate method for timing skew estimation can be developed by picking the skew values that minimize the distance between $\mathbf{y}$ and its projection into set $\mathcal{F}_{\hat{\boldsymbol{\tau}}}$, i.e. $\mathbf{F}_{\hat{\boldsymbol{\tau}}}\mathbf{L}(\mathbf{F}_{\hat{\boldsymbol{\tau}}}\mathbf{L})^{\dagger}\mathbf{y}$. This distance is shown as vector e_C in the figure. This skew estimation is equivalent to the joint least-squares estimate in (3.6). Although this method is expected to have better performance because it directly minimizes the noise, the difficulty of implementing it makes it less desirable.

Finally, we discuss two extensions to basic least-squares procedure. First we develop an iterative method for computing increasingly accurate estimates; after which we present a method for incorporating unknown component gains into the estimation procedure.

3.2.2 Relinearization

For values of τ_i/T_s that are not sufficiently close to zero, the approximation given by (2.14) may only provide a coarse reconstruction of the original signal because it relies on a Taylor series expansion of (2.10) around $\boldsymbol{\tau} = \mathbf{0}$. In this section, we present an iterative method for improving the accuracy of the initial least-squares estimate. Similar to Newton's method, we perform successive approximations by first computing the least-squares estimate $\hat{\boldsymbol{\tau}}$ by (3.15) and then computing the first order Taylor series approximation of $x[n]$ around the updated point $\boldsymbol{\tau} = \hat{\boldsymbol{\tau}}$

$$\hat{x}[n] \quad \approx \quad x[n]\Big|_{\boldsymbol{\tau}=\hat{\boldsymbol{\tau}}} + \sum_{i=1}^{M-1} \frac{\partial x[n]}{\partial \tau_i}\Big|_{\boldsymbol{\tau}=\hat{\boldsymbol{\tau}}}(\tau_i - \hat{\boldsymbol{\tau}}). \tag{3.16}$$

as shown in Appendix A.

From this updated reconstruction formula, it is possible to formulate a new least-squares problem whose solution is a more accurate estimate of $\boldsymbol{\tau}$. With increasingly accurate estimates $\hat{\boldsymbol{\tau}}$, the local approximation of $x[n]$ can improve and allow for better estimation. Simulation in subsequent sections demonstrate this performance increase. An alternate approach to achieving increasingly accurate estimates is to incorporate a higher order Taylor series expansion.This technique has been explored for a similar problem in [33].

3.2.3 Gain Mismatch Calibration

In the general calibration of distributed sampling systems, nonuniform gains can also exist among the constituent components. In this setup, the output of the ith component is modeled as

$$y_i[n] = g_i x(nMT_s + iT_s + \tau_i) + w_i[n]. \tag{3.17}$$

where the g_i are unknown gains. Although the gains vary among the components, we assume that each gain is within 10% of unity. For high resolution components, one can conveniently compensate for the system gains without excessive noise enhancement by multiplying each component output $y_i[n]$ by $1/g_i$. Without loss of generality, we set $g_0 = 1$.

By folding the gain recovery into the reconstruction equation (2.10), we can compute the Taylor series approximation around the point $\boldsymbol{\tau} = \mathbf{0}, \mathbf{g} = \mathbf{1}$, where $\mathbf{g} = [g_1 \;\; g_2 \;\; \dots \;\; g_{M-1}]$

and $\mathbf{0}$, $\mathbf{1}$ are vectors of zeros and ones respectively, shown in Appendix A. From this first order approximation, we can setup a similar least squares problem that now includes gains in the vector of unknown parameters

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\gamma} - \mathbf{R}\boldsymbol{\theta}\| \tag{3.18}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{1/g} \end{bmatrix} \tag{3.19}$$

$$\mathbf{R} = \begin{bmatrix} | & & | & | & & | \\ \mathbf{r}_1 & \dots & \mathbf{r}_{M-1} & \mathbf{s}_1 & \dots & \mathbf{s}_{M-1} \\ | & & | & | & & | \end{bmatrix} \tag{3.20}$$

$$\mathbf{r}_i = (\mathbf{L} - \mathbf{I})\mathbf{D}_i\mathbf{H}\boldsymbol{y} \tag{3.21}$$

$$\mathbf{s}_i = -(\mathbf{L} - \mathbf{I})\mathbf{D}_i\boldsymbol{y} \tag{3.22}$$

$$\boldsymbol{\gamma} = (\mathbf{L} - \mathbf{I})\mathbf{D}_0\boldsymbol{y} \tag{3.23}$$

and for notational compactness $\mathbf{1/g}$ is used to denote a vector containing the inverses of the gains. Again, relinearization techniques around $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\tau}})$ allow for an increase in performance. The performance of the least-squares method with both unknown gains and timing skews is presented in Section 3.3.

3.2.4 Adaptive Filter Implementation

In the development of the least-squares calibration algorithms, it was convenient to use vectors to represent time-domain signals and matrices to represent filtering operations. This notation allows for greater insight into the methods developed for parameter estimation and signal reconstruction.

In practice, the block implementation of such a system may be cumbersome for efficient implementation as it requires the multiplication of large matrices. As mentioned earlier, FIR filters may be used to replace the matrix operations, such as those discussed in [44], [23], [29],[46]. For example, reconstruction (2.30) can be simplified through the FIR derivative filter presented in (2.14). We now use these filter implementations to develop the least-squares calibration into a sequential algorithm via an adaptive filter structure.

The block implementation of the timing skew estimate (3.15) takes $O(M^2N)$ complexity and a large amount of memory. We develop a recursive least-squares (RLS) implementation that distributes the computation over time and sequentially adapts to shifts in parameters. To start, we rewrite the matrix $\mathbf{R}$ in terms of its rows $\mathbf{u}$, where each row now represents signal values at a certain time step

$$\begin{aligned} \mathbf{R} &= \left[\begin{array}{cccc} \mathbf{u}[0]^T & \mathbf{u}[1]^T & \ldots & \mathbf{u}[N-1]^T \end{array} \right]^T & (3.24) \\ \mathbf{u}[n] &= \left[\begin{array}{cccc} r_1[n] & r_2[n] & \ldots & r_{M-1}[n] \end{array} \right] & (3.25) \end{aligned}$$

where $r_i[n]$ is the nth element of $\mathbf{r}_i$. The values can also be computed by the filtering operations

$$r_i[n] = \sum_{m (\text{mod } M) = i} p[n-m]s[m] \tag{3.26}$$

where the filter p implements the high-pass filter $(\mathbf{I} - \mathbf{L})$ and $s[n] = (h * y)[n]$. Also, the vector $\boldsymbol{\gamma}$ in (3.14) can be written as individual elements $\boldsymbol{\gamma}[n]$, where $\boldsymbol{\gamma}[n] = (p * y)[n]$.

In the recursive setup [22], at time step n, we use $\mathbf{u}[n]$ and $\boldsymbol{\gamma}[n]$ in order to update our filter taps (timing skews). We define a forgetting factor λ such that the weight of the $(n-i)$th sample is λ^{-i}, where $\lambda = 1$ yields the traditional least squares estimate. We also define a regularization constant δ to stabilize the solution. The recursive solution to the least-squares estimation (3.15) is given by:

RLS Initialization

$$\begin{aligned} \hat{\boldsymbol{\tau}}[0] &= \mathbf{0} & (3.27) \\ \mathbf{P}[0] &= \delta \mathbf{I} & (3.28) \end{aligned}$$

RLS Update

$$\begin{aligned}
\boldsymbol{\pi}[n] &= \mathbf{P}[n-1]\mathbf{u}[n] && (3.29)\\
\mathbf{k}[n] &= \frac{\boldsymbol{\pi}[n]}{\lambda + \mathbf{u}^T[n]\boldsymbol{\pi}[n]} && (3.30)\\
\xi[n] &= \boldsymbol{\gamma}[n] - \hat{\boldsymbol{\tau}}^T[n-1]\mathbf{u}[n] && (3.31)\\
\hat{\boldsymbol{\tau}}[n] &= \hat{\boldsymbol{\tau}}[n-1] + \mathbf{k}[n]\xi[n] && (3.32)\\
\mathbf{P}[n] &= \lambda^{-1}\mathbf{P}[n-1] - \lambda^{-1}\mathbf{k}[n]\mathbf{u}^T[n]\mathbf{P}[n-1] && (3.33)
\end{aligned}$$

where $\mathbf{P}$ represents an estimate of the input $\mathbf{u}$ auto-correlation, which aids in whitening.

The adaptive filter both spreads the computational complexity across time and handles time skews that vary with time. Instead of $O(M^2N)$ complexity every N samples, we now have $O(M^2)$ complexity every sample. Using $\delta = 0.001$ and $\lambda = 0.95$, we find that the performance of the block algorithm is matched after the steady state is reached, as shown in Sec. 3.3.6.

In practice, the RLS algorithm above is often replaced by the lower complexity least mean-squares (LMS) algorithm. In LMS, no estimation of the covariance matrix $\mathbf{P}$ is made. Instead the $\hat{\boldsymbol{\tau}}$ estimates are updated by the gradient direction that minimizes the $\xi[n]$ error (the direction to minimize the out-of-band energy):

LMS Update

$$\begin{aligned}
\xi[n] &= \boldsymbol{\gamma}[n] - \hat{\boldsymbol{\tau}}^T[n-1]\mathbf{u}[n] && (3.34)\\
\hat{\boldsymbol{\tau}}[n] &= \hat{\boldsymbol{\tau}}[n-1] + \mu\mathbf{u}[n]\xi[n] && (3.35)
\end{aligned}$$

where μ denotes the step-size parameter of the update. Figure 3-4 shows a system implementation of the reconstruction algorithm.

While the convergence rate of LMS is generally slower than RLS, the LMS algorithm only requires $O(M)$ complexity per sample, making it more attractive for system implementations. In simulations performed with a sufficiently small step-size, the algorithm converged to the same skew estimates as the RLS method. Table 3.1 shows a comparison of the sequential algorithms.

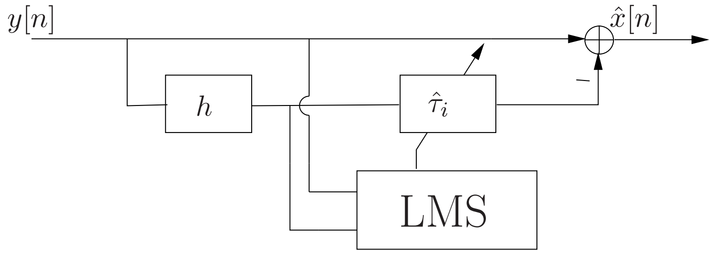

Figure 3-4: System implementation of the LMS adaptive filter for timing skew estimation. The 'LMS' box operates on $y[n]$ and $(h * y)[n]$ to produce estimates of the skews $\boldsymbol{\tau}$

Method	Computation/ Block	Computation/ Sample	Convergence Speed
Block LS	$O(M^2N)$		
RLS	$O(M^2N)$	$O(M^2)$	Moderate
LMS	$O(MN)$	$O(M)$	Slow

Table 3.1: Speed and compolexity comparison of sequential least-squares methods to block least-squares method

3.2.5 Other Signal Classes

In Section 1.3.2, we discussed the valid classes of input signals that can be used to calibrate a distributed sampling system. The main requirement imposed is that the system must have knowledge of a spectrum band where discrete time aliasing occurs with nonuniform sampling and where the input spectrum is also zero. For signals that are non-zero in $0 \leq |\omega| < \omega_c$, aliasing exists in the $\omega_c < |\omega| < \pi$ band and the current lowpass filter $\mathbf{L}$ with cutoff ω_c used in calibration is sufficient. If the aliasing exists in a different known band, $\mathcal{S} \subset [0, \pi)$, the lowpass filter matrix $\mathbf{L}$, which imposes the bandlimited constraint, can be replaced with a bandpass filter (notching out the $\mathcal{S}$ band). This new filter imposes the constraint that the reconstructed signal has minimal energy content in the known aliasing band.

For completely blind calibration, the input spectrum may be unknown. In this case, it may be possible to determine the aliasing band based off the spectrum of the output. For input signals with almost constant power across frequencies, the output frequency bands with lower power contain the aliasing and be treated as out-of-band spectrum.

For certain non-bandlimited signals, the same general method can be applied by replacing the sampling matrix and signal class. The method presented exploits the convex structure of the input class as well as the fact that the sampling frame provides an overredudant basis. If these two properties are satisfied, solutions can be obtained through straightforward changes in the formula. One such input class is the shift-invariant spaces described in Section 2.5.

3.3 Performance Results

To measure the performance of the calibration algorithms, we simulate random bandlimited Gaussian inputs, as described in Section 2.6: using 2^{15} samples/block, 33% oversampling, 12-bit converters, and unity gains. After the timing skews are estimated, the final estimate $\hat{x}[n]$ is produced using the time-skew estimates in the ideal reconstruction formula (2.10) for additional precision.

3.3.1 LS and ReLin Calibration

In Figures 3-5 and 3-6, we plot the performance of the calibration methods by showing the effective bits of the calibrated signal vs. uncalibrated signal for systems with 2 ADCs and 16 ADCs respectively. For each trial, a random bandlimited Gaussian signal is generated along with a random set of timing skews. The trial signal is calibrated (skews and signal estimated) and performance is plotted with a point on each of the curves.

The full calibration upper bound (squares) shows recovery performance in each trial when the true skew values are used in the ideal reconstruction formula; note that out-of-band energy in the input, quantization noise, and numerical precision errors limit the accuracy below 12 effective bits for large skews. The lower bound (dashed line) plots the performance when no recovery is performed, i.e. effective output bits equals effective input bits. Recovery performance is shown for the least-squares 'LS' estimate (circle); and for the 2-ADC system, performance is also shown for multiple iterations of the relinearization 'ReLin' method (x). The number of iterations is approximately equal to $100\tau_1/T$.

As the timing skew decreases (increasing number of uncalibrated bits), the recovery algorithm yields a higher number of effective output bits. The performance lines intersect at the point $(0, 0)$ and exhibit a linear behavior until they intersect the upper bound in performance, at which point performance is capped. Using the single least-squares 'LS' calibration, any input above four effective bits achieve the upper bound. As the number of calibration iterations increases, the slope of the 'ReLin' calibration curve increases and the curve moves closer to the full calibration curve. With many iterations of the relinearization

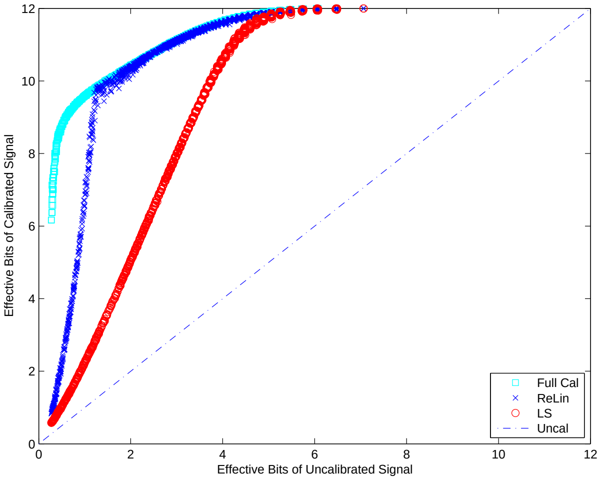

Figure 3-5: Effective number of calibrated bits vs. uncalibrated bits for a 12-bit 2-ADC system with unknown timing skews. Performance is measured using the least-squares (LS) estimate and the relinearization (ReLin) estimate for a bandlimited Gaussian input oversampled by 33%.

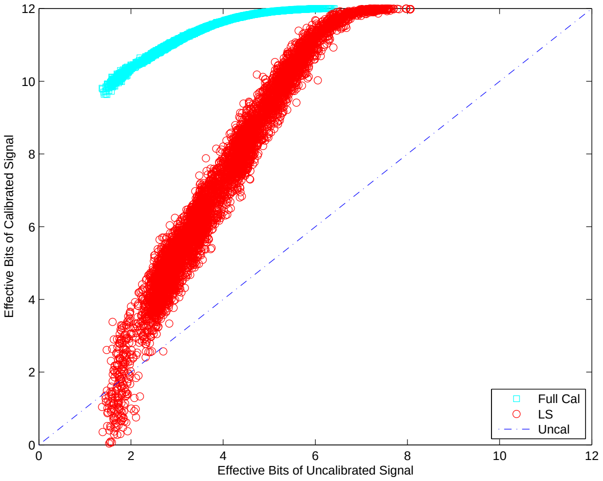

Figure 3-6: Effective number of calibrated bits vs. uncalibrated bits for a 12-bit 16-ADC system with unknown timing skews. Performance is measured using the least-squares (LS) estimate for a bandlimited Gaussian input oversampled by 33%.

method, large timing skews producing signals as low as 1.8 effective bits can achieve the same bound.

The LS performance is limited by how well the reconstruction approximation $(\mathbf{I} - \mathbf{TH})$ performs in comparison to $\mathbf{F}^{-1}$. The combination of 'Least-squares skew estimate'+'Ideal reconstruction using $\hat{\boldsymbol{\tau}}$' employed here matches the performance curve of the 'First-order reconstruction using true $\boldsymbol{\tau}$' seen in Fig. 2-7. Thus, the calibration only produces an estimate as good as the reconstruction approximation can yield. Alternate tests verify this limitation by showing that second order implementation of the LS skew estimate, shown in Appendix A,

$$\hat{\boldsymbol{\tau}} = \arg\min_{\boldsymbol{\tau}} \|(\mathbf{L} - \mathbf{I})(\mathbf{I} - \mathbf{TH} + \mathbf{T}^2\mathbf{H}_2 - \mathbf{THTH})\boldsymbol{y}\|^2 \tag{3.36}$$

combined with the ideal reconstruction formula has performance that matches the second-order reconstruction in Fig. 2-7.

In the 16-ADC system, the performance of the single least-squares calibration matches the full calibration curve for input signals with greater than 5.5 effective bits. Like the 2-ADC case, performance is limited by the accuracy of the approximation, which is shown in Fig. 2-8. Although the relinearization performance is withheld to allow for clarity in the graph, tests show that for uncalibrated signals with more than 4.5 effective bits, relinearization can achieve the upper bound in performance. For the 16-ADC system with five effective uncalibrated bits, the relinearization method achieves an output performance of 11.5 bits, which is 2.5 bits better than the least-squares method. The relinearization performance drops off sharply as we decrease the number of effective input bits below 4.5. In this range, some tests even showed a 'ReLin' performance lower than 'LS', possibly due to a nonlinear structure around the point of approximation.

The input class of bandlimited Gaussian signals is used due to its robust properties. The class provides the minimum deviation from all possible classes of bandlimited inputs by maximizing the signal entropy for a fixed power. However, to test other signals, calibration was also performed using single tone inputs. The trials showed similar performance to the wideband signals; however, tests required that the out-of-band spectrum be specified precisely based on the input tone frequency.

3.3.2 Cutoff Frequency Considerations

In the time-interleaved setup, sampling generally occurs after the use of an anti-aliasing filter in the analog domain. This filter ensures that high frequencies are not aliased during sampling. Although we model the uniformly sampled signal $x[n]$ to have cutoff frequency ω_c, we cannot guarantee that the signal has non-zero spectral content for $\omega < \omega_c$ or that the spectral content goes to zero for $\omega > \omega_c$, which may lead to degraded performance. If the spectrum is known, this issue can often be resolved by changing the cutoff frequency of the lowpass filter used in skew estimation. However, the spectrum of the input is not always known *a priori* and we examine this case further through simulations.

We first tested the effects of zero content in a range $\omega_s < \omega < \omega_c$ by using lowpass filters whose cutoff frequencies were $(5\%, 10\%, 20\%)$ larger than the maximum frequency of the signal. The performance for the 10% and 20% cases are presented in Fig. 3-7. In each of the cases, the LS performance remained the same as the base case, while the ReLin performance decreased with larger filter cutoff frequencies. The ReLin degradation is clear for fewer than four uncalibrated bits. For two uncalibrated bits, the $(5\%, 10\%, 20\%)$ cases yielded a ReLin calibration decrease of $(0.4, 1.8, 4.0)$ bits from the base case.

We also tested the effects of non-zero signal in $\omega > \omega_c$ by using inputs that have varying amounts of out-of-band signal content. As a general model for the out-of-band content, we added a white Gaussian signal into this spectrum ($\omega_c < \omega$) with a power $(20, 40, 60)$dB lower than the main band ($\omega < \omega_c$). The -20dB and -40dB performance curves are presented in Fig. 3-8. At -60dB, the out-of-band energy did not affect calibration performance, while -40dB, performance only degraded slightly. The -20dB case yielded a significant drop: for $(3, 5, 7)$ uncalibrated bits, LS calibration dropped by approximately $(2, 6, 5)$ bits respectively.

The results of both sets of filtering tests are as expected: out-of-band signal content can decrease performance by obscuring the aliased content in this band and larger than necessary filter cutoffs allow for a smaller region for aliasing to be detected, which more visibly degrades performance during relinearization.

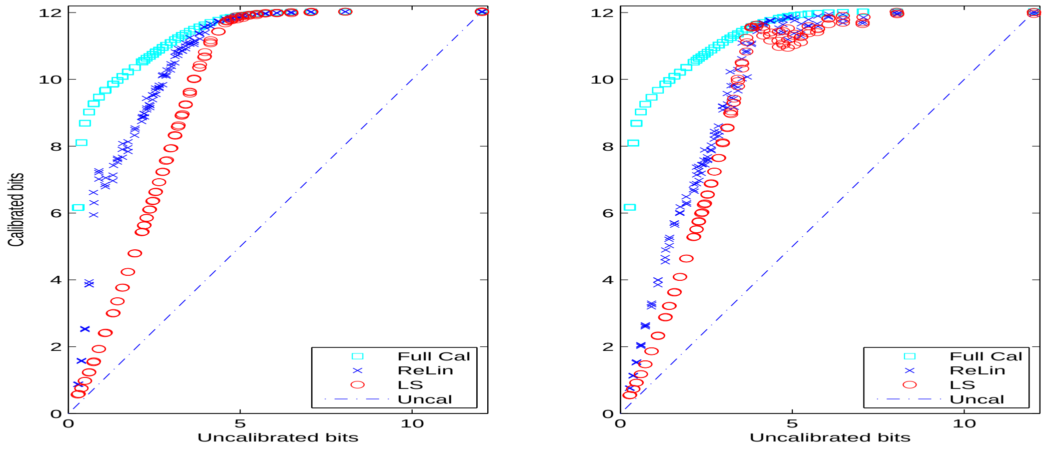

Figure 3-7: Calibration performance on systems with varying lowpass filter cutoffs, (10% overestimated filter cutoff on left, 20% on right). Plots show effective number of calibrated bits vs. uncalibrated bits for a 12-bit 2-ADC system with unknown timing skews.

