Mitigating Timing Noise in ADCs through Digital Post-Processing

by

Daniel Weller

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

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Daniel Weller, MMVIII. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author . . . .
Department of Electrical Engineering and Computer Science
May 7, 2008

Certified by . . . .
Vivek K Goyal
Assistant Professor
Thesis Supervisor

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

Mitigating Timing Noise in ADCs through Digital Post-Processing

by

Daniel Weller

Submitted to the Department of Electrical Engineering and Computer Science
on May 7, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

The accuracy of analog-to-digital converters (ADCs) is greatly affected by the uniformity of the times at which samples of the analog input signal are taken. The error in the sample times, known as jitter, or timing noise, has a non-linear, time-varying and input-dependent effect on the sample values, as opposed to the additive noise usually considered. At present, the error due to jitter is minimized through the use of low-jitter clocks, limiting the suitability of such ADCs for low-power applications. This thesis investigates the problem of mitigating the effects of jitter through digital post-processing of the samples, which would allow ADCs to use less accurate clocks without compromising accuracy.

This thesis equates mitigating jitter with estimating the parameters of a bandlimited input signal based on samples collected in the presence of timing noise. Two approaches are considered: classical, observation model-driven estimation, and Bayesian estimation that incorporates a prior model of the signal parameters. For both approaches, algorithms are derived that achieve lower mean-squared-error (MSE) by taking the non-linear effect of the jitter into account. In the non-random case, iterative approximations to the maximum likelihood estimator are developed, including an Expectation-Maximization algorithm. To bound the MSE of such algorithms, the unbiased Cramér-Rao lower bound is approximated using Gauss-Hermite quadrature. For the Bayesian approach, a Taylor series-based estimator and several variants on the Gibbs sampler that all approach the Bayes least squares estimate are designed. These estimators are compared in performance to the optimal linear estimators derived without taking jitter into account. The proposed algorithms are shown to tolerate significantly more jitter than the baseline linear algorithms. Applications of these results and extending these algorithms to correcting spatial uncertainty are discussed briefly as well.

Thesis Supervisor: Vivek K Goyal
Title: Assistant Professor

Acknowledgments

Throughout the course of this research project in my first two years at MIT, especially during the writing of this thesis, I have benefited from the encouragement, experience, and example of others. First, I would like to thank my research supervisor, Professor Vivek Goyal, who challenged me to consider old problems in new ways. Without his support and guidance, I would not have even finished the title page.

I would also like to thank the other members of the STIR group: Julius Kusuma, whose investigations into estimating the parameters of FRI signals motivated my own into bandlimited signals; Lav, Adam, Vinith, John, and Ha, who provided ample encouragement and criticism along the way; and visiting professors José Moura and Byeungwoo Jeon, whose advice helped me tremendously. Outside my own research group, I would like to thank Vincent Tan of SSG, whose work motivated my use of the Gibbs sampler, and Sourav, Dennis, and Tom of DSPG, who also provided welcome support and illuminated new directions for my research. Professors Munther Dahleh, my graduate counselor, and Alan Oppenheim, who supervised my first-year seminar in Electrical Engineering and Computer Science, were invaluable in helping me to adapt to a new school, a new city, and life as a graduate student.

I could not have done any of this without the support of my family and friends. Thank you, Mom and Dad, for bearing with me and helping me see the big picture, that all this hard work now will pay off one day. Brian, thank you so much for being such a supportive brother – your comments and suggestions, especially those regarding quadrature, were incredibly helpful. I would also like to thank my grandparents and the rest of my family for their continuing support. I could not thank you all enough!

To all my friends, in the EECS GSA, the Sidney-Pacific house council, and Grad Hillel, as well as those I've met under more casual circumstances, thank you for helping me maintain my sanity, and making sure that my life at MIT is not "all research, all the time." Good luck to all of you with the rest of grad school, and I hope I will see much more of you than I've had time for the last few months. Eric, Rezy, Sai-Hei, Carlos, and Gustavo, thanks for putting up with me as a roommate

these past couple years. Again, thank you all for the fun times and memories – these moments I will remember long after the details of this thesis vanish from my mind.

Finally, since there's no way I could remember to thank everybody, thank you to all of you who have had a part in encouraging and supporting me to this point. I hope I can return the favor one day.

My work at MIT was supported in part by the Office of Naval Research (ONR) through a National Defense Science and Engineering Graduate Fellowship, and by Analog Devices Corporation.

Contents

Chapter 1

Introduction

Sampling is essential to digital signal processors, communication receivers, and digital control systems to represent analog signals digitally for analysis or further processing. In general, these generated samples are susceptible to two kinds of error: amplitude error and error in the timing of the samples. Mitigating the amplitude error is a well-studied problem, and simple approaches such as noise shaping are well-known to effectively mitigate this type of noise. The timing error, also known as jitter, was studied previously in the 1960s, by Balakrishnan [1], Brown [6], Liu and Stanley [24], and others. However, at present, the timing error is often corrected at the source, using a clock with relatively low phase noise, rather than through any kind of signal processing.

The primary motivation for revisiting the mitigation of jitter is to enable analog-to-digital converters (ADCs) to use less accurate clocks more suitable for ultra low-power applications. As the digital components in mixed-signal systems continue to shrink, minimizing the power consumption of the analog portion of the system becomes more important. According to Lee and Hajimiri in [21], the power consumed by an ADC is proportional to factors such as the desired accuracy and the rate that samples are collected. The intuition behind such an observation is rather simple: using conventional algorithms, the presence of jitter substantially increases the MSE for a given number of bits of accuracy, thereby requiring effectively more bits of data to achieve the same desired MSE. The notion of connecting MSE to the number of bits used to

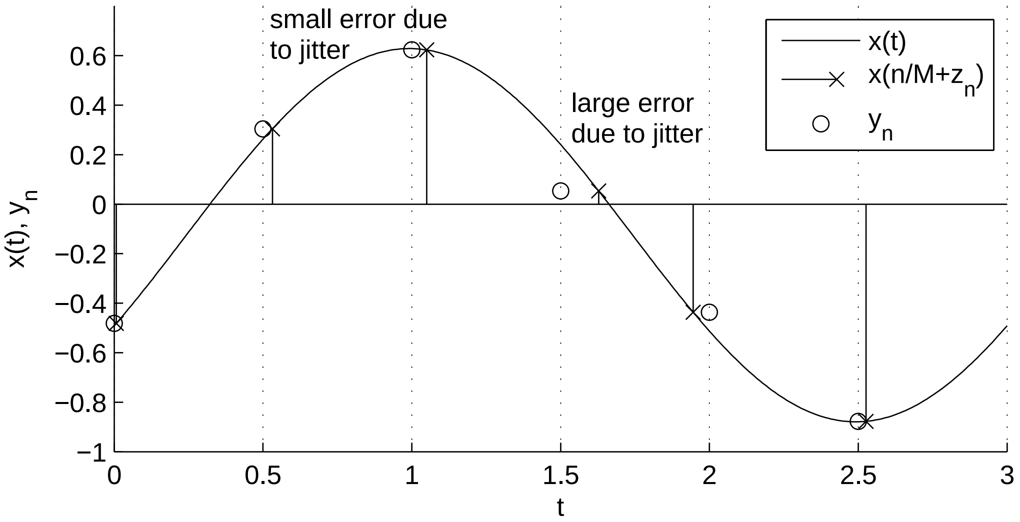


Figure 1-1: The effect of jitter in sampling a signal $x(t)$. The error introduced in the samples y_n by the jitter z_n can range from negligible to rather large.

represent a sample is more natural in the context of scientific notation: if a voltage were measured to be known to six decimal places, its precision would be $\pm 0.5\mu V$. However, if the noise floor suddenly increased by 20 dB, the measured voltage would only be known to five decimal places. To regain the original measurement precision of $\pm 0.5\mu V$, the device would have to be ten times more sensitive than before.

The effect of jitter can be described mathematically in terms of either signal-to-noise ratio (SNR) or effective number of bits (ENOB). In [5], Brannon uses a linear jitter model that ties the effect of the jitter to the maximum slew rate of the bandlimited input signal. The sampled signal is degraded because its frequency spectrum is the convolution of the jittered sampling impulse train's distorted frequency spectrum with that of the input signal. The output SNR, accounting for jitter only, is

$$\mathrm{SNR(dB)} = -20\log_{10}(\Omega_B \sigma_z) + \mathrm{constant}, \tag{1.1}$$

where Ω_B is the maximum non-zero frequency of the bandlimited signal, and σ_z is the standard deviation of the jitter. More generally, Brannon provides the output SNR relationship when thermal effects are included:

$$\mathrm{SNR(dB)} = -20\log_{10}\left((\Omega_B\sigma_z)^2 + \left(\frac{1+\epsilon}{2^N}\right)^2 + \left(\frac{\sigma_w}{2^N}\right)^2\right)^{1/2} + \text{constant}, \qquad (1.2)$$

where N is the number of bits, σ_w is the standard deviation of the thermal (additive) noise, and ϵ is the average differential nonlinearity (DNL) of the ADC. When the additive noise is sufficiently small, the effect of the jitter dominates, and (1.2) is appproximately equal to (1.1). According to [21], the power dissipated is proportional to the SNR, which is defined in that article as the ratio of mean-square signal voltage to mean-square noise voltage. Thus, for large enough σ_z,

$$P_{diss} \propto 1/(\Omega_B\sigma_z)^2. \qquad (1.3)$$

Alternatively, in [31], Walden demonstrates that reducing the standard deviation of the jitter by 50% increases the effective number of bits by one. (Brannon also briefly references this phenomenon.) Each additional bit is equivalent to doubling the root-mean-square (rms) accuracy of the output, as described before. According to the tradeoff presented in [30] for high-speed ADCs,

$$\frac{\text{Speed} \times (\text{Accuracy (rms)})^2}{\text{Power}} \approx \text{constant}. \qquad (1.4)$$

The consequence of both (1.3) and (1.4) is that doubling the standard deviation of the jitter reduces the power dissipated by a factor of four.

Employing a more tolerant post-processing method, a clock generator with high jitter can replace a more exact clock, and the ADC will still achieve the same desired performance, but on a much lower power budget. In addition, being able to utilize a clock signal with substantial amounts of jitter would be expected to have additional benefits, similar to how [22] and [23] advocate modulating the power supply signal to reduce electromagnetic interference across the chip.

Timing noise is a consequence of statistical variations in the edge-to-edge period of the clock signal. Such variation is due to random fluctuations of the temperature or voltage of the oscillator, as well as the responsiveness of the sample-and-hold circuit. Jitter, unlike the typically studied additive white noise, both varies over time and is non-linear in its effect on the samples. In [21], Lee and Hajimiri use a simple sinusoid to explain the time-varying nature of jitter. Additionally, the effect of jitter is signal-dependent. For instance, a slowly varying signal, such as a ramp function or a low-frequency sinusoid, would not be affected much by jitter. However, jitter would have a much greater effect on the samples of a high-frequency sinusoid (see Figure 1-2). Thus, attempting to categorize jitter as additive white noise independent of the signal itself greatly oversimplifies the problem, and the jitter error will be poorly mitigated as a result.

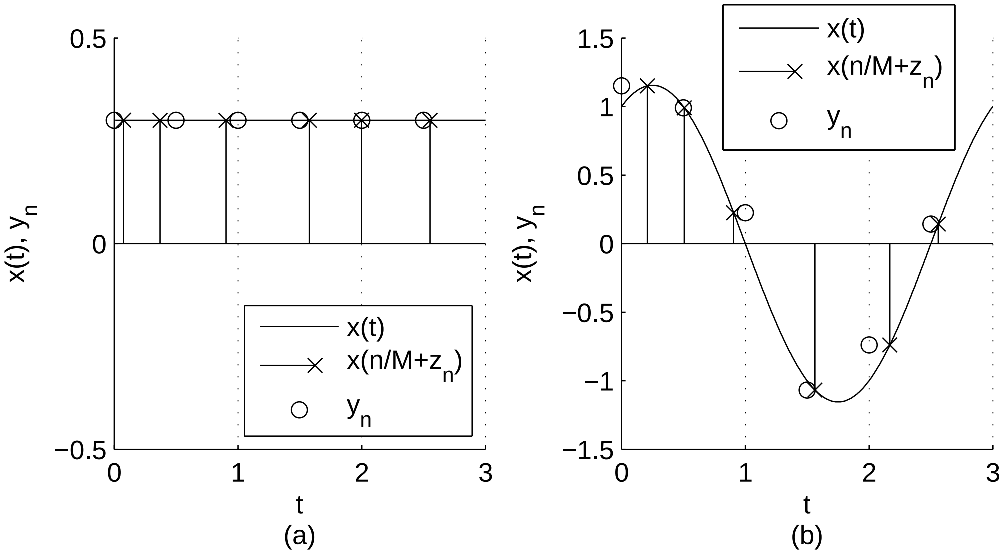


Figure 1-2: Signal dependent, non-linear, time-varying nature of jitter. Jitter affects the samples y_n of (a) flat and (b) sinusoidal signals $x(t)$ differently.

In this thesis, jitter is incorporated into the observation model more consistently with its natural definition, i.e. uncertainty in the sample times.

1.1 Formulating the Estimation Problem

A bandlimited signal with bandwidth $1/T$ can be specified using the sinc function and its integer shifts as a cardinal basis ($x(kT) = x_k$):

$$x(t) = \sum_{k=-\infty}^{\infty} \operatorname{sinc}\left(\frac{t}{T} - k\right) x_k. \tag{1.5}$$

Similarly, a periodic bandlimited signal with period KT can be specified using the periodic-sinc function $\operatorname{psinc}_K(t) = \frac{\sin(\pi t)}{K \sin(\pi t/K)}$. The resulting basis is also a cardinal one:

$$x(t) = \sum_{k=0}^{K-1} \operatorname{psinc}_K\left(\frac{t}{T} - k\right) x_k. \tag{1.6}$$

Without loss of generality, we may assume for the rest of this thesis that $T = 1$. Given K parameters x_k, sampling such a signal at oversampling rate M yields exact samples

$$y_n = \sum_{k=0}^{K-1} \operatorname{psinc}_K\left(\frac{n}{M} - k\right) x_k. \tag{1.7}$$

Two sources of noise are present in this problem: additive noise and jitter. The additive noise, represented by w_n, is considered to be zero-mean white Gaussian noise, with constant known variance σ_w^2. For simplicity, the jitter z_n is also assumed to be zero-mean white Gaussian noise, with constant known variance σ_z^2. Incorporating both noise sources yields the observation model

$$y_n = \sum_{k=0}^{K-1} \operatorname{psinc}_K\left(\frac{n}{M} + z_n - k\right) x_k + w_n. \tag{1.8}$$

Written in matrix form,

$$\mathbf{y} = \mathbf{H}(\mathbf{z})\mathbf{x} + \mathbf{w}, \tag{1.9}$$

where $[\mathbf{H}(\mathbf{z})]_{n,k} \triangleq \operatorname{psinc}_K(n/M + z_n - k)$. Denote by $p(x)$, $p(y; x)$, and $p(y|x)$, the probability density function (pdf) of x, the pdf of y parameterized according to the deterministic variable x, and the pdf of y conditioned on the random variable x, respectively. In this thesis, the random variable(s) associated with a pdf are given implicitly unless otherwise explicitly stated. Framing the problem in standard estimation-theoretic manner, mitigating the effect of both additive and timing noise is equivalent to finding the best estimate of the signal parameters $\mathbf{x}$ given the observations $\mathbf{y}$, according to some criterion, such as mean-squared error. In this thesis, the signal parameters are assumed to be either 1) deterministic and unknown, or 2) independent and identically distributed (iid) uniformly between -1 and 1. Both noise sources are assumed to be independent of both the parameters $\mathbf{x}$ and each other. This formulation differs from the problem of compensating for unknown timing offsets in time-interleaved ADCs, as presented in [11]: the jitter on each sample is independent, whereas the time-interleaved ADC problem assumes that the timing offset for each ADC remains a constant parameter of each ADC.