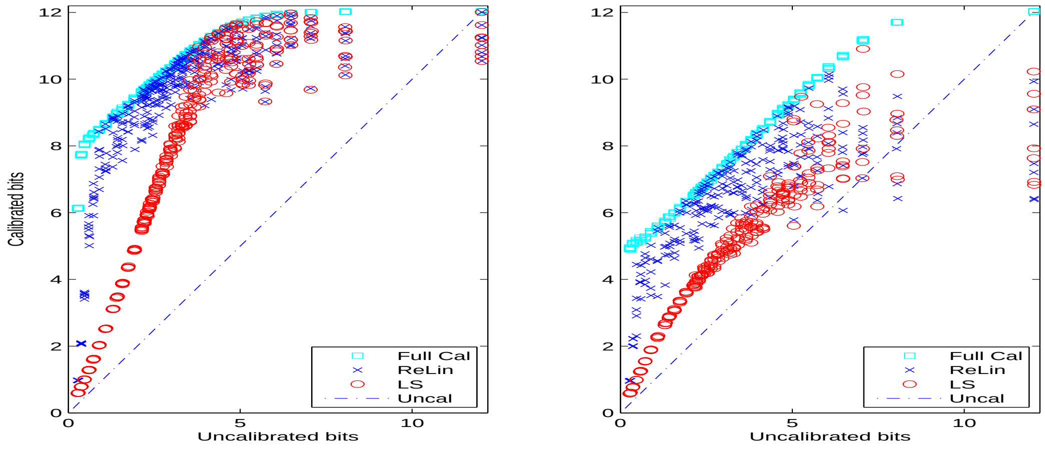

Figure 3-8: Calibration performance on systems with varying levels of out-of-band signal content, (−40dB on left, −20dB on right). Plots show effective number of calibrated bits vs. uncalibrated bits for a 12-bit 2-ADC system with unknown timing skews.

3.3.3 Block Size, Oversampling, and Quantization Level

Tradeoffs in performance were also measured for varying input block sizes and amounts of oversampling. After a baseline amount of oversampling (15%) and block size (2^{13} samples/ADC), varying these parameters had marginal effects on the performance. However, the convergence speed of the algorithm was highly dependent on the oversampling factor; with more oversampling, fewer relinearization iterations were necessary to achieve the same performance. To model lower resolution ADCs, the input was also quantized to 4-bit and 8-bit levels. Performance curves were similar to the 12-bit curves with upper bound limits at 4 bits and 8 bits respectively.

For a fixed oversampling rate, tests with smaller numbers of converters showed better performance. With more converters, the number of timing skews increased causing additional sources of estimation error. In general, the decrease in performance for larger numbers of ADCs is more apparent at lower levels of uncalibrated bits.

3.3.4 Practical Filter Lengths

To test the tradeoff between complexity and performance, simulations were also conducted with truncated filters, Figs. 3-9 and 3-10 . Using the recursive least-squares (RLS) implementation, the filtering processes of $p[n]$ and $h[n]$ were limited to $(175, 75, 30)$ and $(30, 10, 5)$ taps respectively. Shortening the high-pass filter p did not significantly change the performance of the LS calibration; however, the ReLin performance decreased with shorter filters. For three uncalibrated bits, the 175 tap ReLin increased LS performance from 8 to 11 bits; but the 30 tap ReLin remained at 8 bits. The $h[n]$ derivative filter had the opposite effect: LS performance decreased with fewer taps but ReLin performance remained the same. For five uncalibrated bits, the five tap LS provided six calibrated bits whereas the 30 tap filter yielded 11 calibrated bits. Thus, different system implementations will require a different set of filter lengths based upon the computational power and performance requirements.

Aside from low parameter estimation complexity, real-time systems will also require low input reconstruction complexity. In Chapter 2, we compared multiple reconstruction methods of varying complexity: the noise-free ideal method, the approximation method, and the high-noise method. As discussed previously, the high-noise (Constrained Least Squares) method is computationally inefficient due to the matrix inversions involved. Thus,

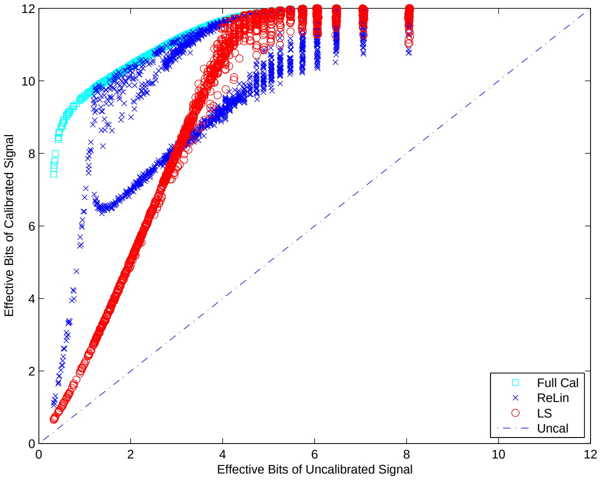

Figure 3-9: LS and ReLin calibration performance for varying number taps in the low-pass filter. Three distinct 'LS' curves from top to bottom representing filters of lengths $(175, 75, 30)$. Plots show effective number of calibrated bits vs. uncalibrated bits for a 12-bit 2-ADC system with unknown timing skews.

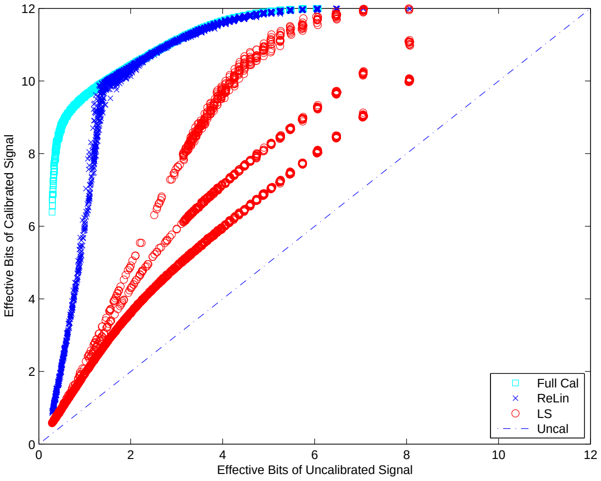

Figure 3-10: LS and ReLin calibration performance for varying number taps in the derivative h filter. Three distinct 'ReLin' curves from top to bottom representing filters of lengths $(30, 10, 5)$. Plots show effective number of calibrated bits vs. uncalibrated bits for a 12-bit 2-ADC system with unknown timing skews.

we are left to analyze the performance-complexity tradeoff of the first two methods.

The noise-free full reconstruction (2.10) uses a different filter for each of the M sampling converters. The filter is non-linearly dependent upon the skew estimates and requires a significant number of multiplies to compute. On the other hand, the reconstruction approximation only has one filter, which is fixed. This filter is used in skew estimation so no extra filtering is necessary. Using the approximation in reconstruction as well as parameter estimation saves computation, takes less memory, and does not require extra computation with changing timing skews. However, the system is limited by the approximation used in skew estimation, thus for least-squares estimates, it is sufficient to use the approximation in reconstruction. For relinearization and more complicated skew estimate procedures, the full reconstruction may provide a necessary improvement.

3.3.5 Relinearization Iterations

As stated before, the tests were performed with $\sim 100\tau/T_s$ iterations. This number was selected manually as iterations past this point generally made little improvement. In practice, the system does not have knowledge of τ and the number of iterations cannot be set in this manner. Instead, the system can implement a cutoff τ_s such that when the skew estimates between iterations do not change by more than τ_s, the iterations stop.

Other methods include amplifying or attenuating the skew change between iterations through a step-size. The use of step-size is already present in sequential method in Section 3.2.4 through the μ parameter. This step-size can also be variable and depend on the dynamics of the skew changes.

3.3.6 Sequential Implementation

In Sec. 3.2.4, we presented the RLS and LMS methods as two possible sequential implementations of the least-squares calibration algorithm. In Fig. 3-11, we show the convergence speeds of these algorithms by plotting the calibrated ENOB over time for a fixed set of skews. At each point in time, skew estimates are calculated. Theses estimates are then used over the entire signal in order to determine the ENOB. We used system parameters of $\delta = 0.001$, $\lambda = 0.95$ for the RLS algorithm and $\mu = 0.05$ for the LMS algorithm. Performance is compared against the block calibration performance with block size $N = 2^{15} = 32768$.

As we can see, the block calibration has a performance of ~ 7.5 effective bits. The

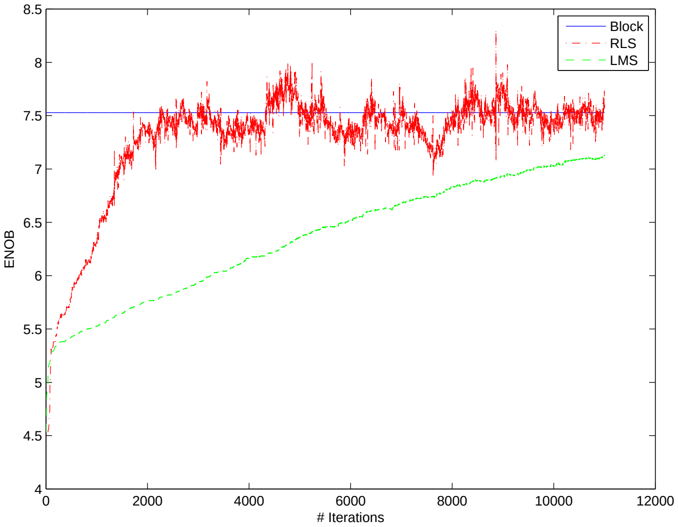

Figure 3-11: Speed of convergence for RLS and LMS shown through calibrated ENOB performance and compared to the block calibration for an $M = 4$ component system

test shows convergence of RLS after 2000 samples and LMS after 15000 samples. Thus, convergence rate of RLS is faster than the LMS; however the complexity of computation is lower for LMS. Within this simulation, the RLS fluctuates after it reaches the block performance. These fluctuations can be minimized by changing the forgetting factor λ over time.

By dynamically changing the (λ, μ) parameters over time, we can increase the convergerence speed by first using a large stepsize and then using a smaller stepsize. However, if a sudden change in the skew parameters occurs, it can result in a longer time for convergence.

3.3.7 TIADC Gains

Additional tests were performed for signal calibration in time-interleaved systems that contain both unknown timing skews and unknown gains. The $M-1$ gains were chosen independently using a uniform distribution. Although the initial least-squares estimate was often of poor quality, the relinearization technique achieved ~ 10 bit performance for systems with $M \leq 8$ converters and with greater than five uncalibrated bits. This performance is similar to the tests when only timing skews were unknown. For systems with $M = 16$ converters, the gain and skew parameter estimation converged to a local minimum, often not close to the true parameters.

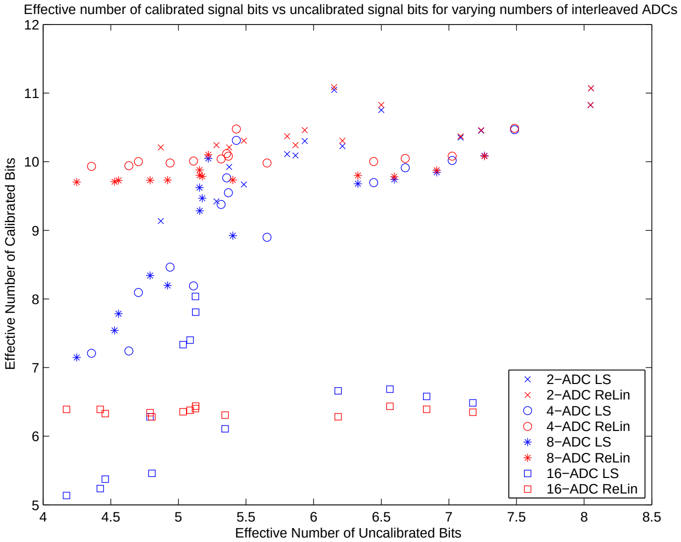

Figure 3-12: Effective number of calibrated bits vs. uncalibrated bits for varying numbers of ADCs with unknown gains and timing skews.

When gain mismatches are present, the number of uncalibrated bits can be both a function of the range of the timing skew and the range of the gains, i.e. a 4-bit signal can be produced by high timing skews or high gains. Signal recovery performance is dependent on whether the gain range or timing skew range is higher. When a larger gain range causes the decrease in uncalibrated bits, the calibration methods are slightly more effective than when a larger skew range causes the decrease in bits. In the gain case, convergence speed of the algorithm proved to be a bigger issue. The convergence was highly dependent on the oversampling factor; with more oversampling, the methods required fewer relinearizations. In the gain case, convergence speed of the algorithm proved to be a bigger issue. The convergence was highly dependent on the oversampling factor; with more oversampling, the methods required fewer relinearizations.

3.4 Hardware Feasibility

In this section, we explicitly state the signal processing operations necessary for calibration and analyze the feasibility of a hardware implementation. As noted previously, the RLS and LMS algorithms of Sec. 3.2.4 are more suited for hardware than the block approach due to their lower memory requirements and distribution of computation. The decreased complexity of the LMS algorithm makes it the most attractive candidate for implementation. We now analyze an LMS hardware implementation.

LMS Implementation

$$
\begin{aligned}
\boldsymbol{\gamma}[n] &= \sum_{m=n-k}^{n+k} p[m]y[n-m] && (3.37)\\
s[n] &= \sum_{m=n-k}^{n+k} h[m]y[n-m] && (3.38)\\
u_i[n] &= \sum_{\substack{m (\mathrm{mod}\ M)=i \\ |m-n|<k}} p[n-m]s[m] \quad i = n \ (\mathrm{mod}\ M) && (3.39)\\
\xi[n] &= \boldsymbol{\gamma}[n] - \hat{\boldsymbol{\tau}}^T[n-1]\mathbf{u}[n] && (3.40)\\
\hat{\boldsymbol{\tau}}[n] &= \hat{\boldsymbol{\tau}}[n-1] + \mu\mathbf{u}[n]\xi[n] && (3.41)\\
\hat{x}[n] &= y[n] - \hat{\tau}_i[n-1]s[n] \quad i = n \ (\mathrm{mod}\ M) && (3.42)
\end{aligned}
$$

In Sec. 3.3.6, we showed that convergence occurs within 15000 samples. For a 500 Mhz TIADC, this implies convergence in 0.03ms, a speed which is acceptable in most systems. This value can be reduced further if necessary. Because convergence time is not a problem, we now examine the calibration complexity.

For LMS implementation, filtering operations are necessary to create signals $\mathbf{u}[n]$ and $[n]$. In Sec. 3.3.4, we show that ~ 30 taps ($k = 15$) are necessary in the $h[n]$ and $p[n]$ filters in order to achieve sufficiently-high calibration performance (without using relinearization).

From (3.37) above, we see that computing $\boldsymbol{\gamma}[n]$ only requires a total of 30 multiplies per time step. Likewise, creating all the elements in vector $\mathbf{u}[n]$ requires $30+30 = 60$ multiplies per time step in (3.38) and (3.39). Multiplying the skew estimates $\hat{\boldsymbol{\tau}}$ and $\mathbf{u}$ in (3.40) takes M multiplies, and finally multiplying $\mathbf{u}$ and ξ in (3.41) is another M multiplies. Finally, (3.42) requires a single additional multiply. This leads to a total of $\sim 90+2M$ multiplies/samples. Thus, calibration can be performed using any general purpose DSP.

In order to compute $\boldsymbol{\gamma}[n]$, a total of $2k$ values of $y[n]$ need to be stored. Note the computation of $u_i[n]$ requires the computation of $s[n]$, thus we need to increase the buffer size by $2k$ samples in order to ensure $s[n]$ has been calculated.

The resulting performance of this LMS implementation is given under the 'LS' curve in Fig. 3-5 for $M = 2$ and Fig. 3-6 for $M = 16$. A further reduction in complexity is possible through better filter design. Increased performance can also be obtained via relinearization, however more steps are required to recompute the M filters and longer filter lengths are necessary.

Until now, we have focused on the case of small timing mismatch. We developed a least-squares calibration method based on the approximation of the reconstruction equation. Generalizations to increase performance through relinearization and to handle gains were also presented. To reduce complexity, we developed sequential algorithms using adaptive filters. Performance plots were also presented for a variety of cases to understand the practicality of the calibration scheme.

In the next chapter, we explore the bounds on calibration performance when computational complexity is not a constraint. These bounds provide an understanding of how well the calibrations methods compare against the optimal performance.

Chapter 4

Resampling Performance Bounds

A variety of calibration algorithms exist for the mitigating of mismatch errors; however, it is unclear what the bounds of calibration performance are for reconstructing the signal. In this chapter, we compute bounds on the expected performance when the timing skews are distributed uniformly.

Thus, we would like to address the question, what is the best we can do in estimating the input from the output samples? Also, how does the performance change if we change the oversampling factor? By answering these questions, we setup a framework on which to compare the effectiveness of all calibration algorithms in such systems.

When performing calibration, we often decouple the problems of skew estimation and signal reconstruction. For our performance analysis, we remove the problem of skew estimation. There are multiple reasons for doing so. With low levels of sampling noise, the skews can be uniquely determined given sufficient computational power. To do so, one can perform a brute force search through the entire parameter space for the skews that produce an accurate reconstruction. This search is possible due to the local continuity of out-of-band energy with respect to the skews. Also, with sufficiently long block lengths, the effects of noise on the estimates can be averaged out.

Accordingly, we focus on input signal reconstruction when the timing skews are known, as in Section 2.4. The goal is to minimize the effects of the sampling noise on the input estimate. Unlike the skew estimation, these effects are not averaged out over time since each sample sees a different noise realization. To measure the input estimation error, we calculate the distribution of eigenvalues for the sampling matrices in (2.16) and how it varies with (M, M_0).

4.1 Sampling Matrices

Within our setup, we assume that the sampling period of the individual components is fixed at M_0T_s; where at least M_0 components are necessary to achieve an overall sampling period of T_s. The number of sampling components is $M \geq M_0$, and we label the factor of oversampling as $r = M/M_0$.

We use the matrix formulation from (2.16) where $N \times 1$ vector $\boldsymbol{x}$ represents the deterministic input sampled at rate T_s. The $N \times 1$ vector $\mathbf{y}$ vector the nonuniform recurrent samples with the same average period as $\boldsymbol{x}$,

$$\boldsymbol{y} \quad = \quad \mathbf{F}\boldsymbol{x} + \boldsymbol{w}. \tag{4.1}$$

This expansion of this equation is equal to

$$\begin{bmatrix} y_0[0] \\ \vdots \\ y_{M-1}[0] \\ \vdots \\ y_0[N_0-1] \\ \vdots \\ y_{M-1}[N_0-1] \end{bmatrix} = \begin{bmatrix} - & \mathbf{f_0^0} & - \\ & \vdots & \\ - & \mathbf{f_{M-1}^0} & - \\ & \vdots & \\ - & \mathbf{f_0^{N_0-1}} & - \\ & \vdots & \\ - & \mathbf{f_{M-1}^{N_0-1}} & - \end{bmatrix} \begin{bmatrix} x[0] \\ \\ \\ \vdots \\ \\ \\ x[N-1] \end{bmatrix} + \begin{bmatrix} w_0[0] \\ \vdots \\ w_{M-1}[0] \\ \vdots \\ w_0[N-1] \\ \vdots \\ w_{M-1}[N-1] \end{bmatrix} \tag{4.2}$$

where the $\mathbf{f_i^j}$ represent rows of $\mathbf{F}$, which are functions of the skews, $f_i^j[n] = \text{sinc}(jM + i - n)$ for $0 \leq n \leq N-1$. The individual components sample with period M_0T_s implying $N_0 = N/M_0$

We group together the rows with common skews in order to reorganize our matrices in terms of sampling components rather than sample times

$$\begin{bmatrix} \boldsymbol{y}_0 \\ \vdots \\ \boldsymbol{y}_{M-1} \end{bmatrix} = \begin{bmatrix} \mathbf{F_0} \\ \vdots \\ \mathbf{F_{M-1}} \end{bmatrix} \boldsymbol{x} + \boldsymbol{w} \tag{4.3}$$

where each $N_0 \times N$ sampling matrix $\mathbf{F_i}$ represents rows of sinc functions corresponding to a single component i and $\mathbf{y}_i$ is the corresponding received vector. Therefore, the overall sampling matrix $\mathbf{F}$ is the concatenation of the matrix from each component. We now examine these sampling matrices in detail.

4.1.1 Single Component Sampling Matrix

Let us start with a simple sampling matrix $\mathbf{F}$ corresponding to a single component. The vector $\boldsymbol{x}$ represents the input sampled at times $t = nT_s$. Suppose a single component also samples with period T_s, starting at time $t = 0$. It captures samples at the exact time and rate as the desired signal $\boldsymbol{x}$. The sampling matrix in this case is the identity matrix $\mathbf{F} = \mathbf{I}$, i.e., $y[n] = x[n] + w[n]$.

Now, suppose the component samples at half this rate (period $2T_s$) and starts at time $t = 0$, the corresponding $N/2 \times N$ sampling matrix is every other row of the identity matrix

$$\mathbf{F} \quad = \quad \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{4.4}$$

In the signal domain, this matrix is equivalent to $y[n] = x[2n]+w[n]$. Both of these sampling matrices have the property that their rows are orthonormal i.e.,

$$\mathbf{F}\mathbf{F}^T = \mathbf{I}. \tag{4.5}$$

It is also possible for a component to have a non-zero starting time, e.g. $t_0 = T_s/2$. For a sampling period of T_s, the matrix can be approximated through the matrix $[\mathbf{F}]_{i,j} = \text{sinc}(i-j+1/2)$, for $0 \le i, j \le N-1$. This sampling is equivalent to $y[n] = x[n+1/2]+w[n]$, where $x[n + 1/2]$ represents a 1/2 sample fractional delay on $x[n]$. The effects of filter and signal truncation are discussed later. For a component with the same start time $t_0 = T_s/2$ and double the sampling period $2T_s$, the corresponding $N/2 \times N$ sampling matrix can be formed by selecting every other row.

The sampling matrices can be viewed as orthogonal projection directions. The period T_s sampling matrices have size $N \times N$ and have orthonormal rows which span the $\mathcal{R}^N$ space.

Thus, these matrices sample at the full dimensionality of the space. By selecting certain rows like the period $2T_s$ matrices, we attain a subset of orthonormal vectors for the space. We define these matrices as 1/2-rate; they use sinc functions to sample half the dimensions of the space.