1.2 Outline

The remainder of this thesis is divided into four chapters. In the next chapter, general background is provided on the different types of estimators investigated and the mathematical and theoretical concepts used in the later chapters. In the two chapters that follow, both the non-random and Bayesian estimation problems are explored. For the non-random case, iterative algorithms are developed to approximate the most likely values of the parameters and contrast these algorithms' performance against the basic linear estimator that does not consider jitter. Also, a lower bound on the unbiased MSE is approximated and compared against these estimators. In the Bayesian case, the best linear estimator is derived and compared to the best linear estimator if jitter were not included in the observation model. Then, the optimal Bayesian estimator is approximated using a variety of numerical and statistical techniques; their performance is contrasted with the no-jitter model's optimal linear estimator. In the final chapter, the overall usefulness of the proposed algorithms is evaluated, and future directions and applications are discussed.

Chapter 2

Background

This thesis takes two perspectives on the same basic estimation problem: guessing the signal parameters $\mathbf{x}$ from a sequence of observations $\mathbf{y}$. The first views the parameters as deterministic, yet unknown quantities. In this setup, there is no prior knowledge about $\mathbf{x}$, and estimation methods must rely exclusively on the data. The second way to approach this problem is to assume a prior distribution on the signal parameters, e.g. iid Uniform$(-1, 1)$ or iid Normal(μ, σ^2), where

$$\text{Uniform}(a, b) = U(x; a, b) \triangleq \begin{cases} \frac{1}{b-a} & a \leq x \leq b, \\ 0 & \text{otherwise} \end{cases}, \tag{2.1}$$

and

$$\text{Normal}(\mu, \sigma^2) = \mathcal{N}(x; \mu, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}. \tag{2.2}$$

Then, a wide variety of Bayesian estimation methods can be used. In this chapter, different estimators are introduced in their general forms for both the deterministic (non-random) and the Bayesian cases. Discussion specific to the observation model will be postponed to the following chapters.

Both of these approaches are complicated by the fact that the likelihood function (and the posterior density) does not have a closed form. In particular, for the non-random case,

$$p(\mathbf{y};\mathbf{x}) = \int_{\mathbb{R}^N} \mathcal{N}(\mathbf{y};\mathbf{H}(\mathbf{z})\mathbf{x},\sigma_w^2\mathbf{I})\mathcal{N}(\mathbf{z};\mathbf{0},\sigma_z^2\mathbf{I})\,d\mathbf{z}, \tag{2.3}$$

and for the Bayesian case,

$$p(\mathbf{y}|\mathbf{x}) = \int_{\mathbb{R}^N} \mathcal{N}(\mathbf{y};\mathbf{H}(\mathbf{z})\mathbf{x},\sigma_w^2\mathbf{I})\mathcal{N}(\mathbf{z};\mathbf{0},\sigma_z^2\mathbf{I})\,d\mathbf{z}. \tag{2.4}$$

To overcome this difficulty, this integral may be approximated in several ways, including using a Taylor series expansion or using Gauss-Hermite quadrature, a method of numerical integration.

Even with such approximations, non-standard probability distributions will need to be manipulated. As a way around this, various stochastic methods can be used to great advantage. Basic methods such as importance sampling and rejection sampling are introduced, along with more sophisticated Markov chain Monte Carlo (MCMC) methods like Gibbs sampling and slice sampling. Properly utilizing these methods, useful approximations to the Bayes Least Squares (BLS) estimate and the Cramér-Rao lower bound can be computed.

2.1 Non-Random Parameter Estimation

Fundamentally consistent with the frequentist philosophy that all the information necessary to perform estimation is contained in the observation model, non-random parameter estimation assumes nothing about the unknown parameters. In the absence of knowledge of a prior distribution that generated the unknown parameters, the mean-square-error (MSE) is difficult to minimize, since it explicitly relies on the true value of the unknown parameters. However, numerous methods have been developed to both provide a lower bound on the MSE for unbiased estimators as well as generate reasonable estimates for the unknown parameters. The Cramér-Rao lower bound, the best unbiased linear estimator, and the maximum-likelihood estimator are all such methods explored in this thesis.

2.1.1 Cramér-Rao Lower Bound

The Cramér-Rao lower bound on the error variance of unbiased estimators relates the minimum error covariance an unbiased estimator can achieve to the Fisher information matrix; i.e.

$$\mathbb{E}\left[(\hat{\mathbf{x}}(\mathbf{y})-\mathbf{x})(\hat{\mathbf{x}}(\mathbf{y})-\mathbf{x})^T\right] \geq \mathbf{I}_{\mathbf{y}}(\mathbf{x})^{-1}, \tag{2.5}$$

where "$\geq$" is in the positive-semidefiniteness sense. In turn, this implies that the trace of $\mathbf{I}_{\mathbf{y}}(\mathbf{x})^{-1}$ is a lower bound on the minimum achievable MSE for an unbiased estimator for any particular value of $\mathbf{x}$. The Fisher information matrix can be expressed in a couple different ways:

$$\mathbf{I}_{\mathbf{y}}(\mathbf{x}) \triangleq \mathbb{E}\left[\left(\frac{\partial l(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}}\right)\left(\frac{\partial l(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}}\right)^T\right] = -\mathbb{E}\left[\frac{\partial^2 l(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}\partial \mathbf{x}^T}\right], \tag{2.6}$$

where $l(\mathbf{x};\mathbf{y}) \triangleq \ln p(\mathbf{y};\mathbf{x})$ is the log-likelihood function.

An unbiased estimator that satisfies (2.5) with equality is termed efficient. Such an efficient unbiased estimator $\hat{\mathbf{x}}_{\text{eff}}(\mathbf{y})$ exists if and only if

$$\hat{\mathbf{x}}_{\text{eff}}(\mathbf{y}) = \mathbf{I}_{\mathbf{y}}(\mathbf{x})^{-1}\frac{\partial l(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}} + \mathbf{x}, \tag{2.7}$$

and the estimator is valid (constant with respect to the true value of the parameters $\mathbf{x}$) [18].

When there is no jitter, i.e. $\mathbf{z} = \mathbf{0}$ in (1.9), the observation model is linear, and therefore, an efficient estimator exists that is itself linear:

$$\hat{\mathbf{x}}_{\text{eff},\mathbf{z}=\mathbf{0}}(\mathbf{y}) = (\mathbf{H}(\mathbf{0})^T\mathbf{H}(\mathbf{0}))^{-1}\mathbf{H}(\mathbf{0})^T\mathbf{y}. \tag{2.8}$$

The minimum achievable MSE, which is the error variance of the efficient estimator, is equal to

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{\text{eff},\mathbf{z}=\mathbf{0}}(\mathbf{y})-\mathbf{x}\|_2^2\right] = \sigma_w^2 \operatorname{tr}\left((\mathbf{H}(\mathbf{0})^T\mathbf{H}(\mathbf{0}))^{-1}\right). \tag{2.9}$$

The Cramér-Rao lower bound only applies to the MSE of unbiased estimators. It is often possible (and desirable) to develop a biased estimator with an even lower MSE.

2.1.2 Linear Estimators

A linear estimator is specified as $\hat{\mathbf{x}}_L(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are fixed. This estimator is linear in the data $\mathbf{y}$.

When the observation model is linear in $\mathbf{x}$, i.e. $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$, it is possible to construct an unbiased linear estimator:

$$\hat{\mathbf{x}}_L(\mathbf{y}) = \mathbb{E}\left[\mathbf{H}\right]^{+}\left(\mathbf{y} - \mathbb{E}[\mathbf{w}]\right). \tag{2.10}$$

When $\mathbf{H}$ is a fixed known matrix, the observation model is linear, and the efficient estimator formula in (2.7) applies here. However, in most cases, the linearity constraint prevents the estimator from being efficient. In fact, when the observation model includes a random matrix $\mathbf{H}$, such as $\mathbf{H}(\mathbf{z})$, the linear estimator can be expected to have a suboptimal MSE since the effect of such perturbations is accounted for by only the mean of the matrix.

2.1.3 Maximum-Likelihood (ML) Estimation

Another logical approach to estimating a deterministic parameter using the observation model is to determine the value of $\mathbf{x}$ that maximizes the likelihood of observing the acquired data. Stated mathematically,

$$\hat{\mathbf{x}}_{ML}(\mathbf{y}) = \arg\max_{\mathbf{x}} p_{\mathbf{Y}}(\mathbf{y};\mathbf{x}) = \arg\max_{\mathbf{x}} l(\mathbf{x};\mathbf{y}). \tag{2.11}$$

Several problems arise when attempting to perform this optimization directly. First of all, the (log-)likelihood function needs to be available. Oftentimes, this is the case; however, the likelihood function featured in this research problem does not have a closed form, which makes maximizing the likelihood more difficult. Secondly, if the objective function is not separable with respect to $\mathbf{x}$, the optimization can become intractable. Finally, a non-concave log-likelihood function can have many local maxima, so finding the global maximum analytically can be extremely difficult.

For example, consider the known-jitter observation model, $\mathbf{y} = \mathbf{H}(\mathbf{z}^*)\mathbf{x} + \mathbf{w}$ with $\mathbf{z}^*$ known. The log-likelihood function is

$$l(\mathbf{y}; \mathbf{x}) = -\frac{1}{2\sigma_w^2}(\mathbf{y} - \mathbf{H}(\mathbf{z}^*)\mathbf{x})^T(\mathbf{y} - \mathbf{H}(\mathbf{z}^*)\mathbf{x}) - \text{constants} \tag{2.12}$$

By orthogonality, (2.12) is maximized when

$$\mathbf{x} = \hat{\mathbf{x}}_{ML,\mathbf{z}=\mathbf{z}^*}(\mathbf{y}) = \mathbf{H}(\mathbf{z}^*)^+\mathbf{y} \tag{2.13}$$

where $\mathbf{H}(\mathbf{z}^*)^+$ is the pseudoinverse, assuming that the choice of $\mathbf{z}^*$ does not reduce the $\mathbf{H}(\mathbf{z}^*)$ matrix to less than full column rank. This estimator is actually efficient (and linear). In fact, it is true that when an efficient estimator exists, that estimator also maximizes the log-likelihood function [18]. In addition, even if an efficient estimator does not exist, the ML estimator is asymptotically efficient (the error covariance converges to the Cramér-Rao bound as $M \to \infty$) as long as the Cramér-Rao bound exists [18].

For the observation model with jitter, the resulting likelihood function is too complex to maximize directly. To reduce the difficulty of the ML estimation problem, several iterative approaches are considered, including the EM algorithm, described next.

2.1.4 Expectation-Maximization (EM) Algorithm

The EM algorithm, as summarized by Bilmes in [3] and described in detail by Dempster, Laird, and Rubin in [10], iteratively maximizes the EM algorithm by augmenting the vector of observations with "hidden data" ($\mathbf{z}$). The hidden data is usually chosen so that it would trivialize the optimization problem if known. The augmented vector of observations is termed the "complete data;" the original observations $\mathbf{y}$ are called the "incomplete data."

"EM" refers to the maximization of an expectation performed in each iteration.

Consider the expected value of the log-likelihood of the complete data for a new guess of $\mathbf{x}$, with respect to the likelihood of the complete data given the observed data $\mathbf{y}$ and the previous iteration's guess for $\mathbf{x}$:

$$Q(\mathbf{x};\hat{\mathbf{x}}^{(i)}) = \mathbb{E}\left[\log p(\mathbf{y},\mathbf{z};\mathbf{x}) \mid \mathbf{y};\hat{\mathbf{x}}^{(i)}\right]. \tag{2.14}$$

Then, the new guess for $\mathbf{x}$ is the value that maximizes (2.14):

$$\hat{\mathbf{x}}^{(i+1)} = \arg\max_{\mathbf{x}} Q(\mathbf{x};\hat{\mathbf{x}}^{(i)}). \tag{2.15}$$

If maximizing this function is too difficult, it suffices to simply increase it; this modification produces a "Generalized EM" algorithm. Why does the EM algorithm work? Following the most general derivation in [10], begin with the relation between the likelihood of the incomplete and complete data,

$$\log p(\mathbf{y};\mathbf{x}) = \log p(\mathbf{y},\mathbf{z};\mathbf{x}) - \log p(\mathbf{z}|\mathbf{y};\mathbf{x}). \tag{2.16}$$

Taking the expected value of both sides with respect to $p(\mathbf{z}|\mathbf{y};\mathbf{x}')$, the left side remains simply the log-likelihood of the observed data, since it does not depend on the incomplete data $\mathbf{z}$:

$$\log p(\mathbf{y};\mathbf{x}) = \underbrace{\mathbb{E}[\log p(\mathbf{y},\mathbf{z};\mathbf{x}) \mid \mathbf{y};\mathbf{x}']}_{u(\mathbf{x})} \underbrace{-\mathbb{E}[\log p(\mathbf{z}|\mathbf{y};\mathbf{x}) \mid \mathbf{y};\mathbf{x}']}_{v(\mathbf{x})}. \tag{2.17}$$

As a consequence of Jensen's inequality,

$$\mathbb{E}[\log p(\mathbf{z}|\mathbf{y};\mathbf{x}) \mid \mathbf{y};\mathbf{x}'] \leq \mathbb{E}[\log p(\mathbf{z}|\mathbf{y};\mathbf{x}') \mid \mathbf{y};\mathbf{x}'], \tag{2.18}$$

with equality if and only if $p(\mathbf{z}|\mathbf{y};\mathbf{x}') = p(\mathbf{z}|\mathbf{y};\mathbf{x})$ almost everywhere. Thus, $v(\mathbf{x})$ in (2.17) will become less negative than before. To increase the log-likelihood, it suffices to simply choose a value of $\mathbf{x}$ that increases (or even better, maximizes) $u(\mathbf{x})$.

Repeat with the new guess of $\mathbf{x}$ and continue until $\hat{\mathbf{x}}$ converges to a stationary point of the likelihood function. If the likelihood function has only one such point, the EM algorithm converges to the ML estimate. However, the likelihood function is rarely concave; therefore, the choice of initial conditions is very important to the success of this algorithm. Although the EM algorithm is guaranteed to converge under very mild conditions [10], the rate of convergence depends on the choice of complete data. In [17], Herzet and Vandendorpe assert that the rate of convergence is worst when the Cramér-Rao lower bound for the incomplete data is much greater than the CRB for the complete data set.