4.1.2 Multiple Sampling Matrices

In the distributed sampling setup, we use multiple components with different starting times to ensure that the signal is oversampled. Thus, we analyze sampling matrcies $\mathbf{F}$ that are a combination of more than one component. By using two sampling components, i.e. 1/2-rate matrices, the overall sampling matrix is

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_0 \\ \mathbf{F}_1 \end{bmatrix} \tag{4.6}$$

where $\mathbf{F}_0$ corresponds to a 1/2 rate component starting at τ_0, i.e., $[\mathbf{F}_0]_{k,l} = \text{sinc}(2k - l - \tau_0)$. Similarly, $\mathbf{F}_1$ corresponds to a 1/2 rate component starting at τ_1. The matrix $\mathbf{F}$ has full row rank with probability one.

In the absence of noise, the input can be reconstructed by applying the inverse of the sampling matrix to the component outputs $\boldsymbol{x} = \mathbf{F}^{-1}\boldsymbol{y}$. However, our goal is to understand the effects of noise on the input estimate. For now, we ignore the fact that $\boldsymbol{x}$ lies in a smaller dimensional subspace of $\mathcal{R}^{\mathcal{N}}$, i.e., the bandlimited property, and address this later.

Applying the inverse matrix to noisy components yields the signal estimate

$$\hat{\boldsymbol{x}} \quad = \quad \boldsymbol{x} + \mathbf{F}^{-1}\boldsymbol{w}. \tag{4.7}$$

It is important to understand how this inverse $\mathbf{F}^{-1}$ changes with the sample timings of the components. Although $\mathbf{F}$ has full rank, the matrix may be ill-conditioned if the sampling components are spaced very close to each other in time. The ill-conditioning is caused by having a singular value close to zero. In this case, the projection of input $\boldsymbol{x}$ along some basis directions becomes dominated by the noise in that direction, and the estimate will have a large amplification of noise along the singular direction. The resulting estimate error

$$\sigma_{\hat{x}}^2 \quad = \quad \frac{1}{N}\,\text{tr}((\mathbf{F}^{-1})(\mathbf{F}^{-1})^T)\sigma_w^2 \tag{4.8}$$

increases in this case.

Thus, different sets of starting times (τ_0,τ_1) for the sampling components result in widely varying estimate errors (4.8). For components which sample at identical times (which has probability 0), the matrix $\mathbf{F}$ becomes singular. On the other hand, with components evenly spaced apart $\tau_1 - \tau_0 = T_s$, the matrix $\mathbf{F}$ becomes orthonormal again and all singular values are equal to 1. It is easy to show that uniform spacing will minimize the overall noise in the estimate, $\sigma^2_{\hat{x}} = \sigma^2_w$.

In the case of $M = 3$ components with period $2T_s$, the $3N/2 \times N$ sampling matrix $\mathbf{F}$ is now the concatenation of three 1/2-rate matrices. The optimal signal estimate is now computed by applying the pseudoinverse of the sampling matrix, yielding error

$$\begin{aligned}
\sigma^2_{\hat{x}} &= \frac{1}{N}\operatorname{tr}((\mathbf{F}^\dagger)(\mathbf{F}^\dagger)^T)\sigma^2_w && (4.9)\\
&= \frac{1}{N}\operatorname{tr}((\mathbf{F}^T\mathbf{F})^\dagger)\sigma^2_w && (4.10)\\
&= \frac{1}{N}\left(\sum_{i=0}^{N-1}\frac{1}{\sigma_i^2}\right)\sigma^2_w && (4.11)
\end{aligned}$$

where the σ_i are the singular values of $\mathbf{F}$.

Similar to the two component case, the matrix becomes ill-conditioned if the components are close in timing; and evenly spaced components yield all singular values equal to 1.5, resulting in the lowest noise variance. What benefit is there to adding the third component? First, the noise variance drops to $\sigma^2_{\hat{x}} = 2\sigma^2_w/3$ for evenly spaced components. Also, if the component timings have a uniform distribution, the third component reduces the probability that $\mathbf{F}$ is ill-conditioned.

Now, we can finally analyze the general (M, M_0) setup, where M components are used to sample when only M_0 are required, as described in Section 5.1. In this case, the sampling matrix $\mathbf{F}$ is the concatenation of M component $1/M_0$-rate matrices. The individual $N/M_0 \times N$ matrices are

$$[\mathbf{F}_i]_{k,l} = \operatorname{sinc}(kM_0 - l - \tau_i), \tag{4.12}$$

where τ_i is the sampling shift (start time) of component i, where $0 \leq \boldsymbol{\tau} < M_0T_s$. The total size of $\mathbf{F}$ is $rN \times N$, where again r is the oversampling factor M/M_0. Because the rows have unit norm, the sum of the singular values for any set of sampling skews $\boldsymbol{\tau}$ is equal to

$$\begin{aligned} \sum_{i=0}^{N-1} \sigma_i^2 &= \operatorname{tr}(\mathbf{F}\mathbf{F}^T) && (4.13) \\ &= \|\mathbf{F}\|_F^2 && (4.14) \\ &= rN && (4.15) \end{aligned}$$

where $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm.

For this fixed singular value sum, the set of component timings $\boldsymbol{\tau}$ that minimize the estimate error (4.11) have a sampling matrix whose singular values are equal,

$$\begin{aligned} \boldsymbol{\sigma}_{opt}^2 &= \underset{\sum \sigma_i^2 = rN}{\arg\min} \frac{1}{N} \operatorname{tr}((\mathbf{F}^T\mathbf{F})^\dagger)\sigma_w^2 && (4.16) \\ &= \underset{\sum \sigma_i^2 = rN}{\arg\min} \sum \frac{1}{\sigma_i^2} && (4.17) \\ &= r && (4.18) \end{aligned}$$

where $\boldsymbol{\sigma}_{opt}^2$ is the optimizing set of singular values (also shown in [21]). Thus, the minimum possible error is

$$\sigma_{\hat{x}}^2 = \frac{1}{r}\sigma_w^2 \qquad (4.19)$$

and there are multiple sets of timing skews which can achieve this value.

In particular, this minimum error is achieved when the M sampling matrices can be combined to form r unitary matrices. This situation is present in multiple timing setups, including the case of M uniformly spaced components and the case of M_0 uniformly spaced times with a redundancy of r components at each time (assuming an integer r). In each of these cases, the $\mathbf{F}$ matrix can be divided into r unitary matrices.

Note, our setup is fundamentally different than a system which observes noisy signal through noisy sampling, $y[n] = x(t_n) + w(t_n)$. In this second situation, the noise realization is fixed, i.e. all components sampling at time $t = 0$ will observe the same value $x(0) + w(0)$. Unlike the previous setup, which benefited equally from having r sets of M_0 evenly spaced components sampling at the same time as from having the r sets sampling uniformly, in the fixed noise setup the identically timed sets have lower performance. For fixed AWGN noise on the continuous signal $x(t)$, optimal performance is achieved with non-overlapping sets of components.

This can be viewed geometrically as wanting the maximum amount of spacing between the orthonormal vectors of the r unitary bases. Returning to our original setup, we ignore any fixed noise on the input and consider it to be random between each sample.

Throughout the previous discussion, we implied that an integer number of components is necessary to sample the input at Nyquist rate, i.e., $M_0 \in \mathbb{Z}$. It also possible for M_0 to be a fraction, e.g., $M_0 = 1.5$ components samples with period $3T_s/2$ to form 2/3-rate sampling matrices. The same results hold; however we continue to imply integer M_0 due to its direct physical meaning.

Within our representation of the distributed sampling problem, we use matrices to represent filtering operations and vectors to represent signals. Although truncation effects occur due to the finite block length, we consider the effects negligible for sufficiently large N. This assertion is verified in simulations where we use a $N + 2 \times N$ sized matrix to provide extra resolution to the first and last sample of $\boldsymbol{x}$. For large N, the decrease in error is marginally small, allowing for the truncations to be made without significant changes in performance.

Although uniformly spaced components achieve minimum error, the M components in setups like extreme oversampling have a uniform distribution and no guarantee can be made on their spacing. In order to quantify the performance bound, it is necessary to analyze the distribution and statistics of the reconstruction noise in the extreme setup. To do so, we will first provide a lower bound on the expected performance of the sinc sampling matrix. We then compare this result to the expected error for sampling projections using arbitrary random matrices.

4.2 Random Sampling Skews

For systems with a large number of components, we model their skews τ_i as being uniformly distributed in the $[0, M_0 T_s]$ interval. The randomness of these skews leads to a random charateristic of the sampling matrices, i.e., the projection directions for each sampling component are not deterministic. The noise power depends on these skew values and how they are spaced.

For example, suppose that the set of skews are all located close to each other, the resulting projection directions of the matrices will also be close together. In this case, the input in these directions will have high-SNR, but the overall SNR will be dominated by the lack of components sampling in other directions. As we increase the oversampling factor, the probability of all the skews being close together decreases and the expected noise on the input estimate also decreases. In this section, we compute statistics of the noise power for randomly distributed timing skews and we measure how they change with respect to system parameters such as r or M_0. For an ease of notation, we assume that the sampling period is unity, $T_s = 1$.

Let us briefly restate the problem. We model the sampling matrix $\mathbf{F}$ as the concatenation of M independent $1/M_0$-rate matrices. The reconstruction error for such a system is directly proportional to $\mathrm{tr}((\mathbf{F}^T\mathbf{F})^\dagger)$. Thus, we aim to calculate the distribution of this error and how it varies with (M, M_0).

As seen before, the tr is related to the singular values of $\mathbf{F}$. For $M \geq M_0$, with probability one, all singular values are greater than zero. It is clear to see that

$$\boldsymbol{\sigma}_{\mathbf{F}^T\mathbf{F}} = \boldsymbol{\sigma}_{\mathbf{F}}^2 = \boldsymbol{\lambda}_{\mathbf{F}^T\mathbf{F}} = \boldsymbol{\sigma}_{(\mathbf{F}^T\mathbf{F})^\dagger}^{-1} = \boldsymbol{\sigma}_{\mathbf{F}^\dagger}^{-2} = \boldsymbol{\lambda}_{(\mathbf{F}^T\mathbf{F})^\dagger}^{-1}. \tag{4.20}$$

By calculating the moments of any term above, we can calculate the moments of any other term.

We define new $N \times N$ matrices

$$\mathbf{G}_i \quad = \quad \mathbf{F}_i^T\mathbf{F}_i \qquad \text{for } 0 \leq i \leq M-1 \tag{4.21}$$

and

$$\mathbf{G} \quad = \quad \mathbf{F}^T\mathbf{F} = \sum_i \mathbf{F}_i^T\mathbf{F}_i = \sum_i \mathbf{G}_i. \tag{4.22}$$

where $\mathbf{F}$ is the sampling matrix for M components, as seen in (4.3). The eigenvalues of $\mathbf{G}$ and $\mathbf{G}_i$ are related to the singular values of $\mathbf{F}$ and $\mathbf{F}_i$, respectively, by (4.20). The elements of $\mathbf{G}$ are given by

$$[\mathbf{G}]_{k,l} \quad = \quad \sum_{i=0}^{M-1} \sum_{n=0}^{N/M_0-1} \operatorname{sinc}(nM_0 - k - \tau_i) \operatorname{sinc}(nM_0 - l - \tau_i) \tag{4.23}$$

for $0 \leq k, l \leq N-1$.

Due to the complicated structure of matrix $\mathbf{G}$, it seems analytically intractable to compute the error, i.e., $\operatorname{tr}(\mathbf{G}^{-1})$, from (4.23). Instead, we look to lower bound the error. Because the error varies highly for different sets of timing skews, we bound its expecation and see how this changes with r and M_0.

4.2.1 Error Lower Bound

Because $\mathbf{G}$ is a positive semidefinite matrix, the input estimation error is bounded by

$$\frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{\lambda_k} \geq \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{G_{k,k}} \tag{4.24}$$

where $\boldsymbol{\lambda}$ is used to designate the eigenvalues of $\mathbf{G}$. Therefore, to lower bound the expected error, we compute the expectation

$$E_{\boldsymbol{\tau}}\left[\frac{1}{\mathbf{G}_{k,k}}\right] \quad = \quad E_{\boldsymbol{\tau}}\left[\left(\sum_{i=0}^{M-1} \sum_{n=0}^{N/M_0-1} \operatorname{sinc}(nM_0 - k - \tau_i)^2\right)^{-1}\right] \tag{4.25}$$

For finite block lengths, the expectations above are difficult to compute because they change depending on the value of k. We avoid this issue by examing the limiting value as we take the block length N to infinity. To do this, we redefine our indexing to $[-N/2+1, N/2]$ instead of $[0, N-1]$. In doing so, the $\boldsymbol{x}$ vector from (4.3) represents $x[-N/2+1]$ to $x[N/2]$ and the $\mathbf{F}$ and $\mathbf{G}$ matrices also change their indexing. These changes allow us to take the limit without any changes necessary in the development.

$$\lim_{N\to\infty} E_{\boldsymbol{\tau}}\left[\frac{1}{\mathbf{G}_{k,k}}\right] \quad = \quad \lim_{N\to\infty} E_{\boldsymbol{\tau}}\left[\left(\sum_{i=0}^{M-1} \sum_{n=\frac{1}{M_0}(-\frac{N}{2}+1)}^{\frac{N}{2M_0}} \operatorname{sinc}(nM_0 - \tau_i)^2\right)^{-1}\right] \tag{4.26}$$

where the dependence on k drops out because the $\boldsymbol{\tau}$ parameters are distributed uniformly between $[0, M_0]$. By bringing the limit inside the expectation, we see this reduces to summing samples of the sinc function. By Parseval's equation, we find

$$
\begin{aligned}
E_{\boldsymbol{\tau}}\left[\frac{1}{\mathbf{G}_{k,k}}\right] &= E_{\boldsymbol{\tau}}\left[\left(\sum_i \frac{1}{2\pi}\int_0^{2\pi}\frac{1}{M_0^2}\sum_{l=0}^{M_0-1}\sum_{m=0}^{M_0-1} e^{-j\left(\omega-\frac{2\pi}{M_0}l\right)\tau_i}e^{j\left(\omega-\frac{2\pi}{M_0}m\right)\tau_i}\right)^{-1}\right] \qquad (4.27)\\
&= E_{\boldsymbol{\tau}}\left[\left(\frac{M}{M_0}+\frac{1}{M_0^2}\sum_i\sum_{u=1}^{M_0-1}(M_0-u)\left(e^{j\frac{2\pi}{M_0}u\tau_i}+e^{-j\frac{2\pi}{M_0}u\tau_i}\right)\right)^{-1}\right] \qquad (4.28)\\
&= E_{\boldsymbol{\tau}}\left[\left(\frac{M}{M_0}+\frac{2}{M_0^2}\sum_i\sum_{u=1}^{M_0-1}(M_0-u)\cos\left(\frac{2\pi}{M_0}u\tau_i\right)\right)^{-1}\right] \qquad (4.29)
\end{aligned}
$$

For $M_0 = 2$, this becomes

$$
\begin{aligned}
E_{\boldsymbol{\tau}}\left[\frac{1}{\mathbf{G}_{k,k}}\right] &= E_{\boldsymbol{\tau}}\left[\left(\frac{M}{2}+\frac{1}{2}\sum_i\cos\left(\pi\tau_i\right)\right)^{-1}\right] \qquad (4.30)\\
&= \frac{2}{M}E_{\boldsymbol{\tau}}\left[\sum_{j=0}^{\infty}\left(\frac{-1}{M}\sum_i\cos\left(\pi\tau_i\right)\right)^{j}\right] \qquad (4.31)\\
&= \frac{2}{M}+\frac{2}{M}\sum_{j=1}^{\infty}\frac{1}{M^{2j}}E_{\boldsymbol{\tau}}\left[\left(\sum_i\cos\left(\pi\tau_i\right)\right)^{2j}\right] \qquad (4.32)
\end{aligned}
$$

The expectation for $j = 1$ becomes

$$
\begin{aligned}
E_{\boldsymbol{\tau}}\left[\left(\sum_i\cos\left(\pi\tau_i\right)\right)^{2}\right] &= E_{\boldsymbol{\tau}}\left[\sum_i\cos\left(\pi\tau_i\right)^2\right] \qquad (4.33)\\
&= E_{\boldsymbol{\tau}}\left[\sum_i\left(1+\cos\left(2\pi\tau_i\right)\right)/2\right] \qquad (4.34)\\
&= \frac{M}{2} \qquad (4.35)
\end{aligned}
$$

and using similar techniques, we can show that for $j = 2$,

$$
E_{\boldsymbol{\tau}}\left[\left(\sum_i\cos\left(\pi\tau_i\right)\right)^{4}\right] = \frac{M(2M+1)}{8}. \qquad (4.36)
$$

We can derive a lower bound of the $M_0 = 2$ case by ignoring any higher order terms in

(4.32),

$$E_{\boldsymbol{\tau}}\left[\frac{1}{\mathbf{G}_{k,k}}\right] \geq \frac{8M^3+4M^2+2M+1}{4M^4} \tag{4.37}$$

$$= \frac{64r^3+16r^2+4r+1}{64r^4}. \tag{4.38}$$

Therefore, the average error for uniformly distributed skews and $M_0 = 2$ is lower-bounded by (4.38). The tightness of the bound is dependent on the inverse diagonal inequality (4.24). For high values of r, this bound goes to $1/r$ as is the optimal value (4.19).

For $M_0 > 2$, the power series bounding technique is no longer valid and we resort to simulations of the lower bound. In Section 4.4, we provide a comparison between the expected estimate error of the sampling matrices and the lower bound developed here.

4.2.2 Adding the Subspace Constraint

The analysis has assumed that $\boldsymbol{x}$ is an arbitrary vector; in actuality, it is restricted to the bandlimited subspace, i.e. $(\mathbf{I}-\mathbf{L})\boldsymbol{x} = \mathbf{0}$. By incorporating this constraint, the optimal estimator is now $\hat{\boldsymbol{x}} = \mathbf{L}(\mathbf{F}\mathbf{L})^{\dagger}\mathbf{y}$, as given in (2.43). The lower bound can be rederived in a similar manner as above. The resulting bound replaces variables M_0 and τ_i with $M_0' = M_0\omega_c/\pi$ and $\tau_i' = \tau_i\omega_c/\pi$; and the eigenvalues are also scaled down by a factor of ω_c/π. Therefore, we find

$$E_{\boldsymbol{\tau}}\left[\frac{1}{\lambda}\right] = \omega_c' E_{\boldsymbol{\tau}}\left[\left(\frac{M}{M_0\omega_c'} + \frac{2}{(M_0\omega_c')^2}\sum_i \sum_{u=1}^{M_0\omega_c'-1}\left(M_0\omega_c' - u\right)\cos\left(\frac{2\pi u\tau_i}{M_0\omega_c'}\right)\right)^{-1}\right] \tag{4.39}$$

where $\omega_c' = \omega_c/\pi$ represents the fraction of eigenvalues of $\mathbf{L}$ that are equal to one. The summation above assumes $M_0\omega_c'$ is an integer, which only adds a small restriction to the applicable cases.

4.2.3 Graphical Meaning

There is a nice graphical interpretation of the bound generated in (4.26). In general, the estimation noise is dominated by the smallest eigenvalues. These eigenvalues correspond to the directions in which the sinc sampling matrices are deficient, e.g., if the $\mathbf{F}$ matrix has very low values for every element in column j, then we expect smaller eigenvalues for the eigenvectors in this direction, resulting in a higher error in the estimate of the jth sample. Likewise, if column j has large elements, we say that the sampling matrices are 'focusing' more power and we expect the estimate of this sample to be better. This intuition is slightly flawed in that a matrix such as

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{4.40}$$

appears to have sufficient power in each column however two eigenvalues are at zero and the estimator error blows up. One fix to this problem is to view the problem in a continous time domain.

We begin with M component skews on the interval $[0, M_0]$. Each component has a corresponding sinc function that is related to it, i.e., $\text{sinc}(t - \tau_i)$ for component i. By sampling with component i, the input signal is being projected onto samples of this sinc function. We can think about each component as having a finite amount of power it dedicates to time t', where $0 \leq t' < M_0$. Because the M_0-length intervals repeat, the total power to time t' by component i is

$$f_i(t') = \sum_k \text{sinc}(t' + kM_0 - \tau_i)^2. \tag{4.41}$$

Whereas in the previous example (4.40), we discussed the case of the sums over discrete columns; in this case, we are taking the sums over interpolated columns at time t' in the continous space. Therefore the power dedicated to time t by all components is

$$f(t') = \sum_i \sum_k \text{sinc}(t' + kM_0 - \tau_i)^2. \tag{4.42}$$

The correponding error at time t' is the inverse of this value. By taking the expectation of this inverse, the result is independent of t' because τ_i is distributed uniformly; therefore we set $t' = 0$. This results in the exact expectation in equation (4.26) of the bound that we calculated. Therefore, the bound is inverting the 'channel' for each of the locations $0 \leq t' < M_0$ and averaging these noise errors.

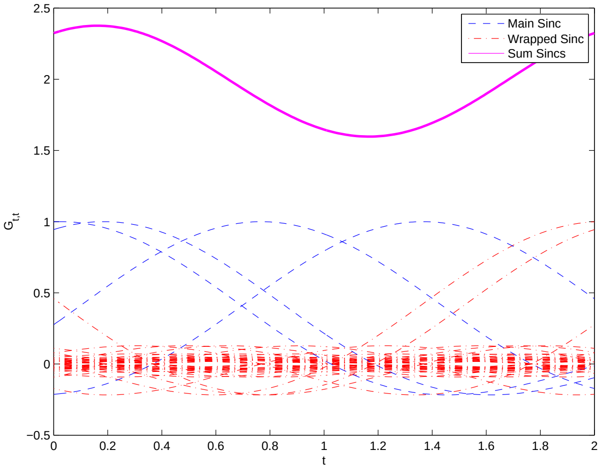

Figure 4-1: Sum of wrapped sinc function for $(M, M_0) = (4, 2)$. The dashed curves represent main sinc lobe in $[0, M_0]$, dashed-dot curves represent sinc wrapping from other intervals, the solid curve represent sum of sinc^2.

In Fig. 4-1, we plot an example of the sinc sum over the $[0, M_0]$ interval for a $(M, M_0) = (4, 2)$ system. Because there are fewer components sampling around time $t' = 1.15$, the sum (solid curve) is minimized at this time and the error in estimating this direction is higher.

4.3 Random Unitary Matrices

In this section, we compare the performance of the sinc sampling matrices to random matrices that are not restricted to the sinc structure. For an N dimensional space, we define the class of $1/M_0$-unitary matrices as matrices containing N/M_0 orthonormal rows. Although the $\mathbf{F}_i$ sampling matrices are $1/M_0$-unitary, their sampling is limited to projecting the input onto sinc functions, as shown in Fig. 4-2. By broadening the $1/M_0$-rate matrix class to arbitrary $1/M_0$-unitary matrices, we expect to achieve an lower-bound on the error due to the added degrees of freedom.