2.2 Bayesian Estimation

The effectiveness of non-random parameter estimation is inherently limited without any prior knowledge of the generating distribution of the unknown parameters. The Bayesian approach assumes knowledge of the prior distribution, and it utilizes this additional information to better estimate the value of the parameters.

In this thesis, it is assumed that the parameters are iid Uniform$(-1, 1)$, essentially asserting that given no data, any parameter value between -1 and 1 is equally likely to be correct. Of course, in the absence of such an assertion, one can still construct a prior that will suit the estimation problem to enable Bayesian inference. Two justifiable approaches are to use either a least informative prior or a conjugate prior.

The least informative prior is commonly defined to be the prior on $\mathbf{x}$ that maximizes the information gained from observing the data $\mathbf{y}$; it is the prior that maximizes the mutual information $I(\mathbf{X}; \mathbf{Y})$ [29]. However, except for a few special cases, the least informative prior is not known and is difficult to find. A maximum entropy model (the prior that maximizes $H(\mathbf{X})$) or Jeffreys prior is commonly used instead for practical reasons. The Jeffreys prior is defined for a single parameter as $p(x) = \frac{1}{\zeta}\sqrt{I(x|\mathbf{y})}$ where $I(x|\mathbf{y})$ is the conditional Fisher Information, and $\zeta \triangleq \int \sqrt{I(x|\mathbf{y})}\, dx$ is a normalization constant.

The conjugate prior is used mainly to ensure tractability of the posterior density; it guarantees that the posterior distribution has the same form as the likelihood function. However, since the likelihood function itself is intractable, using a conjugate prior (even if it could be found) offers no benefit.

2.2.1 BLS Estimation

The objective of our estimation problem is to minimize the MSE of the estimator

$$\hat{\mathbf{x}}_{BLS}(\mathbf{y}) \triangleq \arg\min_{\hat{\mathbf{x}}(\cdot)} \mathbb{E}\left[(\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x})^T(\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x})\right]. \tag{2.19}$$

Equivalently, the BLS estimator is simply the expected value of the posterior; i.e.

$$\hat{\mathbf{x}}_{BLS}(\mathbf{y}) = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \frac{\int \mathbf{x}p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\,d\mathbf{x}}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\,d\mathbf{x}}. \tag{2.20}$$

Using the law of iterated expectation on (2.20), it is easy to see that the BLS estimator is unbiased.

In addition to minimizing the MSE, the BLS estimator satisfies the property that the estimation error is orthogonal to any function of the data (including the BLS estimator itself):

$$\mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}}_{BLS}(\mathbf{y}))\mathbf{f}(\mathbf{y})^T] = \mathbf{0}, \quad \forall \mathbf{f}(\cdot). \tag{2.21}$$

As a consequence, the BLS estimator has the minimum error covariance (in the positive-definiteness sense), and another estimator has equal error covariance if and only if the estimator is equal to the BLS estimator plus a constant.

In the case of a Gaussian linear observation model, where $\mathbf{x}$ and $\mathbf{w}$ have Normal distributions and $\mathbf{H}$ is known, the BLS estimator is linear and also maximizes both the likelihood and posterior density functions. Thus, in the known jitter case, for Gaussian $\mathbf{x}$,

$$\hat{\mathbf{x}}_{BLS,\mathbf{z}=\mathbf{z}^*}(\mathbf{y}) = \mathbf{H}(\mathbf{z}^*)^+\mathbf{y}. \tag{2.22}$$

When the parameters ($\mathbf{x}$) are uniform instead of Gaussian, the BLS estimator is no longer linear.

2.2.2 LLS Estimation

The Linear Least Squares (LLS) estimator is defined to be the estimator that minimizes the MSE over the set of all linear (affine) estimators. The LLS estimator can be expressed in terms of the first and second moments of the observation model:

$$\hat{\mathbf{x}}_{LLS}(\mathbf{y}) = \mathbf{\Lambda}_{\mathbf{xy}}\mathbf{\Lambda}_{\mathbf{y}}^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{y}}) + \mathbf{m}_{\mathbf{x}}. \tag{2.23}$$

From the above expression, it is easy to see that the LLS estimator is also unbiased. In addition, the LLS estimator has the following orthogonality property: (analogous to the BLS orthogonality property stated in (2.21))

$$\mathbb{E}\left[(\mathbf{x} - \hat{\mathbf{x}}_{LLS}(\mathbf{y}))(\mathbf{F}\mathbf{y} + \mathbf{g})^T\right] = \mathbf{0}, \quad \forall \mathbf{F}, \mathbf{g}. \tag{2.24}$$

The LLS estimator has the minimum error covariance in the family of linear estimators as well. This error covariance can also be expressed solely in terms of the second-order statistics of the Bayesian model:

$$\mathbf{\Lambda}_{LLS} = \mathbf{\Lambda}_{\mathbf{x}} - \mathbf{\Lambda}_{\mathbf{xy}}\mathbf{\Lambda}_{\mathbf{y}}{}^{-1}\mathbf{\Lambda}_{\mathbf{xy}}{}^{T}. \tag{2.25}$$

Since $\mathbf{\Lambda}_{LLS} \geq \mathbf{\Lambda}_{BLS}$, approximating the BLS estimate, which will perform better on average, is the focus of the Bayesian portion of this thesis.

2.3 Taylor Series and Numerical Integration

Throughout this thesis, integrals of non-standard functions, usually involving periodic sinc functions, along with weighting functions, usually a Gaussian kernel, need to be evaluated. One general method to consider involves approximating the non-weighting function part of the integrand with a Taylor series expansion and integrating via moments. Another more problem-specific approach that provides more advantageous convergence properties approximates the integrand with orthogonal polynomials chosen based on the weighting function. In the following discussion of approximate integration methods, consider the integration problem of finding $\mathbb{E}[f(\mathbf{x})]$ for some function $f(\mathbf{x})$. These numeric integration methods can be contrasted to the stochastic approximation methods found in the next section for solving the same problem by sampling from a distribution $p(\mathbf{x})$.

2.3.1 Integration via Taylor Series Expansion

Given that $f(x)$ is continuously differentiable on the open ball (a, b), it can be expressed in terms of a convergent power series for all $x \in [a, b]$:

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{2.26}$$

Taylor's theorem [27] states that there exists some $x^* \in (a, b)$ such that

$$f(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k + \frac{f^{(n)}(x^*)}{n!}(x - x_0)^n, \forall x \in [a, b]. \tag{2.27}$$

This theorem can be easily extended to the multivariate case.

Now, suppose we wish to integrate $\int_a^b f(x)p(x)\,dx$. This integral can be approximated by applying Taylor's theorem, and the approximation error can be bounded. Consider the partial Taylor series expansion $f^*(x)$ about $x = x_0$:

$$f^*(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{2.28}$$

Denote the remainder term in (2.27) $R_n(x - x_0)^n$. Then, the integration error can be bounded in terms of R_n:

$$\begin{aligned} \left| \int_a^b f(x)p(x)\,dx - \int_a^b f^*(x)p(x)\,dx \right| &= \left| \int_a^b R_n(x - x_0)^n p(x)\,dx \right| && (2.29) \\ &\leq |R_n| \int_a^b |x - x_0|^n p(x)\,dx. && (2.30) \end{aligned}$$

In the open ball (a, b), the error term is bounded by the maximum absolute value of the nth derivative of f: $|R_n| \leq M_n \triangleq \sup_{x \in (a,b)} |f^{(n)}(x)|$. Assuming the nth derivative is also finite in this range, the supremum exists, and is itself finite. Thus,

$$\left| \int_a^b f(x)p(x)\,dx - \int_a^b f^*(x)p(x)\,dx \right| \leq M_n \int_a^b |x-x_0|^n p(x)\,dx. \tag{2.31}$$

Herein lies the problem with approximating such an integral with Taylor's theorem. The error is proportional to the size of the interval, and in many cases, it scales rather poorly. For instance, when $f(x)$ is a periodic function, Taylor series approximation attempts to fit a polynomial that goes to infinity to a bounded function. Such an approximation, while it converges as n goes to infinity, requires at least as many terms as there are local extrema in the interval to give a good approximation everywhere on the interval.

2.3.2 Gauss Quadrature Method

To compute $\mathbb{E}[f(x)]$, we can consider the general framework $\int_a^b f(x)w(x)\,dx$, where the weighting function $w(x)$ used is the pdf $p_X(x)$. For many such weighting functions, a sequence of orthogonal polynomials $p_0(x), p_1(x), \ldots, p_n(x), \ldots$ can be derived such that

$$\int_a^b w(x)p_m(x)p_n(x)\,dx = c\delta_{m-n}. \tag{2.32}$$

If $c = \int_a^b w(x)\,dx$, the set of orthogonal polynomials is orthonormal.

Any such sequence of orthogonal polynomials $p_0(x), p_1(x), \ldots, p_n(x), \ldots$ can be specified recursively via a three-term recurrence (plus initial conditions) [9]:

$$p_n(x) = (a_n x + b_n)p_{n-1}(x) - c_n p_{n-2}(x). \tag{2.33}$$

Approximating the integral $\int_a^b w(x)f(x)\,dx$ with a summation of the form $\sum_{i=1}^n w_i f(x_i)$ is called "quadrature". Familiar examples of quadrature include Riemann sums, the trapezoidal rule, and Simpson's rule. Choosing the points x_i to be the zeros of the polynomial $p_n(x)$ guarantees that weights can be generated that approximate all polynomials of order $2n-1$ exactly [9]. In addition, with the relatively mild conditions that $f(x)$ be continuously differentiable and the bound on the sequence of deriviatives does not increase super-exponentially, this method produces a uniformly convergent n-term approximation to the integral for any such $f(x)$. The approximation error is

$$\left| \int_a^b f(x)w(x)\,dx - \sum_{i=1}^{n} w_i f(x_i) \right| = \frac{c}{(2n)!} |f^{(2n)}(\xi)|, \tag{2.34}$$

where c is defined in (2.32) and ξ is some fixed point in (a, b).

The difficulty in using quadrature lies in determining the abscissas (x_i's) and associated weights for a given n. In [15], Golub and Welsch developed a simple method for computing the abscissas and weights of an nth-order orthogonal polynomial using the recurrence relation for the sequence of polynomials and eigen-decomposition. This method is explained in [9]. Dividing both sides of (2.33) by a_n,

$$xp_{n-1}(x) = \frac{1}{a_n} p_n(x) - \frac{b_n}{a_n} p_{n-1}(x) + \frac{c_n}{a_n} p_{n-2}(x). \tag{2.35}$$

Consider repeating (2.35) for the entire vector of orthogonal polynomials $\mathbf{p}(x) = [p_0(x), p_1(x), \ldots, p_{N-1}(x)]^T$, and let

$$\mathbf{T} = \begin{bmatrix} -b_1/a_1 & 1/a_1 & 0 & \cdots & 0 \\ c_2/a_2 & -b_2/a_2 & 1/a_2 & & \\ 0 & c_3/a_3 & -b_3/a_3 & & \\ \vdots & & & \ddots & \\ 0 & & & & -b_N/a_N \end{bmatrix}. \tag{2.36}$$

The zeros of $p_N(x)$ are simply the eigenvalues of $\mathbf{T}$, a tridiagonal matrix, since

$$x\mathbf{p}(x) = \mathbf{T}\mathbf{p}(x) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ p_N(x)/a_N \end{bmatrix}. \tag{2.37}$$

Using a similarity transformation, $\mathbf{T}$ can be made symmetric as well; let $\alpha_n = \sqrt{\frac{c_{n+1}}{a_n a_{n+1}}}$

and $\beta_n = -b_n/a_n$. Then, $\mathbf{T}$ is similar to $\mathbf{J}$:

$$\mathbf{J} = \begin{bmatrix} \beta_1 & \alpha_1 & 0 & \cdots & 0 \\ \alpha_2 & \beta_2 & \alpha_2 & & \\ 0 & \alpha_3 & \beta_3 & & \\ \vdots & & & \ddots & \\ 0 & & & & \beta_N \end{bmatrix}. \tag{2.38}$$

The eigenvalues of $\mathbf{J}$ are the zeros of $p_N(x)$. From the eigenvalues x_i and associated eigenvectors $\mathbf{p}(x_i)$, we can find the weights, too. In [15], Golub and Welsch derive the fact that $\sqrt{w_i}$ is the normalizing factor of the eigenvector $\mathbf{p}(x_i)$. Let $\mathbf{q}_i$ be the eigenvector of unit norm for eigenvalue x_i; then, $\mathbf{q}_i = \sqrt{w_i}\mathbf{p}(x_i)$. Using $p_0(x_i)$ since it is a constant,

$$w_i = \frac{q_{i0}^2}{p_0(x_i)^2}. \tag{2.39}$$

2.3.3 Gauss-Hermite Quadrature

One weighting function of interest, defined over $(-\infty, \infty)$, is the normalized Gaussian weighting function $w(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$. The associated sequence of orthogonal polynomials is the class of Hermite polynomials, defined as

$$H_n(x) = n! \sum_{m=0}^{\lfloor n/2 \rfloor} (-1/2)^m \frac{x^{n-2m}}{m!(n-2m)!}, \tag{2.40}$$

or recursively as

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= x \\ H_{n+1}(x) &= xH_n(x) - nH_{n-1}(x). \end{aligned} \tag{2.41}$$

The three-term recurrence in (2.41) can be used to determine the abscissas and weights of the Gauss-Hermite quadrature rule, via the eigenvalue method developed by Golub and Welsch described previously. In the expression for $\mathbf{J}$ (2.38), $\alpha_n = \sqrt{n}$ and $\beta_n = 0$.

Also, these abscissas and weights can be easily adapted for arbitrary μ or σ:

$$\frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} f(x) e^{-\frac{(x-\mu)^2}{2\sigma^2}} \, dx \approx \sum_{i=1}^{n} w_i f(\sigma x_i + \mu). \tag{2.42}$$

Gauss-Hermite quadrature will always converge uniformly to the true value of the integral (as long as $f(x)$ is contained within a superexponential envelope [9]). The rate of convergence for Gauss-Hermite quadrature can be extremely rapid for very smooth functions. More specifically,

$$\text{error} \leq \max_{x} \frac{n!}{(2n)!} \left| f^{(2n)}(x) \right|. \tag{2.43}$$

Clearly, the error is zero for polynomials of degree less than $2n$. Also, the $(2n)!$ term in the denominator will encourage very fast convergence as long as the derivative does not increase exponentially with n.