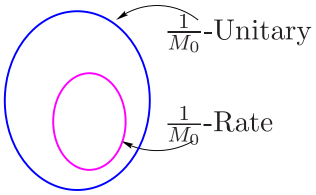

Figure 4-2: $1/M_0$-rate sinc sampling matrices are a subset of the $1/M_0$-unitary matrices

Each matrix $\mathbf{F}_i$ is now modeled N/M_0 rows of a random unitary matrix, where we use a uniform distribution over all possible unitary matrices. This distribution is also known as the Haar distribution (in contrast to the Wishart distribution for matrices with independent Gaussian entries). The $\mathbf{G}_i$ matrices can now be written as

$$\mathbf{G}_i = \mathbf{Q}_i \Sigma_i \mathbf{Q}_i^T \tag{4.43}$$

where the $\mathbf{Q}_i$ are random orthonormal matrices and Σ_i is a diagonal matrix of eigenvalues (N/M_0 at one and the rest at zero). In other words, the $\mathbf{G}_i$ are random matrices whose eigenvalues are distributed with CDF

$$F(\lambda) = \frac{1}{N}\#\{\lambda_m : \; \lambda_m \leq \lambda\} \tag{4.44}$$

$$= \begin{cases} 0 & \lambda < 0 \\ 1 - 1/M_0 & 0 \leq \lambda < 1 \cdot \\ 1 & 1 \leq \lambda \end{cases} \tag{4.45}$$

From this characterization, we can derive information about the overall sampling matrix. To do so, we would like to calculate the eigenvalue distribution of a sum of M random matrices, each with distribution (4.45). We use tools from random matrix theory and free probability to aid in our analysis [15].

It is useful to examine the geometrical pictures of these operations. In general, for two matrices $\mathbf{A}$ and $\mathbf{B}$, the eigenvalues of each $\boldsymbol{\lambda}_A$ and $\boldsymbol{\lambda}_B$ tell us little about the eigenvalues of the sum $\boldsymbol{\lambda}_{A+B}$ or product $\boldsymbol{\lambda}_{AB}$. For example, fixed 1/2-unitary matrices $(\mathbf{F}_1, \mathbf{F}_2)$ produce $(\mathbf{G}_1, \mathbf{G}_2)$ whose summation $\mathbf{G}_1 + \mathbf{G}_2$ has different possible eigenvalues.

The eigenvalues of the sum are all one if $\mathbf{G}_1^T \mathbf{G}_2 = \mathbf{0}$, i.e., all the projection vectors in $\mathbf{F}_1$ are orthogonal to vectors in $\mathbf{F}_2$. Half eigenvalues are two and half are zero if $\mathbf{G}_1 = \mathbf{G}_2$, i.e., all the projection vectors in $\mathbf{F}_1$ are aligned with the vectors in $\mathbf{F}_2$. And a range of other eigenvalues are also possible with other matrices. Thus, we think of each matrix $\mathbf{F}_i$ as a set of projection directions. The farther apart the directions are, the more well balanced the eigenvalues of $\mathbf{G}$ will be.

Figure 4-3 shows three different examples of $M - 2$ components in an $M_0 = 1.5$ system. The projection vectors of the first component (solid) and the second component (dashed) are plotted in (x, y, z) coordinates. In the first plot, three vectors are orthogonal, and the

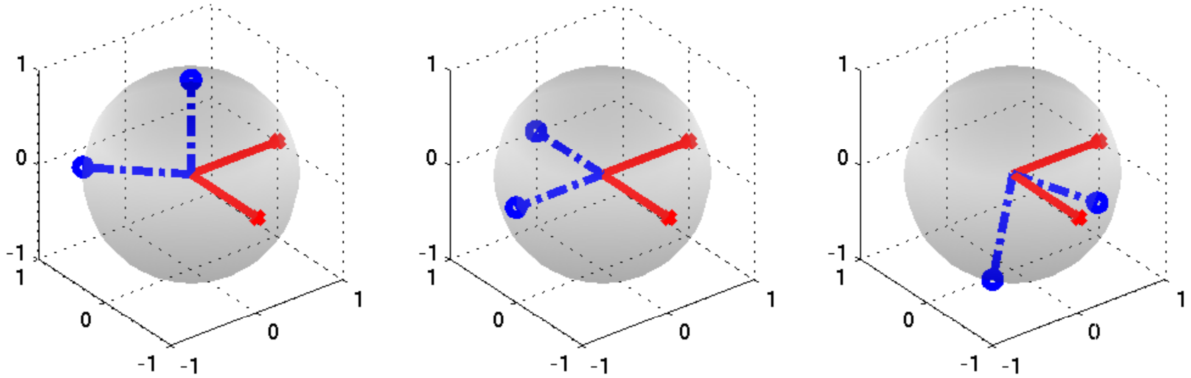

Figure 4-3: Projection directions of random matrices

fourth adds more information about the (x, y) plane. In the second plot, the first and second component are redundant and no information is gather about the z direction. And the third plot shows a random pair of components where the close spacing of projection vectors may decrease the eigenvalues in one direction.

When dealing with random matrices, the matrices are defined in terms of their eigenvalues. The eigenvectors, on the other hand, are chosen using the Haar distribution. Because of this unitary invariance of the projection directions, it is possible to make more precise claims on the eigenvalues of a summation of random matrices and on the eigenvalues of a product of random matrices. In the next section, we present different representations for these matrices, which allow us to calculate the eigenvalues of certain operations, such as addition and multiplication.

4.3.1 Random Matrix Representations

There are multiple ways to define and classify subclasses of random matrices. In this discussion, we limit ourselves to matrices that have a limiting eigenvalue distribution, i.e., as we increase the size N of the matrix, the eigenvalue distribution converges to a fixed distribution $F(\lambda)$. It is clear that our $\mathbf{G}_i$ matrices fall into this category.

There are multiple transforms of the limiting distribution that yield other useful representations. To start, we describe the basic Stieljes transform, which is calculated by

$$\begin{aligned} m_{\mathbf{G}}(z) &= \int \frac{1}{\lambda - z} dF(\lambda) \qquad (4.46) \\ &= E_\lambda \left[\frac{1}{\lambda - z} \right]. \qquad (4.47) \end{aligned}$$

This transform can also be rewritten in terms of the eigenvalue moments

$$m_{\mathbf{G}}(z) \quad = \quad -\sum_{i=0}^{\infty} E[\lambda^i] \frac{1}{z^{i+1}}. \tag{4.48}$$

Therefore, if we can compute $m_{\mathbf{G}^{-1}}(z)$, we can make statements about the moments of the estimation error.

To do so, we will follow a simple series of steps

$$m_{\mathbf{G}_i}(z) \rightarrow m_{\mathbf{G}}(z) \rightarrow m_{\mathbf{G}^{-1}}(z) \tag{4.49}$$

where the first arrow involves the summation of random matrices and second involves the inversion.

The Stieljes transform of the eigenvalue distribution of $\mathbf{G}_i$ is given by

$$m_{\mathbf{G}_i}(z) \quad = \quad -\frac{1}{z}\left(1 + \frac{1}{M_0}\frac{1}{z-1}\right). \tag{4.50}$$

We can rewrite this transform as a bivariate polynomial

$$L_{m(z),z} \quad = \quad \sum_i \sum_j a_{i,j} m^i z^j \tag{4.51}$$

where the solution to $L_{m_(z),z} = 0$ is given by Stieljes polynomial $m(z)$. In our case

$$L_{m_{\mathbf{G}_i}(z),z} \quad = \quad M_0 m z^2 - M_0 m z + M_0 z - M_0 + 1, \tag{4.52}$$

and the solution is given by (4.50). These transforms are computed using the *rmtool* toolbox developed in [42].

Through other matrix operations described in [42], we can compute the eigenvalues for an addition of random matrices by the free probability addition of their bivariate polynomials. By doing so, we find that the summation of M matrices $\mathbf{G}_i$ in (4.22) has a bivariate polynomial

$$L_{m_{\mathbf{G}}(z),z} = m^2(M_0 z^2 - M_0 M z) - m(M_0 M - 2M_0 z - M^2 + M_0 M z) + M_0 - M_0 M. \tag{4.53}$$

From this equation, we can compute the distribution of the eigenvalues of $\mathbf{G}$.

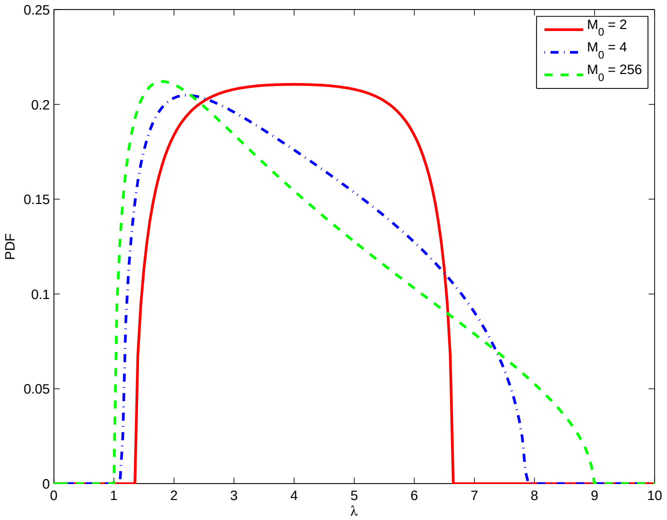

Figure 4-4: PDF of eigenvalues of $\mathbf{G}$ for $r = 4$ factor oversampling and varying numbers of base components M_0

The distribution of the eigenvalues for $r = 4$ is plotted in Fig. 4-4 for varying numbers of base components M_0. The plot represents the marginal distribution but can be viewed as a histogram of eigenvalues for large N. As M_0 increases, the PDF of the eigenvalues shifts its density to left while broadening its support. This result is expected; by fixing the oversampling ratio and increasing M_0, the rM_0 matrices can align to produce larger eigenvalues in some directions and smaller eigenvalues in others.

In Fig. 4-5, the eigenvalue distribution is shown for $M_0 = 4$ components and varying oversampling ratios. By increasing the redundancy r, the PDF tends towards a uniform distribution with a higher mean. These distribution plots allow us to understand how the eigenvalues grow with oversampling ratio, which directly translates to how the input estimation error shrinks with oversampling ratio.

Because analytically calculating the distribution function is difficult in closed form, we instead solve for the moments of the eigenvalues. Using (4.11), we find

$$E[\lambda] = r \tag{4.54}$$

$$E\left[\lambda^2\right] = r^2 + r - r/M_0. \tag{4.55}$$

and the expectations of the negative moments can be computed from

$$L_{m_{\mathbf{G}^{-1}}(z),z} = (M_0 z - M_0 M z^2)m^2 + (M_0 M - M_0 M z - z M^2)m - M^2. \tag{4.56}$$

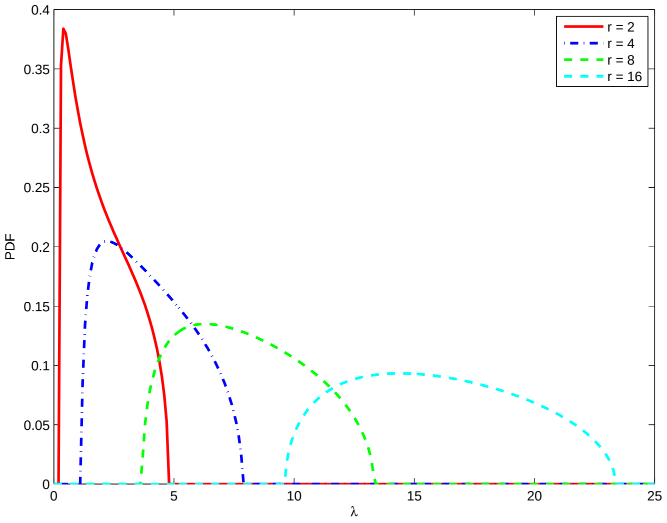

Figure 4-5: PDF of eigenvalues of $\mathbf{G}$ for $M_0 = 4$ factor oversampling and varying oversampling ratios r

We find that for $r > 1$

$$E\left[\frac{1}{\lambda}\right] = \frac{1}{r-1}\left(1 - \frac{1}{rM_0}\right) \tag{4.57}$$

$$E\left[\frac{1}{\lambda^2}\right] = \frac{1}{r^2}\frac{1}{(r-1)^3}\frac{1}{M_0^2}\left(rM_0 - 1\right)\left(r^2 M_0 - 2r + 1\right) \tag{4.58}$$

These statistics are the limiting values as the matrix size goes to infinity. From these equations, we see that as the oversampling factor r increases, the error goes to zero as $1/r$ and variance goes to zero as $1/r^3$.

These expected error values give insight into the geometry of the problem and the added benefit of not restraining the sampling matrices to sinc functions. The expectations for any value of M_0 are bounded between

$$\frac{1}{r} < E\left[\tfrac{1}{\lambda}\right] < \frac{1}{r-1} \tag{4.59}$$

$$\frac{1}{r^2} < E\left[\tfrac{1}{\lambda^2}\right] < \frac{r}{(r-1)^3} \tag{4.60}$$

where the limits are taken for $M_0 = 1$ and $M_0 \to \infty$. For a given oversampling factor r, if the performance is low because M_0 components is too large, then adding M_0 more components, i.e., $r + 1$ oversampling, will achieve the same performance as if $M_0 = 1$ with r held constant.

In Section 4.4, the simulations show that the expected error $1/M_0$-unitary matrices is a lower bound for the $1/M_0$-rate matrices. It remains to be proven that this bound holds for all values of r and M_0; however, such a result would imply that the decreased degrees of freedom in the randomness of each $1/M_0$-rate matrix leads to a higher probability of the skews bunching closer. This skew bunching leads to projection directions being aligned closer together and produces a higher error in the other directions, which dominates estimation performance.

4.3.2 Adding the Subspace Constraint

For a bandlimited input, we find $\mathbf{G}_L = \mathbf{L}^T\mathbf{F}^T\mathbf{F}\mathbf{L}$, where $\mathbf{L}$ is represented through a random matrix with $N\omega_c'$ eigenvalues at one and the rest at zero. This formulation allows us to keep the random matrix representation, without having to represent $\boldsymbol{x}$ through its subspace equivalent representation. The bivariate polynomials for the $\mathbf{G}_L$ are

$$\begin{aligned} L_{m_{\mathbf{G}_L}(z),z} &= (-Mz^2+z^3)M_0m^2+((-2z+\omega_c'z-z^2)M_0M+M^2z+2z^2M_0)m \\ &\quad +M^2-M_0M-M^2\omega_c'-M_0Mz+\omega_c'M_0M+M_0z \qquad (4.61) \\ L_{m_{\mathbf{G}_L^\dagger}(z),z} &= (-z^3M_0M+z^2M_0)m^2+(((2\omega_c'-3)z^2+z)MM_0+2zM(\omega_c'-1)-M^2z^2)m \\ &\quad -((\omega_c'^2-3\omega_c'+2)z+\omega_c'-1)M_0M+(\omega_c'-1)^2M_0+(\omega_c'-2)M^2z \qquad (4.62) \end{aligned}$$

yielding moments

$$\begin{aligned} E\left[\lambda\right] &= \omega_c'r && (4.63) \\ E\left[\lambda^2\right] &= \frac{\omega_c'r}{M_0}(rM_0+M_0\omega_c'-\omega_c') && (4.64) \\ E\left[\frac{1}{\lambda}\right] &= \frac{\omega_c'}{rM_0}\frac{rM_0-\omega_c'}{r-\omega_c'} && (4.65) \\ E\left[\frac{1}{\lambda^2}\right] &= \frac{\omega_c'}{r^2M_0^2}\frac{r^3M_0^2-3\omega_c'r^2M_0+\omega_c'^2r(M_0+2)-\omega_c'^3}{(r-\omega_c')^3}. && (4.66) \end{aligned}$$

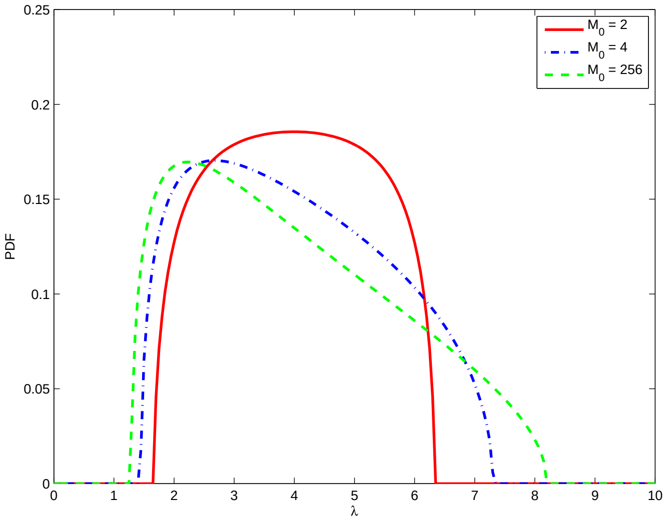

Figure 4-6: PDF of eigenvalues of $\mathbf{G}_L = \mathbf{L}^T\mathbf{F}^T\mathbf{F}\mathbf{L}$ for $r = 4$ factor oversampling and varying numbers of base components M_0. The plot omits an impulse of area $1-\omega_c' = 0.25$ at $\lambda = 0$.

Again, the first and second order moments of the error are bounded w.r.t the oversampling ratio

$$\frac{\omega_c'}{r} < \quad E\left[\tfrac{1}{\lambda}\right] \quad < \frac{\omega_c'}{r-\omega_c'} \tag{4.67}$$

$$\frac{\omega_c'}{r^2} < \quad E\left[\tfrac{1}{\lambda^2}\right] \quad < \frac{\omega_c' r}{(r-\omega_c')^3}. \tag{4.68}$$

The distribution of the eigenvalues for $r = 4$ is plotted in Fig. 4-6. As expected, the PDFs have the same general shapes as the Fig. 4-4, except that the density is reduced at every point due the $1 - \omega_c'$ fraction of eigenvalues at zero.

4.4 Bound Comparison

Within this chapter, we have presented methods for analyzing and bounding the expected error of a distributed sampling system with uniformly distributed skews and a deterministic bandlimited input. The skews are treated as known values and the error is measured through the input estimate error, $\text{tr}(\mathbf{G}^{-1})\sigma_w^2$. We now plot the performance by fixing the number of base components M_0 and testing different oversampling factors.

In Figs. 4-7 and 4-8, the noise error curves are plotted for $M_0 = 2$ and $M_0 = 8$ respectively. The x-axis represents the oversampling ratio and the y-axis represents the scaling

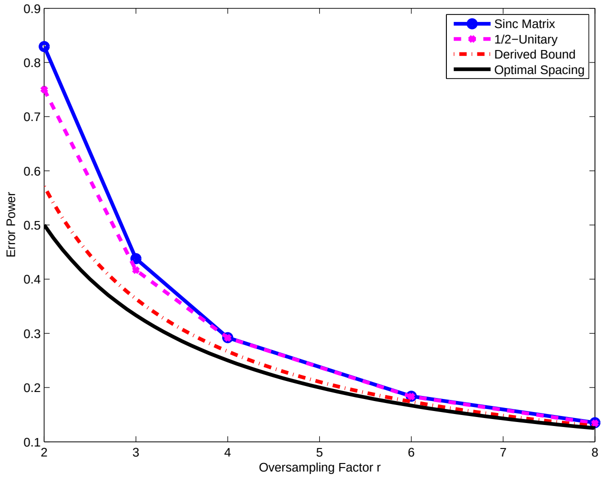

Figure 4-7: Average error versus oversampling factor for $M_0 = 2$ components. Curves plotted for sinc sampling matrices error, $1/M_0$-unitary matrices error, derived lower bound, and minimum possible error.

factor on the error, which can be translated into effective bits loss. In each plot, four curves are shown.

The top curve in solid is the average simulated error of the sinc sampling matrices, where each trial selects the rM_0 skews at random in the $[0, M_0]$ interval. The bottom solid curve represents the best possible error $1/r$, which corresponds to evenly spaced components. The dashed curve is the error when random $1/M_0$-unitary matrices are used instead of the sampling matrices. Finally, the dashed-dot line repesents the lower bound from Section 4.2.1. In Fig. 4-7, this curve is the derived lower bound (4.38); while in Fig. 4-8, this curve is the simulated lower bound (4.24).

As we can see, the curves in both plots quickly converge to $1/r$. The random matrix performance curve provides a better lower bound than the one derived; however, analysis is deficient in proving the random matrices are a lower bound. The variance of the error for these sampling matrices can be simulated and tests have shown higher variance values than for the random $1/M_0$-unitary matrices. Simulations were also carried out for the bandlimited signal case using the $\mathbf{G}_L$ matrices. In this case, the curves had similar shape to the non-bandlimited case with the visible difference being a ω_c' factor reduction of noise.

Figure 4-9 plots the average error scaling for varying M_0 when the oversampling ratio is fixed, $r = 2$. Due to the difficulty of simulating matrices where N and M_0 are both large, the plots are limited to $M_0 = 8$ components. The error power of the optimal spacing is fixed

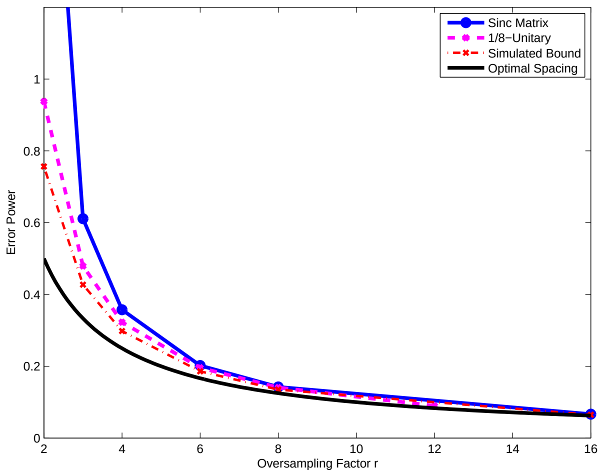

Figure 4-8: Average error versus oversampling factor for $M_0 = 8$ components. Curves plotted for sinc sampling matrices error, $1/M_0$-unitary matrices error, derived lower bound, and minimum possible error.

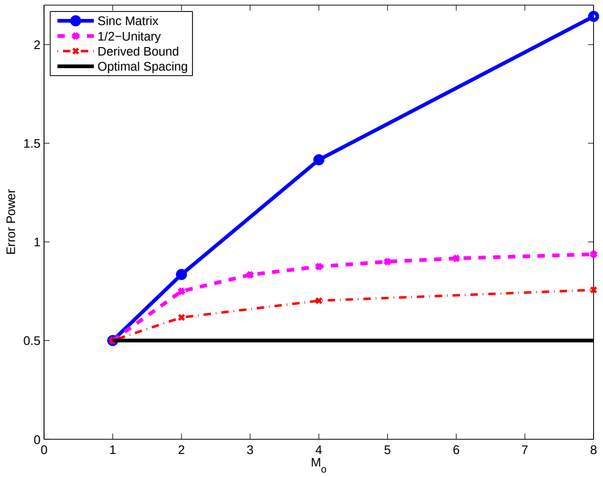

Figure 4-9: Average error versus number of base components for $r = 2$ components. Curves plotted for sinc sampling matrices error, $1/M_0$-unitary matrices error, derived lower bound, and minimum possible error.

at 0.5. The 1/2-unitary matrices converge to a unit error scaling for large M_0; however, the 1/2-rate sinc matrices have a rapidly increasing error.

Although we can see that a significant amount of noise reduction is possible by using oversampling, the previously developed calibration algorithms actually decrease in performance for increasing numbers of sampling components due to the extra mismatch parameters that need to be estimated. In the next chapter, we explore systems with a highly redundant number of sampling components and develop methods for efficient and accurate calibration.

Chapter 5

Extreme Oversampling

Within the previous chapter, we examined the expected noise reduction from increasing the amount of oversampling by using extra sampling components. Using the skew estimation and calibration methods of Chapters 2 and 3, this performance gain is not achievable due to the added difficulty of estimating extra skew parameters. In this chapter, we focus on developing calibration methods for the extremely oversampled system and present methods that exploit the extra sampling components to increase performance and reduce complexity.

In order to ensure a system operates at a required performance target, system engineers often incorporate a design overhead into the system. The amount of overhead is determined by the tradeoff between the added benefit to added cost of the overhead. When the added costs are small, systems are often greatly over-provisioned and a redundant amount of resources are dedicated to tasks.

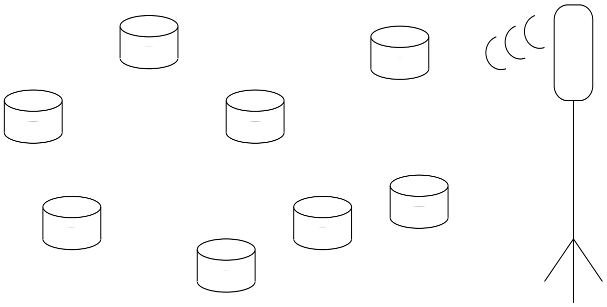

Figure 5-1: Example Sensor Network

In distributed sampling systems, such as sensor networks, the cost of sampling components is often low, allowing the system to use more than the required number of components to sample the input signal. For example, a system that needs five sampling components operating at a certain rate may be use 100 components at that same rate. In this case of extreme oversampling, the base component receives an abundant amount of samples and uses them to estimate the input signal. We examine the properties of extreme oversampling systems and propose signal reconstruction techniques that take advantage of the redundancy in order to reduce complexity.

Because the system is completely uncoordinated, the ordering between the sampling components is not known. The main component receives M sets of periodic samples without any knowledge about the alignment of the samples. This allows for the discrete indexing to be misaligned, e.g., the sample time of $y_2[n]$ lying between the times of $y_1[n+1]$ and $y_1[n+2]$. Although it is possible to apply one of the blind calibration techniques in Section 1.4, the performance is low and the computational complexity is high for the large numbers of components and large skews found in the extreme oversampling case. We develop high-performance calibration algorithms for this setting. In our development, rather than handle the component ordering and calibration at the same time, we treat each problem independently in order to perform signal recovery.

5.1 Problem Formulation

The input $x(t)$ is now modeled as a stationary bandlimited zero-mean Gaussian process with a flat power spectral density

$$S_{xx}(j\Omega) = \begin{cases} \gamma^2/\Omega_c & \text{for } \Omega \in [-\Omega_c, \Omega_c], \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

As described in Section 1.3.1, this flat-spectrum signal class is typical of the narrowband signals found in, e.g., communication systems, where the statistics are constant for a time block; however the results are not sensitive to this aspect of the model.

As in Chapter 4, the sampling period of each individual component is $M_0 T_s$. In the case of extreme oversampling, the number of sampling components (M) greatly exceeds the necessary number for Nyquist sampling (M_0). The average sampling period is T_s/r, where $r = M/M_0$ is the oversampling factor.

The lack of coordination among the nodes leads to arbitrary shifts in the component timings. Because of these arbitrary skews, the output of the ith component is now modeled as

$$y_i[n] = x(nM_0T_s + \tau_i') + w_i[n] \tag{5.2}$$

where the τ_i' represent the overall timing shifts. Without loss of generality, we set $\tau_0' = 0$; thus all sampling timings are relative to the timing of the first component.

Because of the nature of the problem formulation, it is possible that the timing skews τ_i' may be greater than the sampling period of a single component M_0T_s, thus causing a misalignment between components in the discrete indexes. To model this, we separate the overall timing shift into an integer alignment factor m_i and the time skew τ_i, where

$$\tau_i' = m_iM_0T_s + \tau_i \tag{5.3}$$

and $0 \leq \tau_i < M_0T_s$. Note that these skews τ_i are different than the ones in previous chapters because they model an absolute skew from the first component (rather than a deviation from the ideal location). The output of the ith component is

$$y_i[n] = x((n + m_i)M_0T_s + \tau_i) + w_i[n] \tag{5.4}$$

where m_i and τ_i are unknown parameters. Because the τ_i' parameters are arbitrary, we model their respective τ_i skews as having a uniform distribution in the $[0, M_0T_s]$ interval. This random model for the skews will prove useful when estimating the system calibration performance.

Again, the signal recovery problem is to estimate the input $x(t)$, or equivalently $x[n] = x(nT_s)$, as accurately as possible from the component outputs. Figure 5-2 shows an oversampling system with the top lines representing uniform sample timing with $M_0 = 2$ components and the bottom lines representing an example nonuniform sampling pattern with $M = 6$ components.

In the previous sections, we proposed algorithms that estimate the timing skews of the sampling components and subsequently perform reconstruction. It is possible to apply similar techniques in the extreme oversampling case; however, the estimation of all the mismatch parameters is difficult. It is both computationally intensive and more sensitive to noise. For these reasons, we propose the development of alternative strategies that are

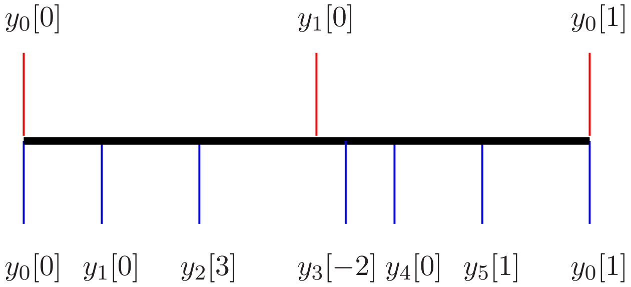

Figure 5-2: Extreme oversampling system with oversampling factor $r = 3$. Top lines represent uniform sample timing with $M_0 = 2$ components. Bottom lines represent an example nonuniform sampling pattern with $M = 6$ components.

tailored to the extreme oversampled system.

There are two major questions in the oversampled system: how to handle the arbitrarily large timing skews and how to efficiently handle the large number of components. We start by decoupling the component alignment problem from the signal recovery problem. The alignment problem is to compute the integers m_i and order the timing skews τ_i. By doing so, we reduce the effective size of each timing shift τ_i'.

After ordering, the timing skews τ_i still exist and the input signal must be estimated from the ordered samples. Subsequently, we develop two methods for mismatch calibration from ordered samples. The first approach uses subset selection in order to reduce the effective number of sampling components. The second approach provides a method for signal estimation through simple filtering operations.

5.2 Ordering Component Samples

In this section, we address the case of unordered sampling and propose methods for ordering the samples. There are a variety of sampling scenarios that deal with unordered sets of samples. One such example is a sensor network, seen in Fig. 5-1. Here, each node collects periodic samples of a common signal and stores these samples into memory. The base station gathers the information but does not have access to when each node started sampling. Another example of unordered sampling is image super-resolution, where multiple images of a common scene are used in order to construct an image with a higher resolution. This 2-D generalization of the sampling problem faces the issue of unordered sampling because images are not always aligned perfectly.

To address the ordering problem, we first examine the case where the integer shifts are zero, $m_i = 0$ for all i, and address methods to handle them later. By doing so, the component signals will be aligned such that the time shift $\tau'_{i,j}$ is reduced to time-skew $\tau_{i,j}$.

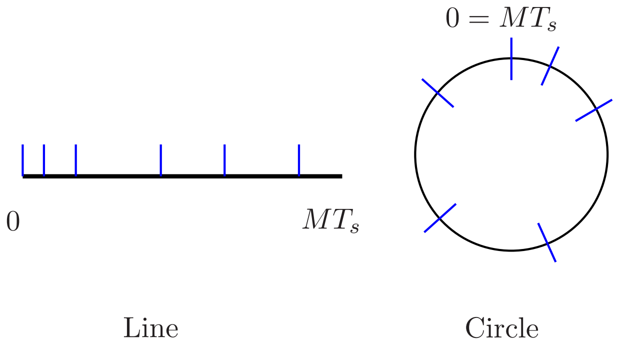

Figure 5-3: 'Line' and 'Circle' methods of visualizing component layout

To visualize the component layout of skews τ_i, we use two methods: 'Line' and 'Circle'. In the traditional 'Line' method, a line represents time, starting at 0 and ending at M_0T_s. The timing skews τ_i of each component can be shown on this line. In this case, components with small skews appear to sample at very different times than components with large skews. In the 'Circle' methods, we wrap the line into a circle, where the starting point equals the ending point. Again, we can show the timing of each component on the circle. The added benefit of this method is that the smallest skew component and the largest skew component are close in distance even though their skews are far apart, which represents the high correlation between their signals. Each method of visualizing the skews will be useful in subsequent development.

There are multiple ways for estimating the ordering of the sampling components. We chose a pairwise approach that measures the relationship between pairs of signals rather than looking at the complete set of component signals simultaneously. By doing so, we reduce the computational complexity.

5.2.1 Generating Correlation and Distance Matrices

To order the components, we first generate a matrix of distance approximations between pairs of components. The method is motivated by the observation that components close in timing will have a high correlation between their samples. Using these matrices, we can generate a global ordering map for the components.

Because the system does not have access to sample timings, we must estimate the relative timing between signals $y_i[n]$ and $y_j[n]$. We define a measurement $g(\cdot,\cdot)$ on pairs of signals that allow us to coarsely estimate the timing difference

$$\hat{\tau}_{i,j} = g(y_i[n], y_j[n]). \tag{5.5}$$

where $\tau_{i,j} = \tau_i - \tau_j$.

In systems where each component samples above the Nyquist frequency, the timing difference can be computed via standard time-delay estimation techniques, where the peak of the convolution is often used [30, 18]. The same techniques are not always possible in systems where individual components sample under the Nyquist frequency.

For a moment, let us try to use a similar convolution estimator for arbitrary bandlimited signals where each component undersamples the input. Because it is not possible to interpolate each received signal $y_i[n]$ in order to perform the continuous time convolution, only discrete shifts can be calculated

$$s[k] = \sum y_i[n] y_j[n-k]. \tag{5.6}$$

However, these values do not provide ample information to determine the time delay. For example, suppose the input $x(t)$ is generated as follows

$$\begin{aligned} x(t) &= \sum x_q[n] \operatorname{sinc}(t/T_s - n) && (5.7)\\ x_q[3n] &= 1 && (5.8)\\ x_q[3n+1] &= 0 && (5.9)\\ x_q[3n+2] &= 1.01 && (5.10) \end{aligned}$$

where T_q is the Nyquist frequency. If we pick components y_0 and y_1 with periods $3T_q$, it is possible that $y_0[n] = 1$ and $y_1[n] = 1.01$ for all n. These two components have a large dot product even though their timing difference is large.

Because the peak of the autocorrelation is not computable with undersampled signals, we develop an alternate estimation method that exploits the stationarity property of our input. Our approach to timing estimation is to assume that the timing skew between two sets of samples has a magnitude proportional to the Euclidean distance between them,

$$|\hat{\tau}_{i,j}| \approx f_d \left(\frac{1}{N} \sum_n (y_i[n] - y_j[n])^2 \right)^{1/2}, \tag{5.11}$$

where f_d is some increasing function. Thus, if two components are close in timing, we expect the distance between their signals will be small. This relationship can be made more precise by focusing on our input class of wide-sense stationary signals. Because the variance is fixed, equation (5.11) can be replaced with skew estimate equation

$$|\hat{\tau}_{i,j}| \approx f_c \left(\frac{1}{N} \sum_n y_i[n] y_j[n] \right), \tag{5.12}$$

where function f_c equals the inverse of the autocorrelation function.

The computation $\sum y_i[n]y_j[n]$ provides an estimate of $R_{x,x}(\tau_{i,j})$. By inverting the autocorrelation value, we form an estimate of the timing difference $\tau_{i,j}$. Other autocorrelation techniques are discussed in [16, 44] for a general nonuniform periodic system; however, the extreme oversampling system allows for greater accuracy due the close spacing of neighboring components.

We briefly examine the example input class of bandlimited Gaussian signals $x(t)$ with power spectral density (PSD)

$$X(j\omega) = \begin{cases} \frac{\pi}{\Omega_c} & |\Omega| < \Omega_c \\ 0 & \text{otherwise} \end{cases}. \tag{5.13}$$

The autocorrelation is given by

$$R_{x,x}(\tau) = \text{sinc}(\Omega_c \tau / \pi) \tag{5.14}$$

as shown in Fig. 5-4.

If we find that the correlation estimate between two components is equal to 0.9, we know that the timing difference between the components (in the absence of noise) is equal to $R^{-1}_{x,x}(0.9) = +/-0.79$. Thus, we have formed a method for estimating the absolute value of the timing difference, which is based off the correlation between components. Because the autocorrelation is symmetric, each absolute timing estimate yields two possibilities for

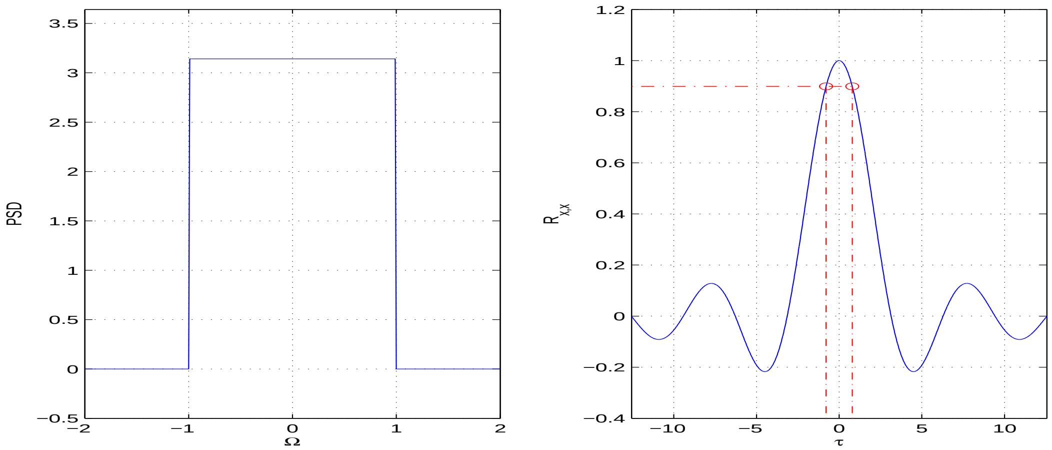

Figure 5-4: Power spectral density (PSD) and autocorrelation R_{xx} for a bandlimited Gaussian signal.

the true timing skew.

This method is accurate for estimating timing differences that are small in magnitude, but we must explore what happens when the components are spaced far apart. Suppose that the correlation is equal to 0.1; in this case, there are multiple absolute timings that could yield this correlation. Rather than dealing with large numbers of possible timing skews, we focus our attention on the correlations that correspond to one absolute timing (two possible timing skews) and ignore all other correlation values. In the next section, we define a range of correlations for which this property holds.

For any signal with a continuous PSD over a finite support, it is possible to show that the autocorrelation function will have a unique peak at $\tau = 0$. Also, the function is non-increasing in a range $[-\tau_c, \tau_c]$ around the peak, i.e. $R_{x,x}(\tau) \leq R_{x,x}(0)$ for $0 \leq |\tau| < \tau_c$. For a given input spectrum, it is necessary to compute the largest coherence range τ_c such that the autocorrelation values of the peak lobe within this interval are greater than the autocorrelation values anywhere outside this interval, i.e. $R_{x,x}(\tau_c) > R_{x,x}(\tau)$ for $\tau > \tau_c$. Therefore, given τ_c, all autocorrelation estimates between $R_{x,x}(\tau_c)$ and $R_{x,x}(0)$ correspond to a single absolute timing, as shown in Fig. 5-5.

The analysis provides a definition of a coherence interval, however no explicit solution is possible. Because each power spectral density has a different autocorrelation, the coherence range τ_c must be calculated from an estimate of the input spectrum. In general, a narrow spectrum corresponds to slower decaying autocorrelation and a larger coherence interval; however, slower decay also implies an autocorrelation estimates that are more sensitive to

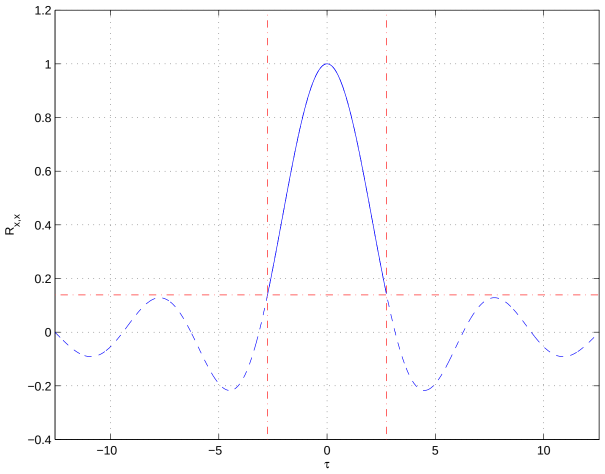

Figure 5-5: Autocorrelation plot of bandlimited Gaussian signal showing the valid coherent range where a autocorrelation value corresponds to at most two timing skews.

noise.

By calculating the coherence range for the input class, we can now order the components. We begin by computing the pairwise correlation between each pair of components i and j. Only the correlations that correspond to timing differences within the coherence range are kept and all others are deleted. This is equivalent to keeping track of only the closest neighbors. Because this method does not consider discrete shifts m_i, we are implicitly using the 'Line' method where components with skews of ϵ and $MT_s - \epsilon$ are not neighbors. From these computations, we can form a correlation matrix

$$[\mathbf{R_L}]_{i,j} = \begin{cases} \sum_n y_i[n]y_j[n] & \text{if } \sum y_i[n]y_j[n] > R_{xx}(\tau_c) \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

by zeroing out any component outside the coherence range. The corresponding distance matrix is

$$[\mathbf{D_L}]_{i,j} = \begin{cases} R_{xx}^{-1}(\sum_n y_i[n]y_j[n]) & \text{if } \sum y_i[n]y_j[n] > R_{xx}(\tau_c) \\ \infty & \text{otherwise} \end{cases} \tag{5.16}$$

where the positive values of the inverse autocorrelation are taken.

The introduction of integer offsets m_i only adds a small amount of complexity. To test whether a pairwise offset $m_{i,j} = m_i - m_j$ is valid, we simply check to see whether the corresponding correlation $\sum y_i[n]y_j[n+m_{i,j}]$ produces a value in the correlation coherence range $[R_{x,x}(\tau_c), R_{x,x}(0)]$. For pairs of components with small $|\tau_{i,j}|$, the correct value of $m_{i,j}$ will result in a high correlation. Any incorrect value of $m_{i,j}$ causes a dramatic reduction in the correlation and can be thrown out. If no value of $m_{i,j}$ produces a high enough correlation, this implies that components i and j are not neighbors.

In order to simultaneously obtain the correlations for multiple values of $m_{i,j}$, it is only necessary to compute the convolution between $y_i[n]$ and $y_j[-n]$. Because different values of $m_{i,j}$ correspond to vastly different $\tau'_{i,j}$ this method is resilient to noise errors. Because skew τ_i ordering is on a finer scale, it is necessary to calibrate for the $m_{i,j}$ parameters first.

When dealing with these integers shift $m_{i,j}$, we are switching to the 'Circle' method of timing skews. Components that have skews on opposite sides of $0, M_0T_s$ point, i.e., $\tau_i - \tau_j + M_0T_s < \tau_c$, will produce high correlation values. Thus, it is necessary to increase these estimates of $m_{i,j}$ by one in order to construct a stable component ordering. With these modified $m_{i,j}$ integer shifts, the corresponding correlation and distance matrices using the 'Circle' method are

$$[\mathbf{R_C}]_{i,j} = \begin{cases} \sum_n y_i[n]y_j[n+m_{i,j}] & \text{if } \sum y_i[n]y_j[n+m_{i,j}] > R_{xx}(\tau_c) \\ \sum_n y_i[n]y_j[n+m_{i,j}-1] & \text{if } \sum y_i[n]y_j[n+m_{i,j}-1] > R_{xx}(\tau_c) \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

and a corresponding distance matrix

$$[\mathbf{D_C}]_{i,j} = \begin{cases} R_{xx}^{-1}(\sum_n y_i[n]y_j[n+m_{i,j}]) & \text{if } \sum y_i[n]y_j[n+m_{i,j}] > R_{xx}(\tau_c) \\ R_{xx}^{-1}(\sum_n y_i[n]y_j[n+m_{i,j}-1]) & \text{if } \sum y_i[n]y_j[n+m_{i,j}-1] > R_{xx}(\tau_c) \\ \infty & \text{otherwise.} \end{cases} \tag{5.18}$$

From these matrices, it is possible to construct a global ordering.

5.2.2 Global Ordering from Pairwise Estimates

In the previous section, we presented a method for generating distance estimates between pairs of component. From these estimates, there are multiple ways to generate a global ordering of the components.

Greedy Nearest Neighbor

We start by presenting a simple greedy algorithm that works in $O(M)$ time in the absence of noise. We use the 'Line' method ($\mathbf{R_L}$ and $\mathbf{D_L}$) and later take into account the circular structure. For each component y_i, we can form a list from matrix $\mathbf{D_L}$ that contains the neighbors ordered by closest distance (highest correlation). No information of the sign of the timing difference is known; the list is only in terms of absolute distance from component i, which ignores whether the components may sample before or after the list component.

Using these M lists of neighbors for the components, we can now form a global ordering. Let us start with an arbitrary component i who has γ_i neighbors in its coherence range ordered as $\{C_{(i,1)}, C_{(i,2)}, ..., C_{(i,\gamma_i)}\}$, where $C_{(i,1)}$ corresponds to the index of the component closest to component i. This component $C_{(i,1)}$ can either sample before or after component i; without loss of generality, we say that it fires after i. Now, the second component $C_{(i,2)}$ either samples before i or after $C_{(i,1)}$. In order to resolve this question, we look at the list corresponding to component $C_{(i,2)}$. If $C_{(i,2)}$ is closer to $C_{(i,1)}$ than it is to i, we determine that it samples after $C_{(i,1)}$; otherwise, it samples before i. Note that if $C_{(i,1)}$ is not on the list of $C_{(i,2)}$, we determine that i is closer since it is guaranteed to be on the list.