2.4 Stochastic Approximation Methods for Estimation

Rather than attempt to compute a particular expectation by integration, stochastic approximation samples from the distribution for the random variable(s) over which the expectation is taken to generate an approximation for the expected value.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be N iid samples generated from $p_{\mathbf{X}}(\cdot)$. Define the sample average $\mathbf{S}_N = \frac{1}{N} \sum_{n=1}^{N} \mathbf{f}(\mathbf{x}_n)$. Then, by the Weak Law of Large Numbers, $\mathbf{S}_N \to \mathbb{E}[\mathbf{f}(\mathbf{x})]$ in probability, if that expected value is finite. For large enough N the expected value can be approximated with the sample mean,

$$\mathbb{E}[\mathbf{f}(\mathbf{x})] \approx \frac{1}{N} \sum_{n=1}^{N} \mathbf{f}(\mathbf{x_n}). \tag{2.44}$$

Therefore, the values of the function f evaluated at samples generated from the distribution can be averaged to approximate $\mathbb{E}[f(\mathbf{x})]$. The challenge inherent in this approach is ensuring that enough samples have been taken to approximate the desired expectation. Chebyshev's inequality can be used to show that the rate of convergence is at least σ^2/N (where σ^2 is the trace of the covariance matrix of $\mathbf{x}$).

Unfortunately, stochastic estimation is rarely so simple, because the sampling distribution is either very complex or impossible to sample from directly. Fortunately, a variety of methods have been developed to approximate the expectation in a similar manner without being required to sample from this distribution directly.

2.4.1 Importance Sampling

The simplest such method to approximate $\mathbb{E}[f(\mathbf{x})]$ when $p(\mathbf{x})$ cannot be sampled from directly is importance sampling. Importance sampling consists of sampling instead from a proposal density $q(\mathbf{x})$ that is close to $p(\mathbf{x})$ and using weighted versions of these samples in the summation in (2.44).

Given a proposal density with the same support as $p(\mathbf{x})$ (or at least containing the support of $p(\mathbf{x})$), the modified algorithm is easily derived:

$$\begin{aligned}\mathbb{E}[f(\mathbf{x})] &= \frac{\mathbb{E}_{q(\cdot)}[f(\mathbf{x})w(\mathbf{x})]}{\mathbb{E}_{q(\cdot)}[w(\mathbf{x})]}, \quad w(\mathbf{x}) \triangleq \frac{p(\mathbf{x})}{q(\mathbf{x})} \\ &\approx \frac{\sum_{n=1}^{N} f(\mathbf{x}_n)w(\mathbf{x}_n)}{\sum_{n=1}^{N} w(\mathbf{x}_n)}, \quad \mathbf{x}_n \sim q(\cdot) \ . \end{aligned} \tag{2.45}$$

However, importance sampling has the drawback that the majority of samples $\mathbf{x}_n$ will cluster around the modes of $q(\mathbf{x})$, which may differ from the modes of $p(\mathbf{x})$, thus biasing the approximation away from the true expected value, for finite N. Essentially, importance sampling is relatively inaccurate for a given number of samples, and this method should only be used when the proposal density has a very similar form to the actual distribution.

2.4.2 Rejection Sampling

Rejection sampling is a Monte Carlo method devised by von Neumann [12] for generating samples from a target distribution using an easy-to-sample proposal density.

This method can be combined with normal stochastic approximation to approximate the desired expected value.

Rejection sampling works like this: consider the proposal density $q(\mathbf{x})$. Choose c such that $cq(\mathbf{x}) \geq p(\mathbf{x})$, for all values of $\mathbf{x}$; this is the most challenging step of the algorithm, since it requires that we can find an envelope for the function $p(\mathbf{x})$. Also, if c is chosen to be too large, the algorithm will end up rejecting a larger number of samples. Sample $(u, \mathbf{x})$, where $u \sim U(0,1)$, and $\mathbf{x} \sim q(\cdot)$. If $ucq(\mathbf{x}) \leq p(\mathbf{x})$, accept the sample; otherwise, reject it, and repeat this step. The accepted sample has the property that it is distributed according to $p(\mathbf{x})$.

The rationalization of rejection sampling is quite straightforward. Given a sample $\mathbf{x} \sim q(\mathbf{x})$, the joint distribution of $(u, \mathbf{x})$ is $q(\mathbf{x})U(0,1)$. Then, the probability of acceptance for a given value of $\mathbf{x}$ is $P(u < p(\mathbf{x})/cq(\mathbf{x}) \mid \mathbf{x}) = p(\mathbf{x})/cq(\mathbf{x})$, and the joint probability $p(\mathbf{x}, \text{accepted}) = p(\mathbf{x})/c$. Using Bayes' Rule, it is easy to show that the probability density of $\mathbf{x}$, given that it was accepted, is simply the desired pdf $p(\mathbf{x})$:

$$p(\mathbf{x}|\text{accepted}) = \frac{p(\mathbf{x}, \text{accepted})}{\int p(\mathbf{x}, \text{accepted})\, d\mathbf{x}'} = \frac{p(\mathbf{x})/c}{\int p(\mathbf{x}')/c\, d\mathbf{x}'} = p(\mathbf{x}). \tag{2.46}$$

2.4.3 Gibbs Sampling

Gibbs sampling, discussed by Geman and Geman in [14], is a Markov chain Monte Carlo method that employs a Markov chain whose stationary distribution is the joint posterior distribution of interest. For convenience, define the notation $\mathbf{x}_{-k}$ be the random vector $[x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K]$. The fundamental assumption of Gibbs sampling is that while the joint distribution $p(x_1, \ldots, x_K)$ is difficult to sample from, it is easy to generate samples from the full conditional distributions $p(x_k|\mathbf{x}_{-k})$. Denote the random variable with this full conditional distribution $x_{k|-k}$.

Thus, by constructing a Markov graph like Figure 2-1 to have the conditioned parameters $x_{k|-k} \sim p(x_k|\mathbf{x}_{-k})$ as the nodes of the graph, the stationary distribution of this graph is the joint distribution $p(x_1, \ldots, x_K)$. A simple random walk can be used to approach this joint distribution, and after a "burn-in" period of iterations until the walk reaches a steady state, further samples can be used as if they were

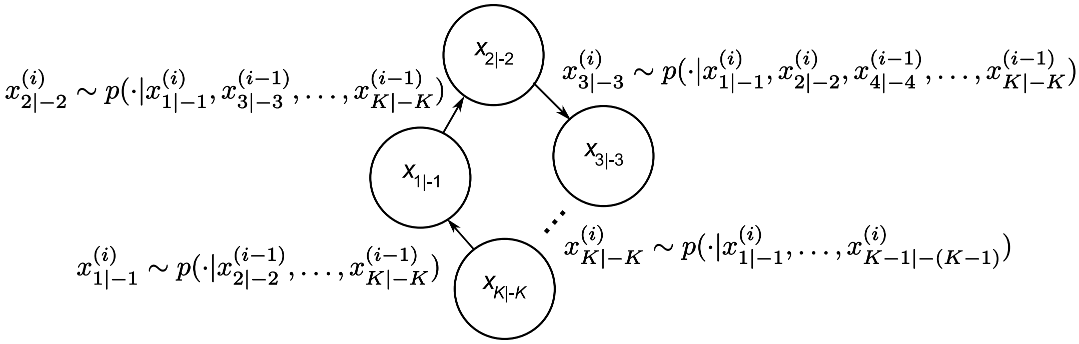

$x_{2|-2}$
$x_{3|-3}$
$x_{1|-1}$
$x_{K|-K}$
$x_{2|-2}^{(i)} \sim p(\cdot|x_{1|-1}^{(i)}, x_{3|-3}^{(i-1)}, \ldots, x_{K|-K}^{(i-1)})$
$x_{3|-3}^{(i)} \sim p(\cdot|x_{1|-1}^{(i)}, x_{2|-2}^{(i)}, x_{4|-4}^{(i-1)}, \ldots, x_{K|-K}^{(i-1)})$
$x_{1|-1}^{(i)} \sim p(\cdot|x_{2|-2}^{(i-1)}, \ldots, x_{K|-K}^{(i-1)})$
$x_{K|-K}^{(i)} \sim p(\cdot|x_{1|-1}^{(i)}, \ldots, x_{K-1|-(K-1)}^{(i)})$

Figure 2-1: Markov chain of the Gibbs sampler. Gibbs sampling consists of repeatedly sampling from these transition distributions in turns until the chain converges to the steady-state. Since the steady-state distribution is the joint pdf $p(\mathbf{x})$, further samples from the transition probabilities will be generated as if they were generated by the joint distribution itself.

generated from the joint distribution directly.

The main drawback to Gibbs sampling is that the burn-in period can be highly variable, and extensive testing is necessary to determine how many steps to wait until the process has converged.

2.4.4 Slice Sampling

A more general Monte Carlo method that frames the sampling problem differently is termed slice sampling; Neal explains this method in detail in [25]. For simplicity, only univariate slice sampling will be discussed here.

Slice sampling derives from the principle that sampling from a distribution $x \sim p(\cdot)$ is the same as sampling uniformly from the region under the density function, $\{(x, y) : 0 < y < p(x)\}$, and discarding the y value. However, for a given distribution, this region is not easily defined.

Slice sampling uses a Markov chain (depicted in Figure 2-2) that converges to this distribution to overcome such difficulties, analogously to the idea of Gibbs sampling where $p(x|y)$ and $p(y|x)$ are the conditional sampling distributions. Essentially, each iteration of slice sampling consists of three steps, with the latter two steps approximately sampling x from the slice of the region with the appropriate y value:

1. Given $x^{(0)}$, compute $p(x^{(0)})$ and sample y uniformly from $(0, p(x^{(0)}))$.

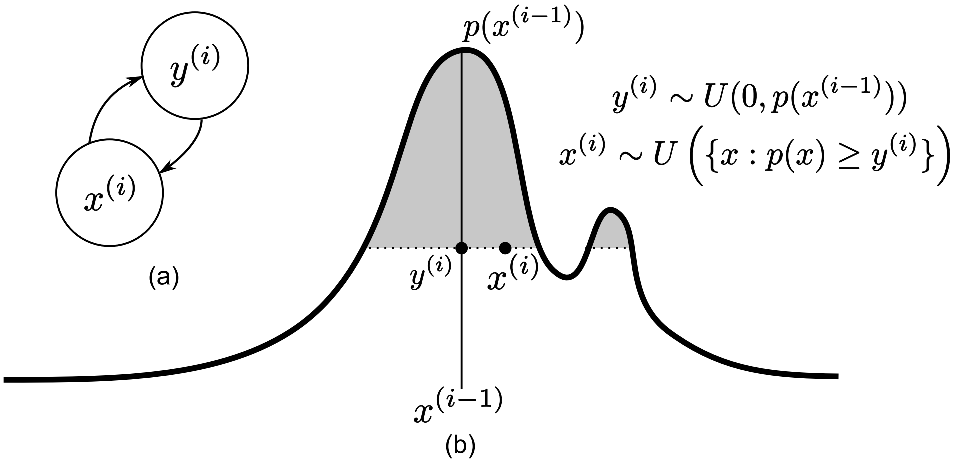

$y^{(i)}$
$x^{(i)}$
(a)
$p(x^{(i-1)})$
$y^{(i)} \sim U(0, p(x^{(i-1)}))$
$x^{(i)} \sim U\left(\{x : p(x) \geq y^{(i)}\}\right)$
$y^{(i)}$
$x^{(i)}$
$x^{(i-1)}$
(b)

Figure 2-2: Slice sampling of $p(x)$ illustrated: (a) Sampling is performed by traversing a Markov chain to approximate $p(x)$, the stationary distribution. Each iteration consists of (b) uniformly choosing a slice $\{x : p(x) = y\}$ and uniformly picking a new sample x from that slice.

2. Construct an interval (L, R) containing $x^{(0)}$ and as much of the slice $S = \{x : f(x) > y\}$ as possible.

3. Uniformly sample $x^{(1)}$ from the portion of slice contained in that interval.

These last two steps in particular are not easily implemented efficiently. Making the interval as small as possible while containing as much of the slice as possible is a difficult task considering that the inverse image $p^{-1}(y)$ is almost always impossible to find, and usually, the implementer does not even know how many times $p(x)$ crosses y. The third step is difficult because even when an interval is small, the fraction of the interval contained in the slice may be a much smaller still, so a sample from the interval may only be in the slice with a low probability.

The ideal choice of an interval containing the slice S would be $(L = \inf(S), R = \sup(S))$; this interval contains the entire slice, and the range of points in the interval not in the slice is minimized. To choose such an interval, a root-finding method can be employed to locate the outermost points where $p(x)$ crosses the generated value y. Newton's method or Halley's method can be used, or if the distribution is approximately monotonic over the interval of interest, a binary-search method may yield exponentially fast convergence. The binary search method for where a monotonically increasing function $p(x)$ takes the value y over the interval $[L, R]$ is described below:

```
Require: $p(x), y, L, R$
  while $R - L > \epsilon$ do
    $x = (L + R)/2$
    if $p(x) > y$ then
      $R = x$
    else
      $L = x$
    end if
  end while
  return $x$
```

The monotonically decreasing version of this algorithm is similar (just switch the inequality in the "if" condition).

However, the probability density function of interest is rarely monotonic. If the support of $p(x)$ is bounded (finite), the entire support can be used; however, the slice still may be a very small portion of the interval, and sampling from the slice may prove very inefficient. In [25], Neal develops a "stepping-out" method for expanding upon an initial estimate for the size of the slice S, as well as a "doubling" procedure for expanding upon a random initial estimate for the size of the slice.

To sample from the slice contained in the chosen interval, the "shrinkage" procedure proposed in [25] is used. Essentially, each rejected sample drawn from the interval is used to shrink the interval; since the interval always had and will continue to contain $x^{(i-1)}$ (part of the slice by construction), as the interval size goes to zero, the probability that the sample from the interval will not be in the slice also goes to zero, assuming that the slice contains an open set around $x^{(i-1)}$. Assuming that $p(x)$ is continuous around $x^{(i-1)}$ is sufficient. The shrinkage method to sample from the slice $\{x : p(x) \geq y\}$ (or portion thereof) contained in the interval $[L, R]$ is described below:

```
Require: $p(x), x^{(i-1)}, y, L, R$
```

```
$x^{(i)} = x^{(i-1)}$
while $R - L > \epsilon$ do
    $x \sim U(L, R)$
    if $p(x) > y$ then
        $x^{(i)} = x$
        $L = R$
    else if $x < x^{(i-1)}$ then
        $L = x$
    else
        $R = x$
    end if
end while
return $x^{(i)}$
```

This algorithm converges on a random sample from the slice contained in the interval.

Chapter 3

Non-Random Parameter Estimation

This chapter focuses on developing and comparing techniques for estimating unknown, deterministic parameters of a bandlimited signal in the presence of jitter and additive noise. When there is no jitter, the efficient estimator (2.8) is linear and has a closed form. The purpose of developing estimators for the true observation model is to do better than the efficient (no-jitter) estimator when the sampling process is affected by jitter.

The general observation model is discussed in the introduction (see 1.9): N observations $\mathbf{y}$ are generated by sampling a periodic (or periodically-extended) signal, parameterized by $\mathbf{x}$, at M times the Nyquist rate. The resulting likelihood function (2.3) does not have a closed form, and the N-dimensional integral requires careful application of numerical integration techniques to ensure computational feasability. The independence assumptions on the timing noise and additive noise sources together make the N-dimensional integral separable, yielding a product of N one-dimensional integrals, which is a much easier expression to manipulate:

$$p_{\mathbf{Y}}(\mathbf{y};\mathbf{x}) = \prod_{n=0}^{N-1} \int \mathcal{N}(y_n; \mathbf{h}_n(z_n)^T\mathbf{x}, \sigma_w^2)\mathcal{N}(z_n; 0, \sigma_z^2)\, dz_n, \tag{3.1}$$

where $\mathbf{h}_n(z_n)^T$ is the nth row vector of the matrix $\mathbf{H}(\mathbf{z})$.

An unbiased linear estimator and several iterative maximum-likelihood estimators are developed for this observation model and compared against the efficient estimator for the no-jitter model when the samples are in fact jittered. The Cramér-Rao bound is approximated using stochastic estimation and Hermite quadrature, and the resulting approximate minimum MSE values are compared against the linear and iterative estimators. For substantial amounts of jitter, the results at the end of this chapter demonstrate that the iterative EM algorithm developed here provides better performance than the no-jitter efficient estimator when MSE performance is paramount.

3.1 Linear Estimators

A variety of linear estimators $\hat{\mathbf{x}}_L(\mathbf{y}) \triangleq \mathbf{A}\mathbf{y} + \mathbf{b}$ can be used to estimate the signal parameters. First, the ML estimator (2.13) derived for the known jitter case is linear; this estimator is efficient in the absence of jitter (or, alternatively, when the jitter is known), and therefore, this estimator is used as the baseline approach against which to compare new estimators.

Another type of linear estimators of interest is the class of linear unbiased estimators for the true observation model. Consider the unbiased linear estimator (2.10); this estimator corresponds to $\mathbf{A} = \mathbb{E}[\mathbf{H}(\mathbf{z})]^+$ and $\mathbf{b} = \mathbf{0}$.

Since an unbiased linear estimator must be unbiased for any value of $\mathbf{x}$, it is easy to see that when $\mathbf{x} = \mathbf{0}$, $\mathbf{b}$ also must be zero. In addition, for the estimator to be unbiased in all other cases, $\mathbf{A}\mathbb{E}[\mathbf{H}(\mathbf{z})] = \mathbf{I}$. Thus, the set of linear unbiased estimators is

$$\hat{\mathbf{x}}_{L,\mathbb{E}[\hat{\mathbf{x}}]=\mathbf{x}}(\mathbf{y}) \in \{\mathbf{A}\mathbf{y} : \mathbf{A}\mathbb{E}[\mathbf{H}(\mathbf{z})] = \mathbf{I}\}. \tag{3.2}$$

Since $\mathbb{E}[\mathbf{H}(\mathbf{z})]$ is of full column rank, the pseudoinverse is a left inverse of $\mathbb{E}[\mathbf{H}(\mathbf{z})]$, thus satisfying the condition for $\mathbf{A}$ in (3.2):

$$\mathbb{E}[\mathbf{H}(\mathbf{z})]^+ = (\mathbb{E}[\mathbf{H}(\mathbf{z})]^T\mathbb{E}[\mathbf{H}(\mathbf{z})])^{-1}\mathbb{E}[\mathbf{H}(\mathbf{z})]^T. \tag{3.3}$$

The Best Linear Unbiased Estimator (BLUE) for the MSE criterion can be derived similarly to the fixed-$\mathbf{H}$ matrix case specified by Kay in [18]. Adapting this formula to a stochastic $\mathbf{H}$ matrix (see Appendix A), however, results in

$$\hat{\mathbf{x}}_{BLUE}(\mathbf{y}) = (\mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbf{\Lambda_y}^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})])^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbf{\Lambda_y}^{-1} \mathbf{y}, \tag{3.4}$$

where the covariance matrix of the data $\mathbf{\Lambda_y}$ depends on the value of the parameters:

$$\mathbf{\Lambda_y} = \mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{x}\mathbf{x}^T\mathbf{H}(\mathbf{z})^T] - \mathbb{E}[\mathbf{H}(\mathbf{z})]\mathbf{x}\mathbf{x}^T\mathbb{E}[\mathbf{H}(\mathbf{z})]^T + \sigma_w^2\mathbf{I}. \tag{3.5}$$

This estimator depends on the actual value of $\mathbf{x}$. Thus, no unique valid linear unbiased estimator minimizes the MSE for all values of $\mathbf{x}$. To determine more carefully what goes wrong, re-examine the expression for $\mathbf{\Lambda_y}$. Because the jitter is independent, the first expectation in (3.6) is separable for $m \neq n$, and the off-diagonal elements of $\mathbf{\Lambda_y}$ are identically zero:

$$\begin{aligned}
[\mathbf{\Lambda_y}]_{m,n} &= \mathbb{E}[\mathbf{h}_m(z_m)^T\mathbf{x}\mathbf{x}^T\mathbf{h}_n(z_n)] - \mathbb{E}[\mathbf{h}_m(z_m)^T]\mathbf{x}\mathbf{x}^T\mathbb{E}[\mathbf{h}_n(z_n)] && (3.6)\\
&= \mathbb{E}[\mathbf{h}_m(z_m)^T\mathbf{x}]\mathbb{E}[\mathbf{x}^T\mathbf{h}_n(z_n)] - \mathbb{E}[\mathbf{h}_m(z_m)^T]\mathbf{x}\mathbf{x}^T\mathbb{E}[\mathbf{h}_n(z_n)] && (3.7)\\
&= 0. && (3.8)
\end{aligned}$$

The diagonal elements of $\mathbf{\Lambda_y}$ are simply $\mathrm{var}(\mathbf{h}_n(z_n)^T\mathbf{x}) + \sigma_w^2$. Using the fact that $\mathbf{h}_n(z_n)^T\mathbf{x}$ is just a scalar,

$$\mathbf{\Lambda_y} = \begin{bmatrix} \mathbf{x}^T\,\mathrm{cov}(\mathbf{h}_0(z_0))\mathbf{x} & & \\ & \ddots & \\ & & \mathbf{x}^T\,\mathrm{cov}(\mathbf{h}_{N-1}(z_{N-1}))\mathbf{x} \end{bmatrix} + \sigma_w^2\mathbf{I}. \tag{3.9}$$

To result in a valid estimator, it is sufficient for the covariance matrix $\mathbf{\Lambda_y}$ to be a scalar matrix (a scalar times the identity matrix), which can only happen when the inner covariance matrices $\mathrm{cov}(\mathbf{h}_n(z_n))$ are equal for all n. If $\mathbf{\Lambda_y}$ were not a scalar matrix, it could not commute with $\mathbb{E}[\mathbf{H}(\mathbf{z})]$ in (3.4), and the resulting estimator would still depend on $\mathbf{x}$. Since (3.9) is only a scalar matrix when $\mathbf{h}_n(z_n)$ is deterministic, the

BLUE expression would be expected to vary with $\mathbf{x}$. Experimentation with $K = 3$, $M = 2$, and $\sigma_w = \sigma_z = 0.1$ verifies that the BLUE estimator is in fact not valid, since the resulting matrix $\mathbf{A}$ in the linear model differs for different values of $\mathbf{x}$:

Table 3.1: BLUE ($\hat{\mathbf{x}}_{BLUE}(\mathbf{y}) = \mathbf{A}\mathbf{y}$) for randomly generated values of $\mathbf{x}$

$\mathbf{x}$	$\mathbf{A} = (\mathbb{E}[\mathbf{H}(\mathbf{z})]^T {\mathbf{\Lambda}_\mathbf{y}}^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})])^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})]^T {\mathbf{\Lambda}_\mathbf{y}}^{-1}$
$\begin{bmatrix} 0.7374 \\ -0.8311 \\ -0.2004 \end{bmatrix}$	$\begin{bmatrix} 0.6527 & 0.1871 & 0.072 & -0.1709 & -0.0114 & 0.2705 \\ 0.1013 & 0.2275 & 0.5427 & 0.3806 & -0.0518 & -0.2002 \\ -0.0313 & -0.1511 & -0.1009 & 0.5131 & 0.3268 & 0.4434 \end{bmatrix}$
$\begin{bmatrix} -0.4803 \\ 0.6001 \\ -0.1372 \end{bmatrix}$	$\begin{bmatrix} 0.507 & 0.269 & 0.0729 & -0.1908 & -0.0553 & 0.3972 \\ 0.049 & 0.216 & 0.637 & 0.2672 & -0.0022 & -0.167 \\ -0.0818 & -0.1644 & -0.0048 & 0.398 & 0.3781 & 0.4749 \end{bmatrix}$
$\begin{bmatrix} 0.8213 \\ -0.6363 \\ -0.4724 \end{bmatrix}$	$\begin{bmatrix} 0.7543 & 0.1541 & 0.0355 & -0.1334 & 0.0196 & 0.17 \\ 0.1072 & 0.2901 & 0.4105 & 0.5137 & -0.1165 & -0.2051 \\ 0.0699 & -0.1837 & -0.1378 & 0.551 & 0.3574 & 0.3433 \end{bmatrix}$

3.2 Iterative ML Estimation

The ML estimation problem requires solving the optimization problem in (2.11), which is non-linear in $\mathbf{x}$. Thus, approximate methods involving repeated iterations of a simpler problem are of interest.

One such iterative method of iterative MAP/ML estimation was previously investigated by Kusuma and Goyal in [19, 20] for the similar problem of delay estimation. Using a one-at-a-time maximization approach, the estimates of $\mathbf{x}, \mathbf{z}$ converge to a local maximum of the joint pdf $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$. The iterations are divided into two steps:

1. Given a prior estimate of the parameters (e.g. delay), compute the MAP estimate for the jitter:
$$\hat{\mathbf{z}}^{(i+1)} = \arg\max_{\mathbf{z}} p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}|\mathbf{y}; \hat{\mathbf{x}}^{(i)}). \tag{3.10}$$

2. Use the new estimate of the jitter to find the most likely estimate of the deterministic parameters:
$$\hat{\mathbf{x}}^{(i+1)} = \arg\max_{\mathbf{x}} p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|\hat{\mathbf{z}}^{(i+1)}; \mathbf{x}). \tag{3.11}$$

The first step can be simplified by recognizing that the posterior density for the jitter is separable, so the individual jitters can be estimated individually. Although the resulting expression to maximize is still non-convex, it is one-dimensional, and standard gradient techniques can be used to find locally optimal points:

$$\hat{z}_n^{(i+1)} = \arg\min_{z_n} \frac{z_n^2}{\sigma_z^2} + \frac{(y_n - \mathbf{h}_n(z_n)^T \hat{\mathbf{x}}^{(i)})^2}{\sigma_w^2}. \tag{3.12}$$

The second step is even simpler; given an estimate for $\mathbf{z}$, the resulting optimization problem is an over-determined least-squares optimization problem, with a standard solution via projection/orthogonalization.

$$\hat{\mathbf{x}}^{(i+1)} = \arg\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{H}(\hat{\mathbf{z}}^{(i+1)})\mathbf{x}||^2 = \mathbf{H}(\hat{\mathbf{z}}^{(i+1)})^+\mathbf{y}. \tag{3.13}$$

A modification of this algorithm is to use the Bayes' Least Squares estimate of the jitter in the first step, instead of the MAP estimate. The BLS estimate can be determined using Gauss-Hermite quadrature to numerically integrate $\mathbb{E}[z_n|y_n;\mathbf{x}]$, for each n separately. The use of quadrature is much faster and more numerically stable than the general local optimization routines that can be employed to solve (3.12). However, this change can disrupt the monotone convergence behavior of the MAP/ML algorithm. In fact, it is easy to construct a situation where such an algorithm never convergences or even diverges:

For simplicity, consider the trivial case where x and z are one-dimensional. Then, suppose we start at $x^{(1)}, z^{(1)}$ in Figure 3-1. The MAP step would remain at $z^{(1)}$, since it is a local maximum, but the BLS step would instead move to $z^{(2)}$, being the mean of the conditional density $p(z|y; x = x^{(1)})$. Then, the ML step would choose $x^{(2)}$, being the maximum of the conditional density $p(y|z = z^{(2)}; x)$. The BLS step would then move back to $z^{(1)}$, and the ML step would move back to $x^{(1)}$, so it is not difficult to imagine cycling between these locations forever.

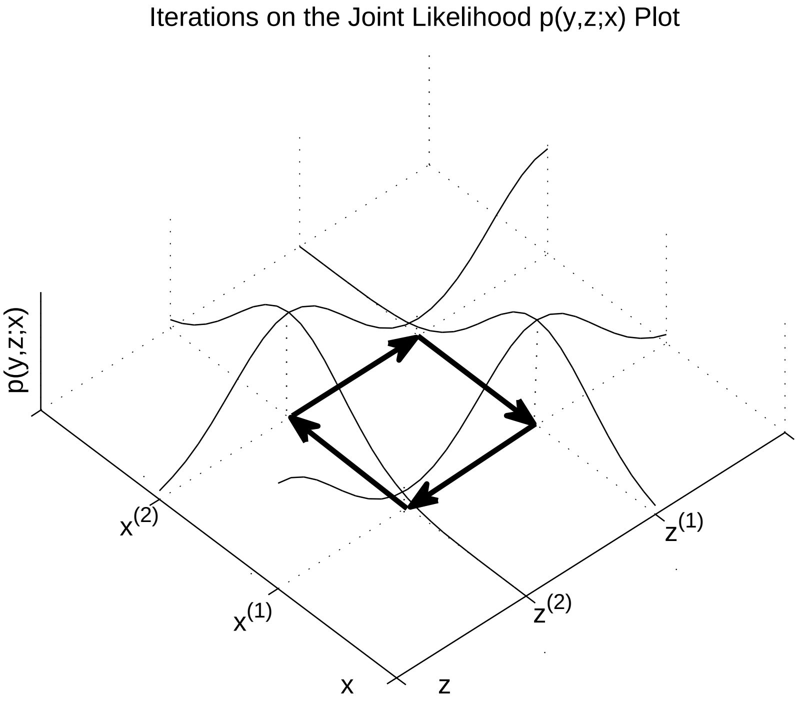


Figure 3-1: Iterative BLS/ML Algorithm: Non-convergent behavior example where the estimates of z and x alternate between $z^{(1)}$ and $z^{(2)}$ and $x^{(1)}$ and $x^{(2)}$, never converging to a single value.

3.3 ML Estimation Using the EM Algorithm

The philosophy behind the Expectation-Maximization algorithm is that the source of the difficulty in ML estimation is the uncertainty present in the observation model. The EM algorithm framework attempts to circumvent this problem by augmenting the observations with intermediate, (usually) hidden random variables that account for the sources of the uncertainty in such a way as to make the new observation model more manageable. For instance, in [3], Bilmes uses the EM algorithm to drastically reduce the difficulty of determining the parameters of a mixture density from observations by augmenting those observations with the hidden knowledge of what distribution in the mixture generated each observation. In another example, Bilmes describes how to use the Baum-Welch [2] algorithm, which is a well-known member of the class of Generalized EM algorithms, to estimate the parameters of a hidden Markov model.