After these steps, we know the relative order of components $\{i, C_{(i,1)}, C_{(i,2)}\}$. Likewise we can continue to sequentially add the $\gamma_i - 2$ neighboring components to the ordering by determining whether each component is closer to the first ordered element or the last ordered element. After ordering the list for component i, we continue to component $C_{(i,1)}$ and so on until all components are in the ordering.

Let us look at an example case of $M = 8$ components, with the lists of neighbors for each component already computed.

$$\begin{array}{llll} 1: & 4 & 6 & 2 \\ 2: & 6 & 1 & \\ 3: & 5 & & \\ 4: & 1 & 5 & 6 \\ 5: & 4 & 3 & \\ 6: & 1 & 4 & 2 \end{array} \quad \rightarrow \quad \begin{array}{llll} 4 & 1 & 6 & 2 \\ 1 & 6 & 2 & \\ 3 & 5 & & \\ 5 & 4 & 1 & 6 \\ 3 & 5 & 4 & \\ 4 & 1 & 6 & 2 \end{array} \tag{5.19}$$

Stitching together these lists, we get the global ordering $\{3, 5, 4, 1, 6, 2\}$ for the components.

Therefore, provided that neighbors exist within the τ_c range for each component and no errors exist, we can compute a complete ordering map

$$\alpha = \{\alpha_0, ..., \alpha_{M-1}\}, \tag{5.20}$$

where α_k represents the component number for the k-th ordered component, i.e. $\tau_{\alpha_k} < \tau_{\alpha_l}$ for all $k < l$. Note, it is possible that the component list ordering is reversed but this issue can be resolved through a simple measurement on the signal estimate developed below.

From estimates of the integer alignments m_i and ordering of the time-skews α, we can create the ordered oversampled output $y_{os}[n]$,

$$y_{os}[n] \quad = \quad y_{\alpha_i}\left[\frac{n-i}{M} - m_i\right] \qquad n \pmod{M} = i. \tag{5.21}$$

for $0 \leq i \leq M-1$. This discrete time signal has an average sampling period of T_s/r.

The nearest-neighbors approach has a low computing-time; however, any errors in the neighbor lists causes the method to fail to produce the correct ordering. Errors arise in the form of permutations in the neighborhood lists. There are multiple reasons that this can happen. First, the correlation calculation (5.15) is only an estimate of the correlation. For an N-length block of samples from each component, the variance of this correlation estimate decays as $1/N$. If the error in the estimate is high, a permutation error may occur with another neighbor. Another source of error is the zero slope of the autocorrelation at $\tau = 0$. Any skews in a neighborhood around this point have similar autocorrelation values, making them sensitive to noise. Lastly, although the stationary statistics are modeled as being constant, slight deviations can change the correlation estimates.

Vector ordering

In this section, we introduce an alternate method for global ordering that is more robust to errors. Rather than looking a single pairwise relationship for component i, we use a vector of correlations $\mathbf{r}_{\mathbf{L}i}$, which is the ith row of $\mathbf{R}_\mathbf{L}$. The basic motivation is that components i and j that are close in timing will have corresponding vectors $\mathbf{r}_{\mathbf{L}i}$ and $\mathbf{r}_{\mathbf{L}j}$ that are close in distance because they will have similar correlation values for their neighbors. The algorithm can still be performed in a greedy way, where the first component selects the closest neighbor and so on.

The algorithm is robust to errors in correlation because multiple correlations are being considered simultaneously. Basic simulations were performed to compare the ordering performance of the greedy nearest neighbor method and vector ordering method. Simulations show an increase in performance with vector ordering when larger amounts of noise are present. In cases with high noise, the incorrect decisions made by the greedy nearest neighbor algorithm propagate to produce increasingly inaccurate orderings. The performance of both methods can increase if the 'Circle' correlation and distance matrices are used. The corresponding list representation is invariant to any circular shift.

Advanced ordering methods

Both of the methods presented previously are greedy methods, where the closest measured component is added to the list at each time step. These methods have the benefit of being low complexity but do not fully incorporate the global structure. A variety of other approaches have been used to address similar problems in the past; in particular, we reformulate the sampling component ordering as a traveling salesman problem and a linear ordering problem. Using these alternative formulations, we can draw upon a wide variety of other algorithms that have been developed.

The component ordering problem can be rewritten in a graph form where vertices represent components. An edge exists between vertices i and j if the distance between these components $\mathbf{D_C}_{i,j}$ is finite. The edge lengths are given by their corresponding values in $\mathbf{D_C}$. In this case, the goal of the 'Circle' component ordering is to find a loop in the graph that visits every point and has the minimum overall distance. This formulation is equivalent to the traveling salesman problem (TSP) [35]. Although the geometry of the ordering problem is simpler than the general TSP problem, the noise on the correlation estimates distorts the geometry. Because the TSP problem is non-polynomial (NP-complete), only approximate solutions are possible in polynomial time. These TSP solutions applied to the component ordering problem are expected to be less sensitive to noise since they use a global approach.

An alternative ordering approach is possible through a linear ordering formulation. For any given matrix, the problem of linear ordering aims to compute the permutations of rows and columns in a matrix that maximize the sum of the weights in the upper triangle [1]. This formulation is equivalent to finding a component ordering such that the distance from first to last component is minimized in $\mathbf{D_L}$. Again, this problem is NP-complete and various approximate solutions have been developed.

These complicated methods of addressing the component ordering problem are useful when the amount of oversampling r is extremely high or when noise is present. Although we do not develop any specific algorithm, the reformulation gives insight into other global approaches that can be used for ordering from a distance matrix.

5.2.3 Ensuring Existence of Neighbors

The ordering methods proposed above rely on the assumption that each component has a multiple element list of neighbors within distance τ_c; however, with a random distribution of skews τ_i this can never be guaranteed. In this section, we define a failure event if a component does not have both a left-side and a right-side neighbor. Although the minimum number of components necessary to prevent any failures is $M = M_0T_s/\tau_c$, a significantly larger number of components is required to ensure no failures exist with high probability. We analyze the probability for a failure and how this decreases with the oversampling level r.

In the 'Line' formulation, the first and last components are allowed to have only a right-side and left-side neighbor respectively. Because the 'Line' probability of failure is strictly less than the 'Circle' probability of failure, we focus on upper-bounding the latter. The skews τ_i are random parameters with uniform distributions in the interval $[0, M_0T_s)$. With the 'Circle' method, for any component i, the neighbors locations are random with uniform distribution $[-M_0T_s/2, M_0T_s/2]$.

We use A_i as an indicator variable that denotes the event that component i does not have a right-side neighbor. The probability of this event is

$$P[A_i] = \left(1 - \frac{\tau_c}{M_0T_s}\right)^{M-1} \tag{5.22}$$

By the union bound, the probability event that at least one component does not have a right-side neighbor is given as

$$P[A] \leq M_0 \left(1 - \frac{\tau_c}{M_0T_s}\right)^{M} \tag{5.23}$$

A failure occurs when components have neither left-side nor right-side neighbors. It is clear to see that this event is equivalent to the event A in (5.23). Thus the probability of failure

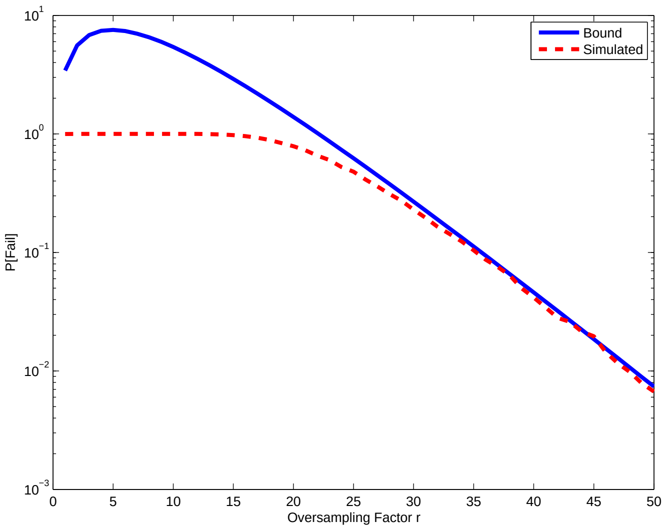

Figure 5-6: Probability of failure for various oversampling factors r in an $M_0 = 4$ system with coherence range $\tau_c/T_s = 0.2$

decays exponentially in M. Fig. 5-6 shows a plot of the probability of failure for varying values of oversampling value r, with $(M_0, T_s, \tau_c) = (4, 1, 0.2)$.

5.3 Efficient Calibration

Using the methods of Section 5.2, we are able to order the samples from all the components; however the exact skew values remain unknown. There exist a variety of applications for which the extreme ordered signal is found naturally. One example is antenna arrays where multiple antennas are lined up to receive a signal. The ordering of the antennas is often known but the spacing may not be uniform.

From these ordered signals, it is still necessary to reconstruct the input signal. Although the effective size of the timing skews has been reduced, the computational complexity of the calibration methods from Chapter 3 is still too high for the large number of sampling components.

We now propose two alternative methods for signal reconstruction in the ordered, extreme oversampled system. The first method uses subset selection to choose an optimal set of components from which to perform calibration. The second method provides a direct reconstruction method through the use of a single lowpass filter.

5.3.1 Subset Selection via Downsampling

Subset selection can be used in a highly oversampled system in order to reduce the computational complexity of input estimation. The goal is to pick the optimal set of M_0 sampling components

$$\beta = \{\beta_0, ..., \beta_{M_0-1}\}, \tag{5.24}$$

where β_k is the index of the kth component selected, and throw out the samples from the other components. Optimality is defined by the set that produces the smallest amount of parameter estimation error. After selecting the subset, the individual component signals are multiplexed to create $z[n]$,

$$z[n] \quad = \quad y_{\beta_i}\left[\frac{n-i}{M}\right] \qquad n \pmod{M_0} = i. \tag{5.25}$$

where we assume that the integer shifts m_i are previously calibrated.

The multiplexed signal has the same sampling rate as the input. We can use it as a direct estimate of the input $\hat{x}[n] = z[n]$, or we can apply a time-skew calibration method like the ones described in the previous chapter. In both cases, choosing components that are evenly spaced in time reduces the deviation from uniform sampling, which reduces input estimation error. An efficient way of subset selection is to use every rth ordered component, resulting in the downsampling of (5.21) by r.

We develop analysis for the performance of this downsampling method by using tools from order statistics [3]. This allows us to bound the distribution of the uncalibrated downsampled signal SNR. In turn, by using least-square calibration, the input/output SNR plots of Section 3.3 will yield the distribution of the calibrated downsampled signal SNR.

For subset selection in the example system of $M_0 = 2$ and $M = 6$ seen in Figure 5-2, it is necessary to select two out of the six components. Suppose we select the first component as the first ordered component, i.e. $\beta_0 = 0$, in the signal $z[n]$. The downsampling approach for subset selection is to select the middle component of the remaining five components as the second component β_1 in the subset signal $z[n]$. This component will have a time-skew whose distribution is determined by the PDF of the median element in five independent samples of a uniform distribution. From this PDF, we can calculate the distribution on the downsampled signal SNR and subsequently the calibrated signal SNR. By increasing the oversampling factor r (for fixed M_0), the distribution of the median time-skew of the $2M-1$ components will concentrate around T_s, such that uniform timing is approximated in $z[n]$, as we shall see in the following analysis.

For the case of $M_0 > 2$, the statistics of the time-skews of the selected components become more complex. In many of these cases, the distributions cannot be explicitly calculated. Instead, we must rely on approximations or simulations to generate performance results. We let $\boldsymbol{\tau}_\beta$ be the vector of skew from the M_0 components selected

$$\boldsymbol{\tau}_\beta = \left[\tau_{\beta_0}\ \tau_{\beta_2}\ \ldots\ \tau_{\beta_{M_0-1}}\right] \tag{5.26}$$

and $\boldsymbol{\xi}$ be the skews normalized by the period of each component

$$\boldsymbol{\xi} = \frac{\boldsymbol{\tau}_\beta}{M_0 T_s}. \tag{5.27}$$

The distribution of the downsampled component skews is

$$f(\boldsymbol{\xi}) = \frac{(M-1)!}{\left(\frac{M}{M_0}-1\right)!^{M_0}} \prod_{i=1}^{M_0} \left(\xi_i - \xi_{i-1}\right)^{M/M_0-1} \tag{5.28}$$

where $\xi_0 = 0$ and $\xi_{M_0} = 1$.

From this PDF, we can compute the moments from the general equation

$$E\left[\prod_{i=1}^{M_0-1} \xi_i^{\alpha_i}\right] = \frac{(M-1)!}{\left((M-1)+\sum_{i=1}^{M_0-1}\alpha_i\right)!} \prod_{i=1}^{ado-1} \frac{\left(i\frac{M}{M_0}-1+\sum_{j=1}^{i}\alpha_j\right)!}{\left(i\frac{M}{M_0}-1+\sum_{j=1}^{i-1}\alpha_j\right)!}. \tag{5.29}$$

For a single variable, this yields

$$E\left[\xi_v\right] = \frac{(M-1)!}{(M-1+1)!}\frac{\left(v\frac{M}{M_0}-1+1\right)!}{\left(v\frac{M}{M_0}-1\right)!} \tag{5.30}$$

$$= \frac{v}{M_0} \tag{5.31}$$

$$E\left[\xi_v^2\right] = \frac{(M-1)!}{((M-1)+2)!}\frac{\left(v\frac{M}{M_0}-1+2\right)!}{\left(v\frac{M}{M_0}-1\right)!} \tag{5.32}$$

$$= \frac{v(vM+M_0)}{M_0^2\,(M+1)} \tag{5.33}$$

$$\mathrm{var}\left[\xi_v\right] = \frac{v(vM+M_0)}{M_0^2\,(M+1)} - \frac{v^2}{M_0^2} \tag{5.34}$$

$$= \frac{-v^2+vM_0}{M_0^2\,(M+1)} \tag{5.35}$$

As expected, the expectation of the timing skews yields uniform spacing between components, and the variance goes to zero for increasing numbers of components M.

The expected sum of the square deviations from uniform spacing is given by

$$\text{Sum Skew Err} = \sum_{v=1}^{M_0-1} Var\left[\xi_v\right] \tag{5.36}$$

$$= \frac{-1}{M_0^2\,(M+1)}\sum v^2 + \frac{M_0}{M_0^2\,(M+1)}\sum v \tag{5.37}$$

$$= \frac{-1}{M_0^2\,(M+1)}\frac{M_0(M_0-1)(2M_0-1)}{6} + \frac{M_0}{M_0^2\,(M+1)}\frac{M_0(M_0-1)}{2} \tag{5.38}$$

$$= \frac{M_0^2+1}{6M_0(M+1)} \tag{5.39}$$

$$\approx \frac{M_0}{6M} \tag{5.40}$$

The number of effective bits is given in (2.83)

$$\text{Output Bits} = \frac{10}{6.02}\log\left(\frac{\sigma_x^2}{\frac{1}{M_0}\sum(\xi_i-\frac{i}{M_0})^2\sigma_d^2+\sigma_w^2}\right). \tag{5.41}$$

With small timing skew errors, the skew noise will dominate the quantization/thermal noise, $\frac{1}{M_0}\sum(\xi_i-\frac{i}{M_0})^2\sigma_d^2 >> \sigma_w^2$. Thus, we find for a B-bit component,

$$\text{Output Bits} \approx B - \frac{10}{6.02}\log\frac{\sigma_d^2}{M_0}\sum(\xi_i-\frac{i}{M_0})^2. \tag{5.42}$$

By Jensen's inequality,

$$E\left[\log \sum (\xi_i - \frac{i}{M_0})^2\right] \leq \log E\left[\sum (\xi_i - \frac{i}{M_0})^2\right] \tag{5.43}$$

$$= \log \frac{M_0^2 + 1}{6M_0(M+1)} \tag{5.44}$$

since $(-\log)$ is a convex function. Thus, we are able to form a lower bound on the expected number of output bits

$$E\left[\text{Output Bits}\right] \approx B - \frac{10}{6.02} E\left[\log \frac{1}{M_0} \sum \xi_i^2 \sigma_d^2\right] \tag{5.45}$$

$$\geq k - \frac{10}{6.02} \log \frac{M_0}{M} \tag{5.46}$$

where $k = B - \frac{10}{6.02} \log(\sigma_d^2/6M_0)$. Thus, we expect the number of bits to grow as $1.66 \log r$, i.e., half bit increase for each double in oversampling factor. The results are presented for the cases of $r = 2$ and $r = 8$ in Section 5.3.3. The effective bits of the signal can be increased further by using the small-skew least-squares calibration method in Chapter 3.

There are many other methods of subset selection which we leave to be explored in future research. These cases include random selection, where each sample is randomly drawn using some prior distribution over the components. Alternatively, one can explore the accuracy of the timing estimates generated by inverting the autocorrelation. Using these estimates, it may be possible to select more evenly spaced components than the simple downsampling method.

Although the downsampling method generates approximately evenly spaced components, it does not exploit the extra sample values gathered in the extreme oversampled system. In the next section, we present an alternative method that uses the extra sample values.

5.3.2 Lowpass Filter Method

Until now, we have provided methods for reducing the number of effective components in the extreme oversampling system; we now develop a signal reconstruction method that uses all the sample values. This low complexity method estimates the input by: ordering the components, multiplexing their outputs, and lowpass filtering the samples. We analyze the performance of this method, which exploits the oversampling in the system.

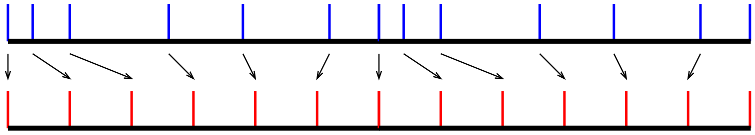

Figure 5-7: Lowpass filter method. The samples obtained from the nonuniform sampling pattern are treated as samples on a uniform grid.

In Figure 5-7, the nonuniform sample times are treated as uniform sample times. The use of all samples shrinks the bandwidth of the signal from ω_c to ω_c/r. The noise introduced through the time-skews in the nonuniform samples is then reduced by filtering the oversampled signal.

To calculate the error of such a scheme, we can re-examine the statistics for each individual component, where $0 \leq i \leq M-1$

$$\begin{aligned} E\left[\xi_i\right] &= \frac{i}{M} && (5.47)\\ E\left[\xi_i^2\right] &= \frac{i(i+1)}{M(M+1)} && (5.48)\\ Var\left[\xi_v\right] &= \frac{v(vM+M_0)}{M_0^2\,(M+1)} - \frac{v^2}{M_0^2} && (5.49)\\ &= \frac{iM-i^2}{M^2(M+1)}. && (5.50) \end{aligned}$$

This yields a total error of

$$\begin{aligned} \text{Sum Skew Error} &= \sum_{i=1}^{M-1} Var\left[\xi_i\right] && (5.51)\\ &= \frac{M(M-1)M/2-(M-1)M(2M-1)/6}{M^2(M+1)} && (5.52)\\ &= \frac{M-1}{6M}. && (5.53) \end{aligned}$$

As we increase the oversampling factor r and therefore increase the number of components M, the error converges to a constant value. The increased oversampling also provides a larger band from which the noise can be filtered; thus providing a noise reduction.

As done previously, the skew error can be used in order to calculated the expected effective number of bits of the lowpass filtered signal. In the next section, we plot the simulation performance for this lowpass filtering method and explore the relationship between oversampling factor r and calibrated signal SNR.

5.3.3 Estimation Performance from Ordered Samples

We now simulate the performance of the two input estimation algorithms. For a given oversampling factor r and base components number M_0, tests were performed by randomly selecting skews from a uniform distribution. The samples were first ordered and then the effective number of bits for each method was calculated. Normalized histograms of performance are plotted in Figs. 5-8 and 5-9.

As expected, simulations show an increase in effective bits for larger oversampling ratios. For both methods, each factor of two increase in r leads to a half bit increase in effective bits. By increasing the base component number M_0 from two to eight, average performance drops by approximately one bit in each case.

In this chapter, we have split the problem of calibration for extreme oversampling systems into many subproblems. Although the goal is low-complexity, the arbitrary sized timing skews introduce difficult challenges. By first ordering the components, the errors that occur are less deleterious than directly trying to estimate the timing skews. Downsampling and filtering the oversampled signal are both efficient ways of estimating the input.

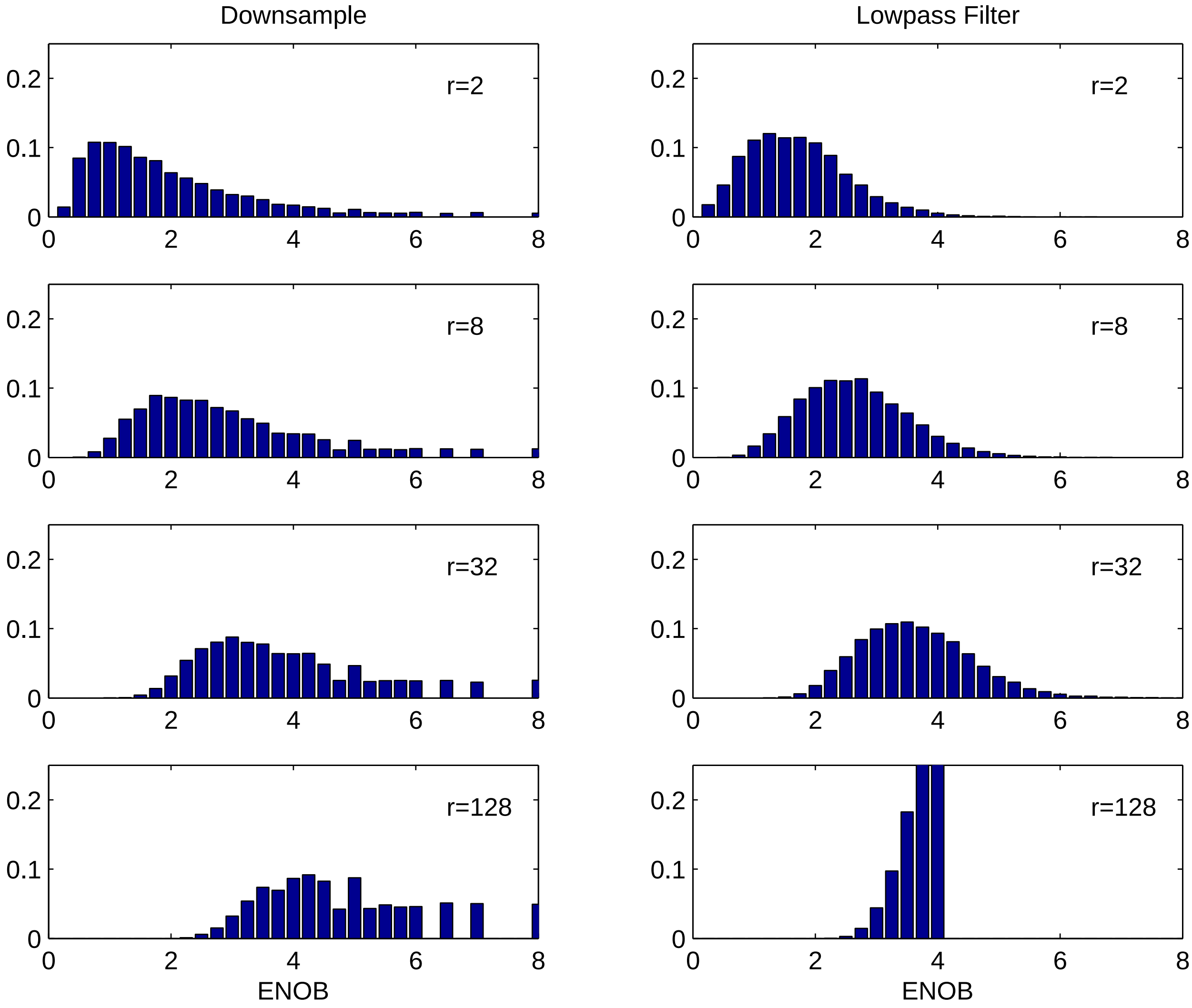

Figure 5-8: Normalized histogram of effective number of bit performance for downsampling method (left) and lowpass filtering method (right) for $M_0 = 8$ system and varying oversampling factors

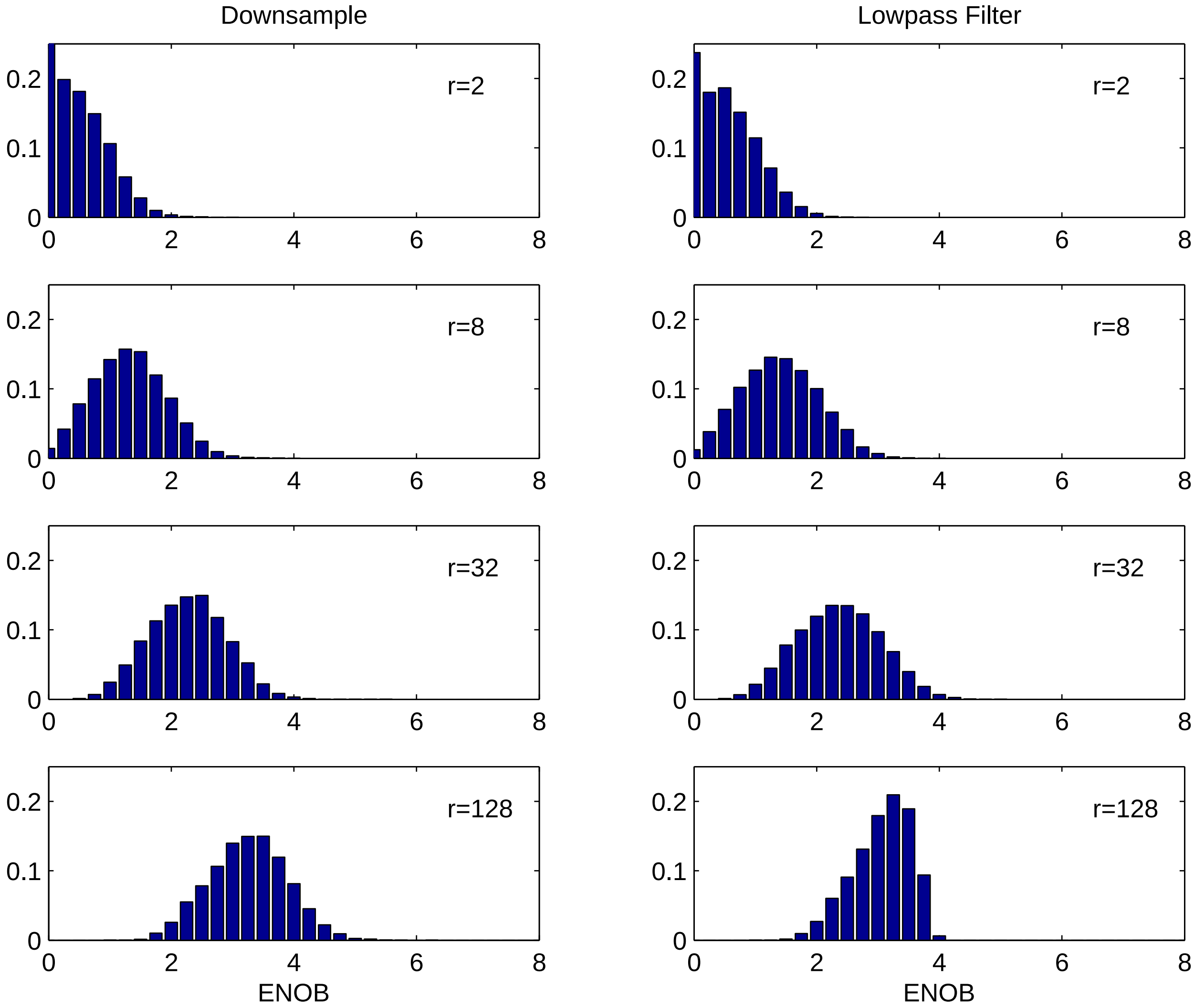

Figure 5-9: Normalized histogram of effective number of bit performance for downsampling method (left) and lowpass filtering method (right) for $M_0 = 8$ system and varying oversampling factors

Chapter 6

Conclusion

In this thesis, we explored the topic of calibration for distributed sampling systems. Although a variety of mismatch effects occur in such systems, we focused on the calibration of component timing skews. Calibration is treated as deterministic parameter estimation problems, requiring unbiased estimates of the input and using no prior on the skews. We examined the problems of mismatch estimation and input reconstruction in such systems and developed new methods for efficient calibration.

6.1 Contributions

In Chapter 2, we explored the problem of signal reconstruction from nonuniform recurrent sampling in many different regimes. To begin, we used Taylor series approximations of the ideal reconstruction in order to generate a reconstruction formula that is linear in terms of the unknown mismatch parameters. The approximation accuracy is high for small timing-skews; and higher orders can be incorporated to increase the accuracy for the larger skews. We also developed a reconstruction for the low-SNR scenario when the signal is oversampled. By restricting the input to a bandlimited subspace in the estimation, the resulting estimator achieves the Cramer-Rao bound. Finally, we presented two iterative algorithms, both which involved projections between subspaces in order to converge upon an estimate of the input signal. Aside from providing a way for reconstruction, these four techniques all give insight into the geometry of the input and output signal space and they motivate new methods for mismatch parameter estimation.

Using the reconstruction linearization, we developed a least-squares method for parameter estimation in Chapter 3. We exploit the linearity of time-skew parameters in high-resolution converters where skews are small relative to the sampling period. The estimation method is motivated by the fact that only the actual values of the timing skews will result in a input estimate that is bandlimited. The least-squares formulation computes the skews that minimize the out-of-band energy of the reconstruction. Due to the linear relationship of the skews in the approximation, the problem can be solved in a closed form. Adaptive filter techniques are presented to make the estimation sequential and reduce the computation and storage requirements. With the addition of unknown gains and relinearization techniques, the method is robust towards handling a broader class of distributed sampling systems. The algorithm shows promising performance and its ability to scale easily makes it an attractive solution for large numbers of components.

Simulation results were presented for various distributed sampling systems. For the 2-ADC system, a single least-squares estimate increased the effective resolution, in one case from five to 12 bits. Using relinearization, the estimation showed increases from 2 to 10 effective bits. In the 16-ADC system, the resolution increased from 6 to 12 bits. Performance changes were also presented for incorrect assumptions on the input spectrum, truncated filters, and gain mismatches. The simulations were limited to fewer than 32 components due to the large numbers of samples necessary to calibrate otherwise.

To understand the limits of calibration performance, we analyzed bounds on the expected error for nonuniform periodic sampling when the skews are distributed uniformly in Chapter 4. A lower-bound was derived for the estimate error from sampling matrices. We compared this to the expected error for random matrices and showed the loss by restricting our sampling matrices to sinc functions. The effects of increasing the oversampling rate through extra converters was also presented.

By using many extra components, a system enters the extreme oversampling regime, where both arbitrary skew sizes and large numbers of components give rise to a new set of problems. In Chapter 5, we developed calibrations algorithms that address this case by separating the problem of component ordering from the problem of signal estimation. Stationary properties of the input are used to estimate the distance between components through the autocorrelation. From these distance estimates, there are a variety of methods to produce a global ordering map of the component timings. The input can then be estimated via subset selection by downsampling or by lowpass filtering and downsampling.

Simulated distributions of performance were presented for both estimation algorithms.

6.2 Future Work

We now consider some directions for future research and development that have been motivated by this thesis.

6.2.1 Hardware Implementation

The least-squares calibration method offers high performance and significantly reduces the complexity compared to any previous method for $M > 4$ components. However, the amount of complexity acceptable for real-time hardware implementation requires further investigation. Modifications to the algorithm to reduce filter size were briefly explored but further analysis must be conducted to optimize the filters. Also, the relinearization methods currently require the computation of more complicated filters. Although the skew estimation can be spread across long periods of time, efficient methods are necessary for implementing the relinearized recovery filter.

6.2.2 Noise Sensitivity

While constructing the least-squares calibration method for estimating the timing skews, we assumed the noise on the input was not large enough to affect the skew estimation. However, for lower resolution converters or higher amounts of thermal or shot noise, the noise can significantly degrade performance. The error propagates through the calibration process; thus understanding the relationship between noise variance and estimate error as well as between estimate error and signal reconstruction error is important for practical system design. Each type of error has a different affect on the calibration process. Thermal or shot noise on the skew estimate can be averaged out with longer block lengths while quantization has a fixed cost. Although some of these effects were analyzed through simulations, a more formal analysis can also provide a better understanding on how to reduce these errors.

6.2.3 Architecture Dependent Nonlinearities

The first order model of the ADC provides a broad umbrella under which many nonlinearities are masked. For many ADC architectures, large benefits can be gained by specifying the major nonlinear effects and why they occur. For example, errors in pipeline converters can cause gaps in the converter output codes. Or in flash converters, errors can arise in the form of bit flips or out-of-sequence bits. In each case, performance can increase for each specific chip design by carefully tailored calibration algorithms. While some errors such as out-of-sequence bits are hard to model, other errors can inspire algorithms which are useful for a broader class of sampling components.

6.2.4 Extremely Fast Calibration

Within the extreme oversampling problem, correlation estimates were measured between each pair of components in order to form a distance matrix. From the matrix, we computed a global ordering of components. The skews were allowed to be larger than a component sampling period. The added computation to handle arbitrary skews makes components ordering almost as challenging as skew estimation. Rather than restrict the skews to being small deviations from uniform timing, as is done in the small-mismatch regime, another useful model is letting the components be unordered within a single sampling interval. The old methods can still be applied, but new approaches also exist and do not require a stationary input. With a focus on speed, we can experiment with other approaches, like a greedy method that continually selects the next closest sample out of the M samples in that interval.

6.2.5 Nonuniform Sampling Periods

Throughout our development, we assumed that the distributed components all sample with the same period. This assumption may not always be true. If the components do not use a common clock signal, the synchronization of clock periods is sometimes equally as difficult as creating uniform timing. While gain mismatch and amplitude offset were treated cursorily, if significant period mismatch exists, it can lead to the largest sources of error. In this case, new calibration methods must be developed to estimate the component period as well as the time skew. A similar modified least-squares approach may be possible, but a low-complexity reconstruction method is necessary.

6.2.6 Compressed Sensing

A large number of similarities exist between the nonuniform sampling problem and the problem of compressive sensing. In compressive sensing, an unknown signal of length N is sparse in some basis $\boldsymbol{\Psi}$, having support $k << N$ in $\boldsymbol{\Psi}$, i.e., $\boldsymbol{x} = \boldsymbol{\Psi}\mathbf{z}$ where $||\mathbf{z}||_0 = k$. The signal is observed through noise-free projections onto vectors in an alternate matrix $\boldsymbol{\Phi}$, i.e., $\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{x}$. The goal is to be able to reconstruct the signal using as few observations (projections) as possible.

Results have shown that by choosing a $\boldsymbol{\Phi}$ that is incoherent with $\boldsymbol{\Psi}$, where coherency is a quantitative measure of the similarity between basis, the signal can be accurately reconstructed with $P = O(k \log(N))$ measurements [8, 13]. This greatly reduces the amount of measurements that need to be made. The theory extends to the bandlimited space where signals are sparse in the frequency domain.

As research progresses for understanding the effects of noise in the compressed sensing problem, there may be insight to be gained about how noise affects input estimation in the distributed sampling setup.

Appendix A

Derivation of Reconstruction Approximations

In this section, we present two derivations of the reconstruction approximation (2.14) based on the small-skew assumption. In the first derivation, we compute the first-order Taylor series of reconstruction equation (2.10) with respect to the timing-skews. The details of the derivation give insight into whether the approximations are valid on a term-by-term basis. In the second derivation, we invert the Taylor series of the output generation equation (2.3). This method obscures the details of the approximation; however it is less complicated and generalizes well for the computation of higher order approximation.

A.1 Approximation via Input Reconstruction Equation

We start by rewriting the problem in discrete-time domain and then compute the approximation of the time-variant reconstruction filter. The discrete time equivalent of (2.10) can be specified by sampling at times $t = nT_s$.

$$\begin{aligned} x[n] &= x(nT_s) && \text{(A.1)} \\ &= \gamma(nT_s) \sum_{\alpha=-\infty}^{\infty} \sum_{i=0}^{M-1} y[M\alpha + i] \frac{a_i(-1)^{\alpha M}}{\pi(n - \alpha M - i - \tau_i/T_s)/M} && \text{(A.2)} \end{aligned}$$

where

$$a_i = \frac{1}{\prod_{k=0,k\neq i}^{M-1} \sin(\pi((i-k)+(\tau_i-\tau_k)/T_s)/M)} \tag{A.3}$$

$$\gamma(nT_s) = \prod_{k=0}^{M-1} \sin(\pi(n-k-\tau_k/T_s)/M) \tag{A.4}$$

In order to produce a first-order approximation of this reconstruction filter, we compute the Taylor series around the point $\boldsymbol{\tau} = 0$.

$$\hat{x}[n] \approx x[n]|_{\boldsymbol{\tau}=\mathbf{0}} + \sum_{j=1}^{M-1} \left.\frac{\partial x[n]}{\partial \tau_j}\right|_{\boldsymbol{\tau}=\mathbf{0}} \tau_j \tag{A.5}$$

To aid our development, we introduce the following variables

$$\nu_{a,b} = \sin(\pi(a-b-\tau_b/T_s)/M) \tag{A.6}$$

$$\nu'_{a,b} = \cos(\pi(a-b-\tau_b/T_s)/M) \tag{A.7}$$

$$\eta_{a,b} = \sin(\pi((a-b)+(\tau_a-\tau_b)/T_s)/M) \tag{A.8}$$

$$\eta'_{a,b} = \cos(\pi((a-b)+(\tau_a-\tau_b)/T_s)/M) \tag{A.9}$$

$$\xi_{a,b} = \sin(\pi(a-b)/M) \tag{A.10}$$

$$\xi'_{a,b} = \cos(\pi(a-b)/M) \tag{A.11}$$

$$\zeta_{a,b,c} = \pi(a-bM-c-\tau_c/Ts)/M \tag{A.12}$$

$$\rho_{a,b,c} = \pi(a-bM-c)/M \tag{A.13}$$

Evaluating the first term of (A.5), we find

$$x[n]|_{\boldsymbol{\tau}=0} = \sum_{\alpha=-\infty}^{\infty}\sum_{i=0}^{M-1} y[M\alpha+i]\frac{\prod_{k=0}^{M-1}\xi_{n,k}}{\prod_{k=0,k\neq i}^{M-1}\xi_{i,k}}\frac{(-1)^{\alpha M}}{\rho_{n,\alpha,i}} \tag{A.14}$$

$$= y[n](-1)^{\alpha M}(-1)^{(n-i)}\lim_{p\to i}\frac{\xi_{p,i}}{\rho_{p,0,i}} \tag{A.15}$$

$$= y[n] \tag{A.16}$$

because the product $\prod_{k=0}^{M-1}\xi_{n,k} = 0$ for all n and the denominators are non-zero for $n \neq \alpha M + i$.

To compute the second term of (A.5), we start by separating the terms that depend on

τ_j in $x[n]$.

$$\begin{align} x[n] &= \sum_{\alpha=-\infty}^{\infty}\sum_{i=0}^{M-1} y[M\alpha+i]\frac{\prod_{k=0}^{M-1}\nu_{n,k}}{\prod_{k=0,k\neq i}^{M-1}\eta_{i,k}}\frac{(-1)^{\alpha M}}{\zeta_{n,\alpha,i}} \tag{A.17}\\ &= \sum_{\alpha=-\infty}^{\infty}\sum_{i=0,i\neq j}^{M-1} y[M\alpha+i]\frac{\prod_{k=0,k\neq j}^{M-1}\nu_{n,k}}{\prod_{k=0,k\neq i,j}^{M-1}\eta_{i,k}}\frac{(-1)^{\alpha M}}{\zeta_{n,\alpha,i}}\left[\frac{\nu_{n,j}}{\eta_{i,j}}\right] \notag\\ &\quad + \sum_{\alpha=-\infty}^{\infty} y[M\alpha+j](-1)^{\alpha M}\left(\prod_{k=0,k\neq j}^{M-1}\nu_{n,k}\right)\left[\frac{1}{\prod_{k=0,k\neq j}^{M-1}\eta_{j,k}}\frac{\nu_{n,j}}{\zeta_{n,\alpha,j}}\right] \tag{A.18} \end{align}$$

Using the following derivatives,

$$\begin{align} \frac{\partial}{\partial\tau_j}\frac{\nu_{n,j}}{\eta_{i,j}} &= -\frac{\pi}{MT_s}\left[\frac{\nu'_{n,j}}{\eta_{i,j}}-\frac{\nu_{n,j}\eta'_{i,j}}{\eta_{i,j}^2}\right] \tag{A.19}\\ \left.\frac{\partial}{\partial\tau_j}\frac{\nu_{n,j}}{\eta_{i,j}}\right|_{\boldsymbol{\tau}=0} &= -\frac{\pi}{MT_s}\left[\frac{\xi'_{n,j}}{\xi_{i,j}}-\frac{\xi_{n,j}\xi'_{i,j}}{\xi_{i,j}^2}\right] \tag{A.20}\\ \frac{\partial}{\partial\tau_j}\frac{1}{\prod_{k=0,k\neq j}^{M-1}\eta_{j,k}}\frac{\nu_{n,j}}{\zeta_{n,\alpha,j}} &= -\frac{\pi}{MT_s}\frac{1}{\prod_{k=0,k\neq j}^{M-1}\eta_{j,k}} \notag\\ &\quad \left[\frac{\nu'_{n,j}}{\zeta_{n,\alpha,j}}-\frac{\nu_{n,j}}{\zeta_{n,\alpha,j}^2}-\frac{\nu_{n,j}}{\zeta_{n,\alpha,j}}\sum_{l=0,l\neq j}^{M-1}\frac{\eta'_{j,l}}{\eta_{j,l}}\right] \tag{A.21}\\ \left.\frac{\partial}{\partial\tau_j}\frac{1}{\prod_{k=0,k\neq j}^{M-1}\eta_{j,k}}\frac{\nu_{n,j}}{\zeta_{n,\alpha,j}}\right|_{\boldsymbol{\tau}=0} &= -\frac{\pi}{MT_s}\frac{1}{\prod_{k=0,k\neq j}^{M-1}\xi_{j,k}} \notag\\ &\quad \left[\frac{\xi'_{n,j}}{\rho_{n,\alpha,j}}-\frac{\xi_{n,j}}{\rho_{n,\alpha,j}^2}-\frac{\xi_{n,j}}{\rho_{n,\alpha,j}}\sum_{l=0,l\neq j}^{M-1}\frac{\xi'_{j,l}}{\xi_{j,l}}\right], \tag{A.22} \end{align}$$

we find

$$\begin{align} \left.\frac{\partial}{\partial\tau_j}x[n]\right|_{\boldsymbol{\tau}=0} &= -\frac{\pi}{MT_s}\left[\sum_{\alpha=-\infty}^{\infty}\sum_{i=0,i\neq j}^{M-1} y[M\alpha+i]\frac{\prod_{k=0,k\neq j}^{M-1}\xi_{n,k}}{\prod_{k=0,k\neq i}^{M-1}\xi_{i,k}}\frac{(-1)^{\alpha M}}{\rho_{n,\alpha,i}}\left[\xi'_{n,j}-\frac{\xi_{n,j}\xi'_{i,j}}{\xi_{i,j}}\right]\right. \notag\\ &\quad \left.\sum_{\alpha=-\infty}^{\infty} y[M\alpha+j](-1)^{\alpha M}\frac{\prod_{k=0,k\neq j}^{M-1}\xi_{n,k}}{\prod_{k=0,k\neq j}^{M-1}\xi_{j,k}}\left[\frac{\xi'_{n,j}}{\rho_{n,\alpha,j}}-\frac{\xi_{n,j}}{\rho_{n,\alpha,j}^2}-\frac{\xi_{n,j}}{\rho_{n,\alpha,j}}\sum_{l=0,l\neq j}^{M-1}\frac{\xi'_{j,l}}{\xi_{j,l}}\right]\right] \tag{A.23} \end{align}$$

We now evaluate this derivative w.r.t. τ_j in two cases.

Case 1: $j \neq n(\text{mod } M)$

In evaluating (A.23), the product

$$\prod_{k=0,k\neq j}^{M-1} \xi_{n,k} = 0 \tag{A.24}$$

because $\xi_{n,\text{mod}(n,M)} = 0$. The denominators in (A.23) are non-zero for this case; thus, the expression reduces to zero,

$$\left.\frac{\partial}{\partial \tau_j} x[n]\right|_{\boldsymbol{\tau}=0} = 0 \tag{A.25}$$

for $j \neq n(\text{mod } M)$.