Likewise, in this estimation problem, the observed data $\mathbf{y}$ is incomplete; however, with knowledge of the jitter values $\mathbf{z}$, the estimation problem becomes manageable. Defining the hidden data to be the jitter $\mathbf{z}$, and the complete data to be the combination of the two, the EM algorithm can be used to maximize the likelihood function by repeatedly maximizing (2.14), the expected log-likelihood of the complete data, conditioned on the observed data and the previous estimate of the parameters.

Each iteration of the algorithm consists of maximizing (3.14) conditioned on the prior estimate of $\mathbf{x}$:

$$Q(\mathbf{x}, \hat{\mathbf{x}}^{(i)}) = \mathbb{E}\left[\log p(\mathbf{y}, \mathbf{z}; \mathbf{x}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]. \tag{3.14}$$

This expectation is actually rather straightforward to maximize. The log-likelihood of the complete data can be expressed as the sum of the log-likelihood of the observations given the jitter and the log-prior on the jitter:

$$\log p(\mathbf{y}, \mathbf{z}; \mathbf{x}) = -\frac{1}{2\sigma_w^2}||\mathbf{y} - \mathbf{H}(\mathbf{z})\mathbf{x}||_2^2 - \frac{1}{2\sigma_z^2}||\mathbf{z}||_2^2 + \text{constant}. \tag{3.15}$$

Expanding the above further to separate the powers of $\mathbf{x}$ and substituting this expanded form into (3.14) yields

$$\begin{aligned} Q(\mathbf{x}, \hat{\mathbf{x}}^{(i)}) = \frac{-1}{2\sigma_w^2}&\left(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbb{E}\left[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]\mathbf{x} + \mathbf{x}^T\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]\mathbf{x}\right) \\ &- \frac{1}{2\sigma_z^2}\mathbb{E}\left[\mathbf{z}^T\mathbf{z}|\mathbf{y}; \hat{\mathbf{x}}^{(i)}\right] + \text{constant}. \end{aligned} \tag{3.16}$$

Maximizing (3.16) with respect to $\mathbf{x}$ involves taking the derivative and setting it equal to zero:

$$\frac{\partial Q(\mathbf{x}, \hat{\mathbf{x}}^{(i)})}{\partial \mathbf{x}} = \frac{1}{\sigma_w^2}\left(\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T|\mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]\mathbf{y} - \mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]\mathbf{x}\right) = \mathbf{0}. \tag{3.17}$$

The Hessian matrix is negative definite since $\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})$ is positive definite; therefore, the extremal value is indeed a maximum:

$$\frac{\partial^2 Q(\mathbf{x}, \hat{\mathbf{x}}^{(i)})}{\partial \mathbf{x} \partial \mathbf{x}^T} = -\frac{1}{\sigma_w^2} \mathbb{E}\left[\mathbf{H}(\mathbf{z})^T \mathbf{H}(\mathbf{z}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right] < \mathbf{0}. \tag{3.18}$$

At the maximum value $\mathbf{x} = \hat{\mathbf{x}}^{(i+1)}$, (3.17) must vanish; thus, rearranging terms,

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T \mathbf{H}(\mathbf{z}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right] \mathbf{x} = \mathbb{E}\left[\mathbf{H}(\mathbf{z}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]^T \mathbf{y}. \tag{3.19}$$

Since $\mathbf{H}(\mathbf{z})^T \mathbf{H}(\mathbf{z})$ is positive definite with probability 1, the solution to (3.19) exists and is unique. Thus, all that remains is to find the values of these expectations. Using the conditional independence of $\mathbf{y}$ given $\mathbf{z}$ and Bayes' Rule (this is where the assumption of whiteness of both $\mathbf{w}$ and $\mathbf{z}$ comes into play),

$$p(\mathbf{z}|\mathbf{y}; \hat{\mathbf{x}}^{(i)}) = \prod_{n=0}^{N-1} \frac{p(y_n|z_n; \hat{\mathbf{x}}^{(i)}) p(z_n)}{p(y_n; \hat{\mathbf{x}}^{(i)})}. \tag{3.20}$$

What makes Gauss-Hermite quadrature an attractive method for numeric integration in both cases is that the integrals are now separable. Below are the approximations used for the left and right sides of (3.19).

$$\begin{aligned}
\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T \mathbf{H}(\mathbf{z}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right] &= \sum_{n=0}^{N-1} \mathbb{E}\left[\mathbf{h}_n(z_n) \mathbf{h}_n(z_n)^T | y_n; \hat{\mathbf{x}}^{(i)}\right] && (3.21)\\
&\approx \sum_{n=0}^{N-1} \frac{1}{p(y_n; \hat{\mathbf{x}}^{(i)})} \sum_{i=1}^{I} w_i \mathbf{h}_n(z_i) \mathbf{h}_n(z_i)^T p(y_n|z_i; \hat{\mathbf{x}}^{(i)}) && (3.22)\\
\left[\mathbb{E}\left[\mathbf{H}(\mathbf{z}) | \mathbf{y}; \hat{\mathbf{x}}^{(i)}\right]\right]_{n,:} &= \mathbb{E}\left[\mathbf{h}_n(z_n)^T | y_n; \hat{\mathbf{x}}^{(i)}\right] && (3.23)\\
&\approx \frac{1}{p(y_n; \hat{\mathbf{x}}^{(i)})} \sum_{i=1}^{I} w_i \mathbf{h}_n(z_i)^T p(y_n|z_i; \hat{\mathbf{x}}^{(i)}) && (3.24)\\
p(y_n; \hat{\mathbf{x}}^{(i)}) &\approx \sum_{i=1}^{I} w_i p(y_n|z_i; \hat{\mathbf{x}}^{(i)}) && (3.25)
\end{aligned}$$

Using Gauss-Hermite quadrature, the complexity is roughly $O(NI)$, which means that the number of computations scales linearly with the number of samples or number of terms in the quadrature. Such computational performance is generally considered reasonable, although for an increasing number of samples, calculations would be better done off-line.

3.4 Approximating the Cramér-Rao Bound

By definition, the Cramér-Rao lower bound on the MSE of an unbiased estimator of the parameters is the trace of the inverse of the Fisher information matrix $\mathbf{I}_{\mathbf{y}}(\mathbf{x})$, defined in (2.6). The Fisher information matrix can be estimated using Gauss-Hermite quadrature and stochastic approximation. Using the independence of the noise, the log-likelihood function factors into a summation of log-likelihoods of the individual data:

$$\ln p(\mathbf{y};\mathbf{x}) = \sum_{n=0}^{N-1} \ln p(y_n;\mathbf{x}). \tag{3.26}$$

Substituting into (2.6),

$$\mathbf{I}_{\mathbf{y}}(\mathbf{x}) = \sum_{n=0}^{N-1} \mathbb{E}\left[\left(\frac{\partial \ln p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)\left(\frac{\partial \ln p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)^T\right]. \tag{3.27}$$

Differentiating (3.26) yields

$$\frac{\partial \ln p(y_n;\mathbf{x})}{\partial \mathbf{x}} = \frac{\frac{\partial p(y_n;\mathbf{x})}{\partial \mathbf{x}}}{p(y_n;\mathbf{x})}. \tag{3.28}$$

The bottom expression is approximated in (3.25). Similarly, the top expression can be approximated using Gauss-Hermite quadrature, resulting in

$$\frac{\partial p(y_n;\mathbf{x})}{\partial \mathbf{x}} \approx \sum_{i=1}^{I} w_i \mathcal{N}(y_n;\mathbf{h}_n(z_i)^T\mathbf{x},\sigma_w^2)\frac{(y_n - \mathbf{h}_n(z_i)^T\mathbf{x})\mathbf{h}_n(z_i)}{\sigma_w^2}. \tag{3.29}$$

Let $\mathbf{F}_n(y_n;\mathbf{x})$ be the resulting approximation for $\left(\frac{\partial \ln p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)\left(\frac{\partial \ln p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)^T$. Since sampling from a Gaussian mixture can be accomplished simply by choosing a mixture randomly according to the weights, and sampling from the chosen Normal distribution, the approximation to $p(y_n;\mathbf{x})$, a Gaussian mixture, is easy to sample. These sampled values can be substituted into the expression inside the expectation in (3.27) to approximate the Fisher information matrix. Taking the trace of the inverse of the result yields an estimate of the Cramér-Rao lower bound. In summary,

$$CRB = \operatorname{tr}\left(\mathbf{I}_{\mathbf{y}}(\mathbf{x})^{-1}\right) \approx \operatorname{tr}\left(\left[\sum_{n=0}^{N-1} \frac{1}{S} \sum_{s=1}^{S} \mathbf{F}_n(y_s; \mathbf{x})\right]^{-1}\right), \tag{3.30}$$

where $y_s \sim \sum_{i=1}^{I} w_i \mathcal{N}(y_n; \mathbf{h}_n(z_i)^T \mathbf{x}, \sigma_w^2)$ and

$$\begin{aligned}\mathbf{F}_n(y_n; \mathbf{x}) = &\left(\frac{\sum_{i=1}^{I} w_i \mathcal{N}(y_n; \mathbf{h}_n(z_i)^T \mathbf{x}, \sigma_w^2)(y_n - \mathbf{h}_n(z_i)^T \mathbf{x})\mathbf{h}_n(z_i)}{\sigma_w^2 \sum_{i=1}^{I} w_i p(y_n|z_i; \mathbf{x})}\right) \\ &\cdot \left(\frac{\sum_{i=1}^{I} w_i \mathcal{N}(y_n; \mathbf{h}_n(z_i)^T \mathbf{x}, \sigma_w^2)(y_n - \mathbf{h}_n(z_i)^T \mathbf{x})\mathbf{h}_n(z_i)}{\sigma_w^2 \sum_{i=1}^{I} w_i p(y_n|z_i; \mathbf{x})}\right)^T.\end{aligned} \tag{3.31}$$

3.5 Simulations

The ML estimation algorithms presented (the alternating MAP/ML estimator, the alternating BLS/ML variant, and the EM algorithm) are all implemented using MATLAB. To evaluate these algorithms, both the (no-jitter) efficient linear estimator in (2.13) and the linear unbiased estimator in (2.10) are also developed. In addition, the Gauss-Hermite quadrature approximation to the Cramér-Rao lower bound given in (3.30) is written.

Before comparing the performance of these iterative algorithms, their convergence properties and sensitivity to initial conditions needs to be studied. While the alternating MAP/ML and EM algorithms are guaranteed to converge, the algorithms do not guarantee any particular rate of convergence, and the BLS/ML algorithm is not guaranteed to converge at all (although in many cases, it will). Since these iterative algorithms all depend on the supplied initial conditions, gauging the effect on these algorithms of choosing different initial conditions is important.

Once the appropriate number of iterations and choice of initial conditions have been determined for these algorithms, simulations are performed to compare the MSE of the best of these iterative algorithms against the MSE of the best estimator for the no-jitter case. In particular, these simulations depict how much more jitter the iterative algorithms would tolerate than the no-jitter linear estimator, for the same MSE. The Cramér-Rao lower bound approximation is used as a point of comparison for the relative efficiency of these estimators.

3.5.1 Convergence and initial condition sensitivity of iterative methods

If these iterative algorithms are to prove functional, a certain degree of convergence must be guaranteed for a given number of iterations. Before presenting the simulation results, some discussion of the expected convergence properties of these algorithms should be reviewed.

The iterative MAP/ML algorithm is a simple "hill-climbing" algorithm, repeatedly maximizing the joint distribution $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$ with respect to $\mathbf{z}$ and $\mathbf{x}$, along different directions. The invariant property of MAP/ML estimation is that the joint probabilty density of the generated values is guaranteed to be increasing over every iteration. Since the distribution function is in fact bounded, the algorithm is guaranteed to converge to a local maximum. The EM algorithm, which also increases the likelihood function with each iteration, is guaranteed also to converge to a local maximum of the likelihood function $p(\mathbf{y}; \mathbf{x})$.

The iterative BLS/ML algorithm is not such a hill-climbing algorithm, so there is no such invariant that the joint probability $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$ increases with each iteration of guessing $\mathbf{z}$ and $\mathbf{x}$. In particular, the BLS step may result in a decreased joint pdf. Thus, this algorithm is not guaranteed to converge. However, empirical simulations presented below suggest that this algorithm is well-behaved as long as the jitter is sufficiently small relative to the additive noise.

These three algorithms are run for 500 iterations each, for a variety of different choices of M, σ_z, and σ_w. The squared Euclidean norm of the difference between successive guesses for $\mathbf{x}$ is plotted for the different algorithms for several of these cases. At convergence, this difference goes to zero. The convergence behavior is shown in Figure 3-2 for several different values of M, σ_z, and σ_w.

This series of plots demonstrates the general convergence trends observed through extensive simulation. Decreasing σ_z and σ_w respectively decrease and increase the

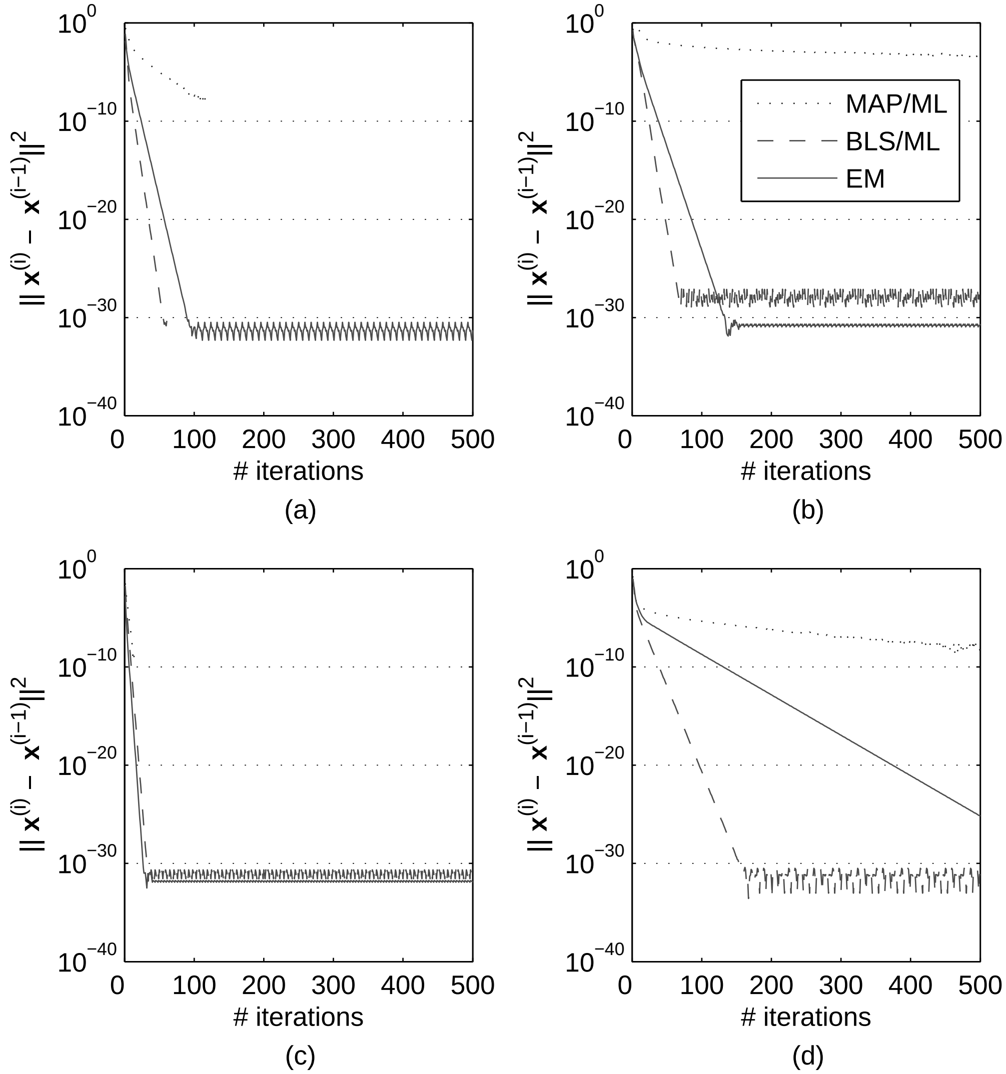


Figure 3-2: Convergence of Iterative (MAP/ML, BLS/ML, EM) Algorithms: The 2-norm of the difference between successive iterations of these algorithms is plotted over 500 iterations for $K = 5$ and different parameter values: (a) $M = 4$, $\sigma_z = 0.5$, $\sigma_w = 0.75$ (b) $M = 16$, $\sigma_z = 0.5$, $\sigma_w = 0.75$ (c) $M = 4$, $\sigma_z = 0.25$, $\sigma_w = 0.75$ (d) $M = 4$, $\sigma_z = 0.5$, $\sigma_w = 0.1$.

number of iterations until convergence. The intuition behind this result for the EM algorithm is that as the jitter decreases, the ML estimate becomes more linear, and thus, the linear iterations themselves are closer to the true ML estimator; whereas the set of consistent estimates with the bandlimitedness factor becomes narrower and more difficult to find as the additive noise decreases. Increasing the oversampling factor affects the rate of convergence similarly to decreasing σ_w, as it also further constrains the set of acceptable $\mathbf{x}$. For the iterative MAP/ML and BLS/ML estimators, varying these parameters has a similar effect on the shape of the joint distribution $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$, so these estimators behave similarly. Note that the iterative MAP/ML estimator has extremely slow convergence relative to the other estimators, whereas the BLS/ML variant has very rapid convergence. Despite the rapid convergence behavior, in actuality, the BLS/ML estimator is not guaranteed to converge; failures during testing would adversely impact the algorithm's performance. Because of these negative properties of the MAP/ML and BLS/ML algorithms, only the EM algorithm will be evaluated for performance in the following sections.

Before evaluating the EM algorithm's performance, however, the sensitivity of this algorithm to initial conditions should also be studied. Since all these algorithms attempt to find local maxima of the objective function, the initial conditions can dictate which local maximum the algorithm converges to, as well as the rate of convergence. The effects of the initial conditions on the different algorithms are quantified according to the resulting squared-error of the results. The error is plotted for each of 25 trials, each using four sets of initial conditions: the no-jitter linear estimate $\mathbf{H}(\mathbf{0})^{+}\mathbf{y}$, the true value of $\mathbf{x}$, and two randomly generated values of $\mathbf{x}$. Two examples demonstrating opposite sensitivities to initial conditions for the EM algorithm are shown in Figure 3-3.

The EM algorithm's sensitivity varies with the shape of the likelihood function; a likelihood function with many extrema increases the number of regions of attraction, thus increasing the algorithm's sensitivity to initial conditions. The bumpiness of the likelihood function increases with larger σ_z and smaller σ_w. Increasing the oversampling factor has the same effect as increasing σ_z, since the jitter is larger relative to the spacing between the samples. However, since random initialization appears to do generally worse than the others, the linear (no-jitter) estimate is the initial condition of choice. This choice has the additional benefit that the result will be guaranteed to have a likelihood no worse than the linear (no-jitter) efficient estimate.

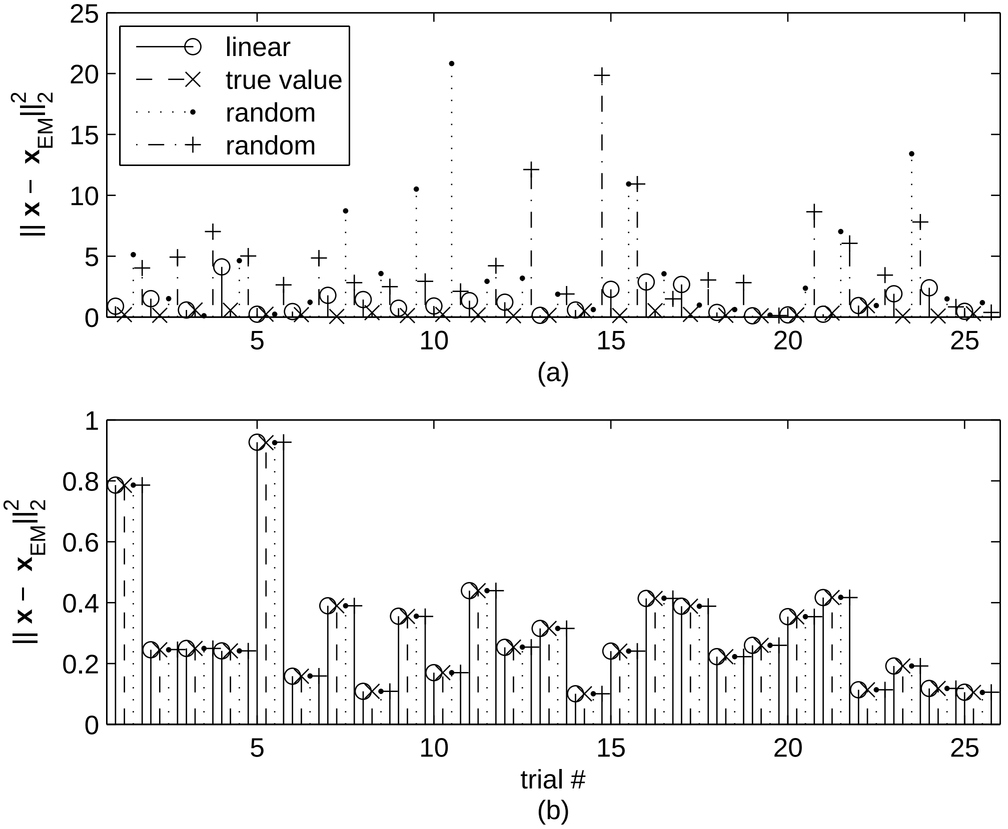


Figure 3-3: Sensitivity to Initial Conditions of EM Algorithm: The resulting error of using the EM algorithm on four different initial points for 25 tests is shown for $K = 5$ parameters and (a) $M = 2$, $\sigma_z = 0.75$, $\sigma_w = 0.1$, and (b) $M = 2$, $\sigma_z = 0.25$, $\sigma_w = 0.25$.

3.5.2 Linear vs. iterative methods

Intuitively, the greatest potential for improvement relative to the (known/no jitter) linear estimator is when the jitter dominates the effect of the additive noise, i.e. when σ_z is large relative to σ_w. However, the iterative algorithms become less accurate as σ_z increases, due to greater quadrature error and the likelihood function becoming more oscillatory. The first of these sources of error can be mitigated by choosing a finer quadrature (increasing I). Increasing the oversampling factor further reduces the impact of both the jitter and the additive noise, causing the likelihood function to be better behaved (more smooth). Thus, the greatest potential for improvement is in a region where $\sigma_z \gg \sigma_w$ and M is large.

Obviously, the amount of jitter an iterative algorithm can tolerate, for a particular choice of additive noise variance σ_w^2 and oversampling factor M, is an important performance measure. The ability to tolerate jitter is determined by comparing the resulting MSE against that of a baseline algorithm, such as the (no-jitter) linear efficient estimator. As shown in Figures 3-4, 3-5, and 3-6, there is a region of σ_z for each set of parameters σ_w and M that results in an improved (decreased) MSE. There are several consequences of this improvement that designers can take advantage of:

1. A designer can use the EM algorithm to achieve the same MSE performance with increased jitter, reducing the required accuracy of the clock, and hence, the power consumption of the ADC.

2. The EM algorithm can be employed to reduce the amount of oversampling required to achieve a certain performance threshold with jitter. Thus, the clock rate decreases, resulting in lower power consumption.

These simulations were performed using 500 random trials to approximate the MSE (each trial consists of randomly chosen parameters $\mathbf{x}$ and noise $\mathbf{z}, \mathbf{w}$) for each set of salient parameters: the oversampling factor M, and the noise standard deviations σ_z and σ_w. The results demonstrate that the EM algorithm is indeed a worthwhile alternative over the (no-jitter) linear efficient estimator to achieve performance gain or lower power consumption.

3.5.3 CRB vs. linear and EM algorithms

The Cramér-Rao lower bound (CRB) provides a bound for how well (in the MSE sense) an unbiased estimator can perform. Although the CRB technically does not apply to biased estimators, it is a helpful metric for determining whether a proposed algorithm is satisfactory. Here, the CRB is estimated using the method in Section 3.4 and plotted for different values of M, σ_z, and σ_w, for a randomly chosen value of $\mathbf{x}$. Then, the approximate MSE for both the unbiased linear estimator in (3.4) and the EM algorithm is compared to the CRB; the result of one such comparison is shown in Figure 3-7.

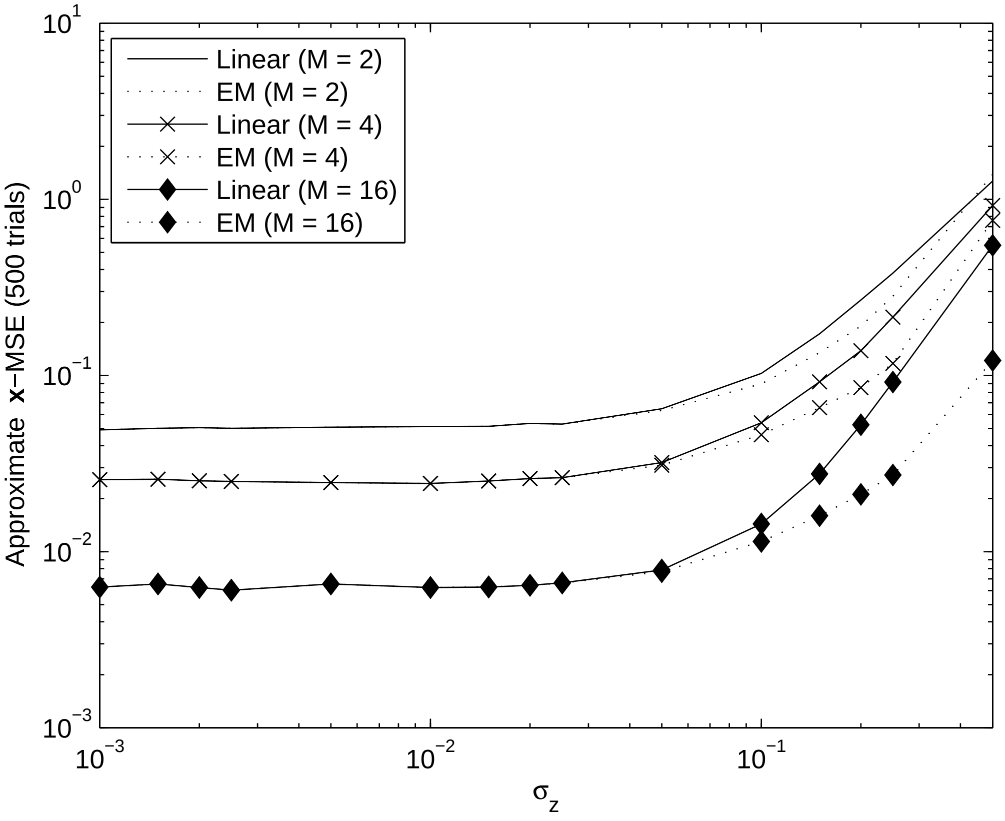


Figure 3-4: Comparing the EM algorithm against the (no-jitter) linear efficient estimator for $K = 10$, $\sigma_w = 0.1$.

Note that for $\sigma_z \ll \sigma_w$, both estimators achieve the Cramér-Rao bound. Also, the EM algorithm continues to achieve the CRB for a range of larger σ_z, whereas the MSE of the linear unbiased estimator increases more rapidly in this region. For $\sigma_z \gg \sigma_w$, both estimators increase more rapidly than the CRB.

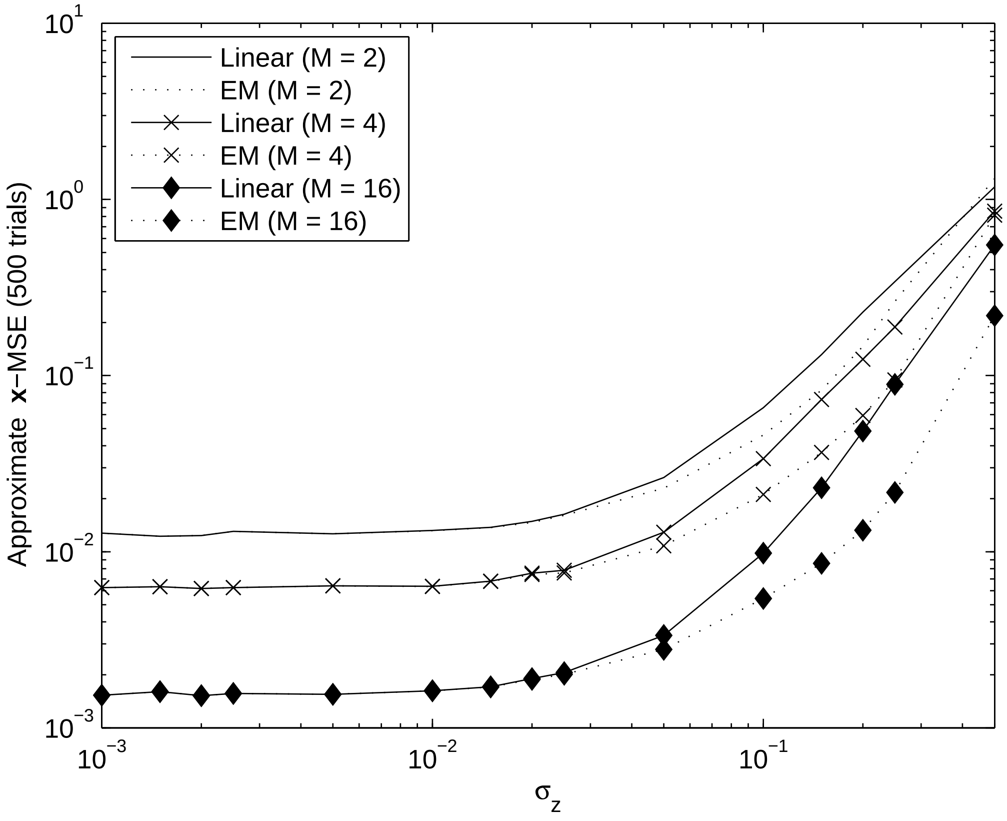


Figure 3-5: Comparing the EM algorithm against the (no-jitter) linear efficient estimator for $K = 10$, $\sigma_w = 0.05$.