Case 2: $j = n(\text{mod } M)$:

Let $n = M\beta + j$. Using the fact that

$$\frac{\prod_{k=0,k\neq n(\text{mod } M)}^{M-1} \xi_{n,k}}{\prod_{k=0,k\neq i}^{M-1} \xi_{i,k}} = (-1)^{(n-i-\beta)}, \tag{A.26}$$

and $\xi'_{n,j} = (-1)^{\beta}$ the derivative in this case simplifies to

$$\begin{aligned}\left.\frac{\partial}{\partial \tau_j} x[n]\right|_{\boldsymbol{\tau}=0} &= -\sum_{\alpha=-\infty}^{\infty} \sum_{i=0,i\neq j}^{M-1} y[M\alpha + i] \frac{(-1)^{n-i-\beta+\alpha M+\beta}}{\rho_{n,\alpha,i}} \frac{\pi}{MT_s} \\ &\quad -\sum_{\alpha\neq\beta} y[M\alpha + j] \frac{(-1)^{n-i-\beta+\alpha M+\beta}}{\rho_{n,\alpha,i}} \frac{\pi}{MT_s} + y[M\beta + j] \times 0 \qquad \text{(A.27)} \\ &= -\sum_{m\neq n} y[m] \frac{(-1)^{n-m}}{n-m} \qquad \text{(A.28)} \\ &= -(h * y)[n] \qquad \text{(A.29)}\end{aligned}$$

By plugging (A.16) and (A.29) into (A.5), we form the reconstruction approximation (2.14). In the correction of a single sample, the derived reconstruction method is equivalent to making the approximation that all the neighbors of a sample are on a uniform grid.

A.2 Approximation via Output Generation Equation

We now present an alternative derivation of the reconstruction approximation by computing the Taylor series of the sinc terms in the output generation (2.3) and then inverting this equation. The sinc function at times $t = nT_s + \tau_i$ reduces to

$$\begin{aligned} \mathrm{sinc}(n + \tau_i/T_s) &= \mathrm{sinc}(n) + \tau_i \frac{\partial}{\partial \tau_i} \mathrm{sinc}(t/T_s)\bigg|_{t=nT_s} + O(\tau_i^2) && \text{(A.30)} \\ &= \begin{cases} \tau_i \frac{-1^n}{n} + O(\tau_i^2) & n \neq 0 \\ 1 + O(\tau_i^2) & n = 0 \end{cases} && \text{(A.31)} \end{aligned}$$

The expansion of the sampling equation becomes

$$\begin{aligned} y[n] &= \sum_m x[m]\,\mathrm{sinc}((n-m) + \tau_i/T_s) && \text{(A.32)} \\ &= x[n] + \tau_i (h * x)[n] + O(\tau_i^2) && \text{(A.33)} \end{aligned}$$

for $i = n (\mathrm{mod}\ M)$.

It is easy to see that the first-order approximation of the inverse

$$x[n] \approx y[n] - \tau_i (h * y)[n] \qquad \text{(A.34)}$$

is accurate to $O(\boldsymbol{\tau}^2)$.

A.3 Higher-order Approximations

To attain increasingly accurate approximations, we can add higher-order terms to (A.31). For $i = n (\mathrm{mod}\ M)$, the sampling equations expands to

$$\begin{aligned} y[n] &= \sum_m x[m]\,\mathrm{sinc}((n-m) + \tau_i/T_s) && \text{(A.35)} \\ &= x[n] + \sum_j \frac{1}{j!} \tau_i^j (h_j * x)[n] && \text{(A.36)} \end{aligned}$$

and the filter

$$h_j[n] = \frac{\partial^j}{\partial t^j} \operatorname{sinc}(t/T_s)\bigg|_{t=nT_s}. \tag{A.37}$$

For the second-order expansion, filter $h_1[n]$ is given by (2.15) and filter

$$h_2[n] = \begin{cases} 2\frac{(-1)^{n+1}}{n^2} & n \neq 0 \\ -3.3438 & n = 0 \end{cases} \tag{A.38}$$

The second-order reconstruction is

$$\hat{x}[n] \quad = \quad y[n] - \tau_i(h_1 * y)[n] - \frac{1}{2!}\tau_i^2(h_2 * y)[n] + \tau_i(h_1 * r_1)[n] \tag{A.39}$$

where $r_1[n] = \tau_j(h * y)[n]$ for $j = n(\text{mod } M)$. Expansions of higher orders can be computed in a similar manner, where the reconstruction for the jth order expansion increases by j extra terms from the $(j-1)$th order expansion.

A.4 Relinearization Derivation

When the starting estimate of the timing-skews is non-zero, a more accurate approximation can be achieved by linearizing the reconstruction around the current estimate. To do so, we rewrite the timing skews in terms of the current estimates $\bar{\boldsymbol{\tau}}$ and the error terms $\tilde{\boldsymbol{\tau}}$

$$\boldsymbol{\tau} = \bar{\boldsymbol{\tau}} + \tilde{\boldsymbol{\tau}}. \tag{A.40}$$

While treating $\bar{\boldsymbol{\tau}}$ as a constant in (A.2), we relinearize around $\tilde{\boldsymbol{\tau}} = \mathbf{0}$. The explicit computation of (3.16) is omitted due to its length but is routine and can be solved using any numerical toolbox.

A.5 Gain Derivation

Nonuniform gains among the channels can be modeled as multiplicative factors g_i to each component. To add gain compensation in the reconstruction, the outputs from the ith component $y[M\alpha + i]$ terms in (A.2) are multiplied by $\kappa_i = 1/g_i$, for all $0 \leq i < M$. The new linearization is computed around $\boldsymbol{\tau} = \mathbf{0}, \kappa = \mathbf{1}$,

$$\hat{x}[n] \approx x[n]|_{\boldsymbol{\tau}=\mathbf{0},\kappa=\mathbf{1}} + \sum_{j=1}^{M-1} \left.\frac{\partial x[n]}{\partial \tau_j}\right|_{\boldsymbol{\tau}=\mathbf{0},\kappa=\mathbf{1}} \tau_j + \sum_{j=1}^{M-1} \left.\frac{\partial x[n]}{\partial \kappa_j}\right|_{\boldsymbol{\tau}=\mathbf{0},\kappa=\mathbf{1}} (\kappa_j - 1). \quad (A.41)$$

Because the reconstruction is linear in the gain κ, the first and second terms are equivalent to the skew-only derivation. The third term is

$$\left.\frac{\partial x[n]}{\partial \kappa_j}\right|_{\boldsymbol{\tau}=\mathbf{0},\kappa=\mathbf{1}} = \sum_{\alpha=-\infty}^{\infty} y[M\alpha + j] \frac{\prod_{k=0}^{M-1} \xi_{n,k}}{\prod_{k=0,k\neq j}^{M-1} \xi_{j,k}} \frac{(-1)^{\alpha M}}{\rho_{n,\alpha,j}} \quad (A.42)$$

$$= \begin{cases} y[n] & j = n(\text{mod } M) \\ 0 & \text{otherwise} \end{cases} \quad (A.43)$$

This yields the overall first-order approximation for $i = n(\text{mod } M)$

$$\hat{x}[n] = x[n] - \frac{\tau_i}{T_s}(h * y)[n] + (\kappa_i - 1)x[n] \quad (A.44)$$

$$= \kappa_i x[n] - \frac{\tau_i}{T_s}(h * y)[n]. \quad (A.45)$$

The gain term only affects the $x[n]$ term. Note, in the small gain mismatch domain, the κ_i term can be represented as

$$\kappa_i = \frac{1}{g_i} = \frac{1}{1 + \tilde{g}_i} \approx 1 - \tilde{g}_i \quad (A.46)$$

where $\tilde{g}_i$ is the mismatch. Therefore, estimating the invese gain κ_i is equivalent to estimating the gain directly.

Appendix B

Uniqueness of Timing Skews

In the absence of noise, only the true time-skew parameters $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}$ will yield a reconstructed signal with no out-of-band energy. To show this, we start by presenting the reconstruction equation for $M = 2$ ADCs. We show that for $\hat{\boldsymbol{\tau}} \neq \boldsymbol{\tau}$, the output will not be bandlimited. Although the case of $M > 2$ can be proven in a similar manner, we use approximation (2.14) to simplify the analysis.

In this section, we look at the 2-ADC system and examine the output of the reconstruction filterbank generated by plugging the estimate $\hat{\boldsymbol{\tau}}$ into (2.10). By doing so, we also gain insight into the effects of inaccurate estimates and quantization noise on the recovery performance. First, we calculate the Fourier transform of the actual output of the ADCs. Next, we construct our filterbank based off the estimated parameters. Then, we compute the output of the filterbank and study how the output changes as the estimates vary from their true values.

Using $X(e^{j\omega})$ as Fourier transform of the oversampled input signal, we get the ADC outputs equal to

$$Y_0(e^{j\omega}) = \frac{1}{2}\sum_{k=0}^{1} X(e^{j(\omega-2\pi k)/2}) \tag{B.1}$$

$$Y_1(e^{j\omega}) = \frac{1}{2}\sum_{k=0}^{1} e^{j(\omega-2\pi k)\tau/2} X(e^{j(\omega-2\pi k)/2}) \tag{B.2}$$

for $-\pi \leq \omega \leq \pi$.

These signals $Y_0(e^{j\omega})$ and $Y_1(e^{j\omega})$ are now the inputs to our reconstruction filterbank, which is constructed from (2.10). Let us follow these signals through the filterbank and compute the outputs of each component. We shall use the following notation to represent the two halves (positive/negative frequency) of the Fourier transform:

$$A^{+}(e^{j\omega}) = \begin{cases} A(e^{j\omega}) & \text{for } 0 \leq \omega < \pi \\ 0 & \text{for } -\pi \leq \omega < 0 \end{cases} \tag{B.3}$$

$$A^{-}(e^{j\omega}) = \begin{cases} 0 & \text{for } 0 \leq \omega < \pi \\ A(e^{j\omega}) & \text{for } -\pi \leq \omega < 0 \end{cases} \tag{B.4}$$

$$A(e^{j\omega}) = A^{+}(e^{j\omega}) + A^{-}(e^{j\omega}) \tag{B.5}$$

First, the signals are upsampled by a factor of 2, creating signals S_0 and S_1:

$$S_0(e^{j\omega}) = \frac{1}{2}\left(X^{+}(e^{j(\omega+\pi)}) + X(e^{j\omega}) + X^{-}(e^{j(\omega-\pi)})\right) \tag{B.6}$$

$$\begin{aligned} S_1(e^{j\omega}) = \frac{1}{M}\Big(&X^{+}(e^{j(\omega+\pi)})e^{j(\omega+\pi)\tau} + X(e^{j\omega})e^{j\omega\tau} \\ &+X^{-}(e^{j(\omega-\pi)})e^{j(\omega-\pi)\tau}\Big) \end{aligned} \tag{B.7}$$

for $-\pi \leq \omega \leq \pi$.

Next, the signals $S_0(e^{j\omega})$ and $S_1(e^{j\omega})$ are filtered via filterbank realization of (2.10), $\hat{H}_0(e^{j\omega})$ and $\hat{H}_1(e^{j\omega})$, producing outputs $R_0(e^{j\omega})$ and $R_1(e^{j\omega})$. The filters, which are computed from the estimate $\hat{\boldsymbol{\tau}}$, are given by

$$\hat{H}_0(e^{j\omega}) = \begin{cases} -\frac{1}{2j}2a_0 e^{j\pi\hat{\tau}/\xi} & \text{for } -\pi \leq \omega < 0 \\ \frac{1}{2j}2a_0 e^{-j\pi\hat{\tau}/\xi} & \text{for } 0 \leq \omega < \pi \end{cases} \tag{B.8}$$

$$\hat{H}_1(e^{j\omega}) = \begin{cases} \frac{1}{2j}2a_0 e^{-j\pi\hat{\tau}/\xi} e^{-j2\hat{\tau}\omega/\xi} & \text{for } -\pi \leq \omega < 0 \\ -\frac{1}{2j}2a_0 e^{j\pi\hat{\tau}/\xi} e^{-j2\hat{\tau}\omega/\xi} & \text{for } 0 \leq \omega < \pi \end{cases}. \tag{B.9}$$

The reconstructed signal is the sum of $R_0(e^{j\omega})$ and $R_1(e^{j\omega})$:

$$\begin{aligned} \hat{X}(e^{j\omega}) = \; & \tfrac{a_0}{2j} \; [\, (e^{-j\pi\hat{\tau}/2} - e^{j\pi\hat{\tau}/2} e^{j(\tau-\hat{\tau})\omega}) X^{+}(e^{j\omega}) \\ - \; & (e^{j\pi\hat{\tau}/2} - e^{-j\pi\hat{\tau}/2} e^{j((\tau-\hat{\tau})\omega+\tau\pi)}) X^{+}(e^{j(\omega+\pi)}) \\ - \; & (e^{j\pi\hat{\tau}/2} - e^{-j\pi\hat{\tau}/2} e^{j(\tau-\hat{\tau})\omega}) X^{-}(e^{j\omega}) \\ + \; & (e^{-j\pi\hat{\tau}/2} - e^{j\pi\hat{\tau}/2} e^{j((\tau-\hat{\tau})\omega-\tau\pi)}) X^{-}(e^{j(\omega-\pi)}) \,] \,. \end{aligned} \tag{B.10}$$

The reconstructed signal will have no 'out-of-band' energy if and only if the aliased terms cancel each other out in these bands. This is equivalent to ensuring that

$$(e^{j\pi\hat{\tau}/2} - e^{-j\pi\hat{\tau}/2} e^{j((\tau-\hat{\tau})\omega+\tau\pi)}) X^{+}(e^{j(\omega+\pi)}) \quad = \quad 0 \tag{B.11}$$

Assuming that the aliased signal has signal content in this band, we find that

$$\begin{aligned} (e^{j\pi\hat{\tau}/2} - e^{-j\pi\hat{\tau}/2} e^{j((\tau-\hat{\tau})\omega+\tau\pi)}) &= 0 && \text{(B.12)} \\ -j\pi\hat{\tau}/2 + j((\tau-\hat{\tau})\omega+\tau\pi) &= j\pi\hat{\tau}/2 && \text{(B.13)} \\ (\tau-\hat{\tau})(\omega+\pi) &= 0, && \text{(B.14)} \end{aligned}$$

which is only true iff $\tau = \hat{\tau}$. Thus, if the timing skew estimate is incorrect, the signal is guaranteed to have out-of-band energy.

To generalize to the $M > 2$ component system, we define the error signal $e[n]$ for an arbitrary estimate $\hat{\boldsymbol{\tau}}$ as follows

$$\begin{aligned} e[n] &= \hat{x}[n;\boldsymbol{\tau}] - \hat{x}[n;\hat{\boldsymbol{\tau}}] && \text{(B.15)} \\ &= \begin{cases} 0 & n \pmod{M} = 0 \\ \frac{\hat{\tau}_i - \tau_i}{T_s}\,(h * y)\,[n] & n \pmod{M} = i \end{cases} && \text{(B.16)} \end{aligned}$$

when using approximation formula (2.14). The reconstruction $\hat{x}[n;\hat{\boldsymbol{\tau}}]$ is bandlimited when $e[n]$ is bandlimited, which is only true if $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}$.

Bibliography

[1] D. Adolphson and TC. Hu. Optimal linear ordering. *SIAM J. Appl. Math*, pages 403–423, November 1973.

[2] A. Aldroubi and K. Gröchenig. Nonuniform sampling and reconstruction in shift-invariant spaces. *SIAM Rev.*, 43:585–620, 2001.

[3] N. Balakrishnan and A. Cohen. *Order Statistics and Inference: Estimation Methods.* Academic Press, 1990.

[4] F. J. Beutler. Error-free recovery of signals from irregularly spaced samples. *SIAM Rev.*, 8:328–335, July 1966.

[5] W. C. Black and D. A. Hodges. Time interleaved converter arrays. *IEEE Journal of Solid-State Circuits*, 15:1022–1029, December 1980.

[6] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. *Advances in Order Restricted Inference*, 37:28–47, 1985.

[7] L. Brooks. *Circuits and Algorithms for Pipelined ADCs in Scaled CMOS Technologies.* PhD thesis, Massachusetts Institute of Technology, 2008.

[8] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies. *IEEE Trans. on Information Theory*, 52:5406–5425, December 2006.

[9] I. Daubechies, A. Grossmann, and Y. Meyer. Painless nonorthogonal expansions. *J. Math. Phys.*, 27:1271–1283, 1986.

[10] V. Divi and G. Wornell. Signal recovery in time-interleaved analog-to-digital converters. In *Proc. IEEE ICASSP*, pages 593–596, 2004.

[11] V. Divi and G. Wornell. Scalable blind calibration of timing skew in high-resolution time-interleaved ADCs. *Proc. IEEE ISCAS*, May 2006.

[12] V. Divi and G. Wornell. Bandlimited signal reconstruction from noisy periodic nonuniform samples in time-interleaved adcs. In *Proc. IEEE ICASSP*, pages 3721–3724, 2008.

[13] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52:1289–1306, April 2006.

[14] K. Dyer, D. Fu, S. Lewis, and P. Hurst. An analog background calibration technique for time-interleaved analog-to-digital convertersanalog background calibration technique for time-interleaved analog-to-digital converters. *IEEE Journal of Solid-State Circuits*, 33:1912–1919, December 1998.

[15] A. Edelman and N. R. Rao. *Random Matrix Theory.* Acta Numerica, 2005.

[16] J. Elbornsson, F. Gustafsson, and J.-E. Eklund. Blind adaptive equalization of mismatch errors in time-interleaved A/D converter system. *IEEE Trans. Circuits and Systems*, 51:151–158, January 2004.

[17] Y. C. Eldar and A. V. Oppenheim. Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples. *IEEE Trans. Signal Processing*, 48(10):2864–2875, October 2000.

[18] M. Feder and E. Weinstein. Parameter estimation of superimposed signals using the em algorithm. *IEEE Trans. Signal Processing*, 36:477–489, April 1988.

[19] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1996.

[20] J. D. Gorman and A. O. Hero. Lower bounds for parametric estimation with constraints. *IEEE Trans. Info Theory*, 36:1285–1301, November 1990.

[21] V. K. Goyal, J. Kovacevic, and J. A. Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, pages 203–233, May 2001.

[22] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 2002.

[23] S. Huang and B. C. Levy. Adaptive blind calibration of timing offset and gain mismatch for two-channel time-interleaved ADCs. *IEEE Trans. Circuits and Systems I*, 53:1278–1288, June 2006.

[24] S. Huang and B. C. Levy. Blind calibration of timing offsets for four-channel time-interleaved ADCs. *IEEE Trans. Circuits and Systems I*, 54:863–876, April 2007.

[25] S. Jamal, D. Fu, M. Singh, P. Hurst, and S. Lewis. Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter. *IEEE Trans. Circuits and Systems*, 51:130–139, January 2004.

[26] Y.-C. Jenq. Digital spectra of nonuniformly sampled signals: a robust sampling time offset estimation algorithm for ultra high-speed waveform digitizers using interleaving. *IEEE Trans. Instrumentation and Measurement*, 39:71–75, February 1990.

[27] H. Jin and E. K. F. Lee. A digital-background calibration technique for minimizing timing-error effects in time-interleaved ADCs. *IEEE Trans. Circuits, Systems II*, 47:603–613, July 2000.

[28] H. Johansson and P. Löwenborg. Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters. *IEEE Trans. on Signal Proc.*, 50:2757–2767, November 2002.

[29] H. Johansson, P. Löwenborg, and K. Vengattaramane. Least-squares and minimax design of polynomial impulse response fir filters for reconstruction of two-periodic nonuniformly sampled signals. *IEEE Trans. Circuits and Systems I*, 54:877–888, April 2007.

[30] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust., Speech, Signal Processing*, pages 320–327, August 1976.

[31] A. Kumar, P. Ishwar, and K. Ramchandran. On distributed sampling of bandlimited and non-bandlimited sensor fields. *Proc. Information Processing in Sensor Networks*, pages 89–98, 2004.

[32] N. Kurosawa, H. Kobayashi, K. Maruyama, H. Sugawara, and K. Kobayashi. Explicit analysis of channel mismatch effects in time-interleaved ADC systems. *IEEE Trans. Circuits Syst. I, Theory Appl*, 51:261–271, March 2001.

[33] J. Kusuma and V. K. Goyal. Delay estimation in the presence of timing noise. *IEEE Trans. Circuits Syst. II, Express Briefs*, to appear 2008.

[34] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay – tools for fractional delay filter design. *IEEE Signal Processing Mag.*, 13:30–60, January 1996.

[35] E. Lawler. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley, 1985.

[36] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian. Analog-to-digital converters: A review of the past, present, and future. *IEEE Signal Processing Magazine*, pages 69–77, November 2005.

[37] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons Inc., 1969.

[38] F. A. Marvasti. *Nonuniform Sampling: Theory and Practice*. Springer, 2001.

[39] P. Marziliano and M. Vetterli. Reconstruction of irregularly sampled discrete-time bandlimited signal with unknown sampling locations. *IEEE Trans. Signal Processing*, 48(12):3462–3471, December 2000.

[40] R. Penrose. A generalized inverse for matrices. *Proc. of the Cambridge Philosophical Society*, 51:406–413, 1955.

[41] A. Petraglia and S. K. Mitra. Analysis of mismatch effects among A/D converters in a time-interleaved waveform digitizer. *IEEE Trans. Instrumentation and Measurement*, 40:831–835, October 1991.

[42] R. Rao and A. Edelman. The polynomial method for random matrices. *Foundations of Computational Mathematics*, October 2007.

[43] D. Seidner and M. Feder. Noise amplification of periodic nonuniform sampling. *IEEE Trans. Signal Processing*, pages 275–277, January 2000.

[44] M. Seo, M. J. W. Rodwell, and U. Madhow. Blind correction of gain and timing mismatches for a two-channel time-interleaved analog-to-digital converter. *39th Asilomar Conf. on Signals, Systems and Computers*, pages 1121–1125, October 2005.

[45] M. Seo, M. J. W. Rodwell, and U. Madhow. Blind correction of gain and timing mismatches for a two-channel time-interleaved analog-to-digital converter: experimental verification. *Proc. IEEE ISCAS*, pages 3394–3397, May 2006.

[46] T. Strohmer and J. Tanner. Fast reconstruction algorithms for periodic nonuniform sampling with applications to time-interleaved ADCs. *Proc. IEEE ICASSP*, April 2007.

[47] T. Strohmer and J. Xu. Fast algorithms for blind calibration in time-interleaved analog-to-digital converters. *Proc. IEEE ICASSP*, April 2007.

[48] C. Vogel. The impact of combined channel mismatch effects in time-interleaved ADCs. *IEEE Trans. Instrumentation and Measurement*, 54:415–427, February 2005.

[49] C. Vogel. A frequency domain method for blind identification of timing mismatches in time-interleaved ADCs. *Norchip Conf.*, pages 45–48, November 2006.

[50] G. Xu, H. Liu, L. Tong, and T. Kailath. A least-squares approach to blind channel identification. *IEEE Trans. Signal Processing*, 43:2982–2993, December 1995.

[51] K. Yao and J. B. Thomas. On some stability and interpolatory properties of nonuniform sampling expansions. *IEEE Trans. Circuit Theory*, pages 404–408, December 1967.

[52] J. Yen. On nonuniform sampling of bandwidth-limited signals. *IRE Trans. on Circuit Theory*, 3:251–257, December 1956.