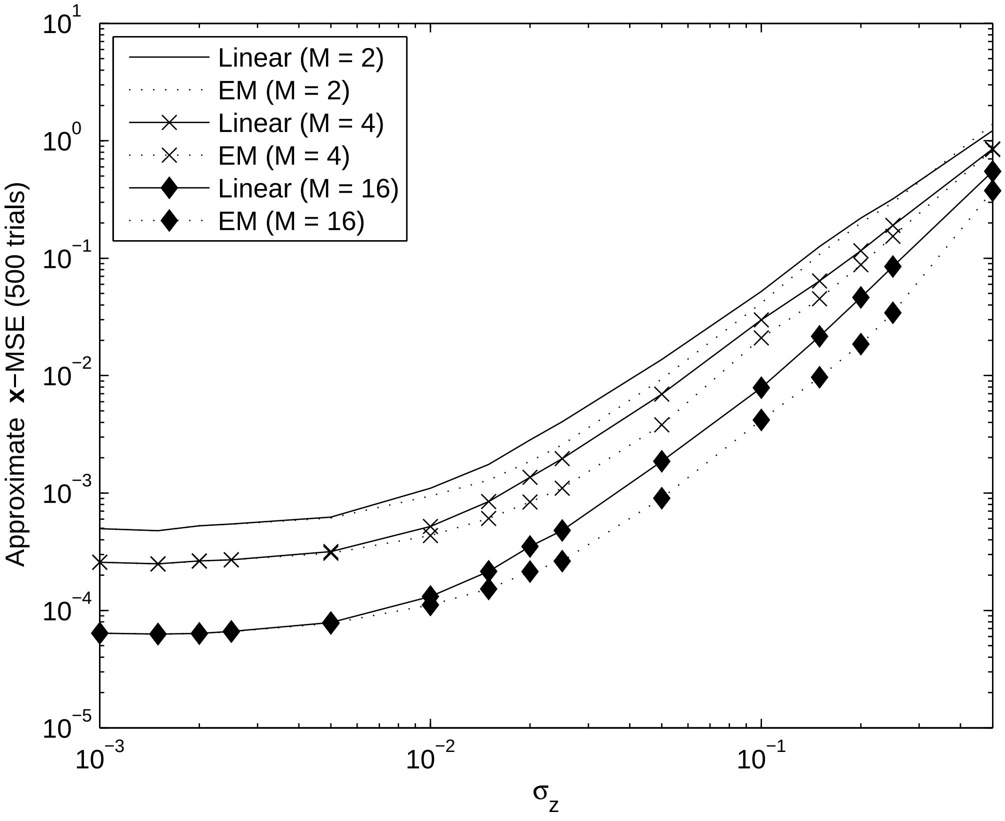


Figure 3-6: Comparing the EM algorithm against the (no-jitter) linear efficient estimator for $K = 10$, $\sigma_w = 0.01$.

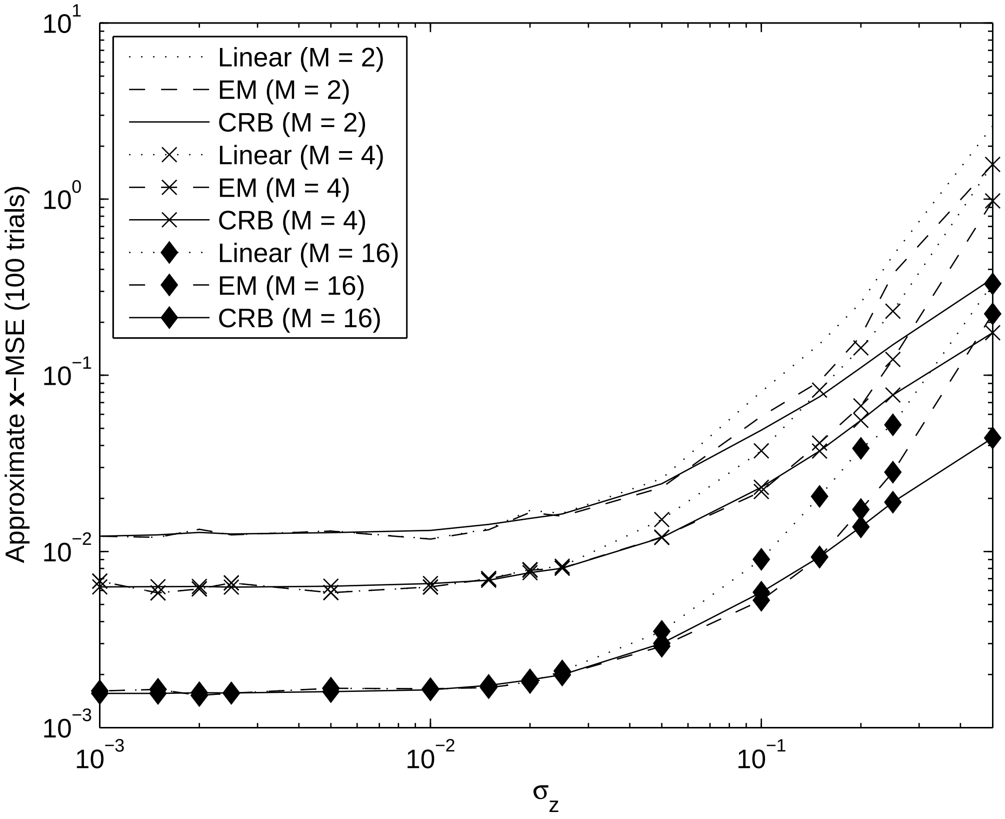


Figure 3-7: The Cramér-Rao lower bound compared against the linear unbiased estimator in (2.10) and the EM algorithm for $K = 10$ parameters chosen randomly, and $\sigma_w = 0.05$. The CRB was approximated using 500 samples from $p(y_n; \mathbf{x})$, and the estimators' MSE performance was estimated using 100 trials each for the same $\mathbf{x}$ and randomly generated noise.

Chapter 4

Bayesian Parameter Estimation

While the results of the previous chapter certainly are encouraging, including a prior on the signal parameters should further improve estimation performance. This chapter also focuses on estimating the parameters $\mathbf{x}$ of a signal using the observation model discussed in the introduction in (1.9). Whereas the previous chapter assumed no model on the parameters, this chapter supposes that the parameters are generated by a random process. Like non-random parameter estimation, the objective of the Bayesian estimation problem is to minimize some cost function, such as the expected squared-error (MSE) of the result. The Bayes Least Squares estimator will be used to accomplish this. Since the Bayes Least Squares estimate is difficult to compute, both deterministic and stochastic methods for approximating this estimator are employed.

For consistency, one prior will be used throughout the discussion in this chapter. While a more specific characterization of the actual input signal would allow for a more accurate model, and hence a better estimate, the maximum entropy model is the "hardest" prior to use in the sense it provides the minimum structure on the unknown parameters. Also, in the maximum entropy case, the parameters should be independent to maximize the joint entropy. Thus, while a Gaussian prior would simplify many calculations and guarantee that the baseline no-jitter linear estimator is the no-jitter BLS estimator, the maximum entropy model is preferred so the problem is approached with the most difficult prior. The Uniform distribution has the maximum entropy of all distributions with finite support [7]. Without loss of generality, then, we may assume that the signal parameters are generated by an iid Uniform$(-1,1)$ random process. The independence and zero-mean assumptions also are very useful throughout this chapter for simplifying the derivations.

The posterior density can be derived using Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int \cdots \int p(\mathbf{y}|\mathbf{x}')p(\mathbf{x}')\,d\mathbf{x}'}. \tag{4.1}$$

Due to the assumption of independence of $\mathbf{x}$, and the assumption that the noise sources are also independent, the prior and likelihood are both separable. Therefore,

$$p(\mathbf{x}|\mathbf{y}) = \frac{\prod_{n=0}^{N-1} p(y_n|\mathbf{x}) \prod_{k=0}^{K-1} p(x_k)}{\int \cdots \int \prod_{n=0}^{N-1} p(y_n|\mathbf{x}') \prod_{k=0}^{K-1} p(x'_k)\,d\mathbf{x}'}. \tag{4.2}$$

The likelihood function $p(y_n|\mathbf{x})$ does not have a closed form, but it can be expressed using integration as in (2.4).

4.1 Linear Estimators

The form of the linear estimator for $\mathbf{x}$ that minimizes the MSE is stated in (2.23). Since $\mathbb{E}[\mathbf{x}] = 0$ and $\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{H}(\mathbf{z})]\mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{w}] = 0$,

$$\hat{\mathbf{x}}_{LLS}(\mathbf{y}) = \mathbb{E}[\mathbf{x}(\mathbf{H}(\mathbf{z})\mathbf{x} + \mathbf{w})^T](\mathbb{E}[(\mathbf{H}(\mathbf{z})\mathbf{x} + \mathbf{w})(\mathbf{H}(\mathbf{z})\mathbf{x} + \mathbf{w})^T])^{-1}\mathbf{y}. \tag{4.3}$$

Since the parameters and noise sources are all independent, and $\mathbf{\Lambda_x} = \sigma_x^2 \mathbf{I}$ is a scalar matrix, where $\sigma_x^2 = 1/3$ is the variance of Uniform$(-1,1)$, the above expression simplifies to

$$\hat{\mathbf{x}}_{LLS}(\mathbf{y}) = \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \left(\mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I} \right)^{-1} \mathbf{y}. \tag{4.4}$$

The expectations in (4.4) do not have a closed form. Gauss-Hermite quadrature or another approximation method can be employed to approximate the expectations, and the close-to-optimal linear estimator can be evaluated numerically.

The error covariance $\mathbf{\Lambda}_{LLS}$ has a similar standard expression, given in (2.25), which reduces to

$$\mathbf{\Lambda}_{LLS} = \sigma_x^2 \left(\mathbf{I} - \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \left(\mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I} \right)^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})] \right). \tag{4.5}$$

The optimal linear estimator for the case with known jitter is similar to (4.4):

$$\hat{\mathbf{x}}_{LLS,\mathbf{z}=\mathbf{z}^*}(\mathbf{y}) = \mathbf{H}(\mathbf{z}^*)^T \left(\mathbf{H}(\mathbf{z}^*)\mathbf{H}(\mathbf{z}^*)^T + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I} \right)^{-1} \mathbf{y}. \tag{4.6}$$

When there is no jitter, this reduces to

$$\hat{\mathbf{x}}_{LLS,\mathbf{z}=\mathbf{0}}(\mathbf{y}) = \mathbf{H}(\mathbf{0})^T \left(\mathbf{H}(\mathbf{0})\mathbf{H}(\mathbf{0})^T + \frac{\sigma_w^2}{\sigma_x^2}\mathbf{I} \right)^{-1} \mathbf{y}. \tag{4.7}$$

Substantial literature already exists exploring the optimal linear estimator problem in detail. In [1], Balakrishnan derives the coefficients of the optimal convolution kernel for reconstructing the original bandlimited signal. Whereas in the case of simple bandlimited reconstruction without jitter, the optimal reconstruction operation convolves the samples with a sinc function, Balakrishnan proves that the coefficients that minimize the MSE of the reconstructed signal $\hat{x}(t) = \sum_n a_n(t) y_n$ are determined by the characteristic function of the jitter $C(\Omega) \triangleq \mathbb{E}\left[e^{j\Omega z}\right]$ and the discrete-time power spectral densities $\phi_{xx}(\omega)$ and $\phi_{yy}(\omega) = \left|C\left(\frac{\omega}{T}\right)\right|^2 \phi_{xx}(\omega) + a^2$:

$$a_n(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} C\left(\frac{\omega}{T}\right) \frac{\phi_{xx}(\omega)}{\left|C\left(\frac{\omega}{T}\right)\right|^2 \phi_{xx}(\omega) + a^2} e^{-j\omega(n-t/T)}\, d\omega, \tag{4.8}$$

where $a^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(1 - \left|C\left(\frac{\omega}{T}\right)\right|^2\right)\phi_{xx}(\omega)\, d\omega$. When $z = 0$, (4.8) reduces to $\mathrm{sinc}(t/T - n)$, as expected. Balakrishnan also contends in [1] that the optimal non-linear estimate is actually linear. However, Balakrishnan considers only non-linear operators on individual data values (he does not include non-linear functions of the entire vector $\mathbf{y}$). Specifically, Balakrishnan considers only non-linear estimators of the form

$$\hat{x}(t) = \sum_n \sum_{k=0}^{\infty} a_k^n(t) f_k(y_n), \tag{4.9}$$

where $f_k(\cdot)$ is any non-linear function. Balakrishnan then shows that the MMSE estimator must satisfy

$$\mathbb{E}[x(t)|y_n] = \mathbb{E}\left[\sum_m \sum_{k=0}^{\infty} a_k^m(t) f_k(y_m)|y_n\right], \tag{4.10}$$

and as a result, the optimal non-linear estimator has $a_k^m(t) = 0$ for all $k \neq 1$, $f_1(y)$ is linear, and the expression for $a_1^m(t)$ is identical to (4.8).

In [24], Liu and Stanley consider the error introduced by the jitter when using a standard low-pass/sinc-interpolation filter to reconstruct a wide-sense stationary signal bandlimited to $\pm\Omega_B$. For the case when the jitter is independent with variance σ_z, the error ϵ^2 is bounded by

$$\epsilon^2 \leq 2(R_{xx}(0) - R_{xx}(\sigma_z)) \leq R_{xx}(0)\Omega_B^2\sigma_z^2, \tag{4.11}$$

where the second inequality holds for small σ_z (i.e. $\Omega_B\sigma_z \ll \pi$). They also compute the error for when the jitter is Gaussian and the input signal power spectral density $S_{xx}(\Omega) = \frac{1}{2\Omega_B}$ for $|\Omega| < \Omega_B$:

$$\epsilon^2 = 2 - \frac{\sqrt{2\pi}}{\Omega_B\sigma_z}\,\mathrm{erf}\left(\frac{\Omega_B\sigma_z}{\sqrt{2}}\right). \tag{4.12}$$

For small enough x, $\mathrm{erf}(x) \approx \frac{2}{\sqrt{\pi}}\left(x - \frac{x^3}{3}\right)$, so

$$\epsilon^2 \approx 2 - \frac{\sqrt{2\pi}}{\Omega_B\sigma_z}\left(\frac{2}{\sqrt{\pi}}\right)\left(\frac{\Omega_B\sigma_z}{\sqrt{2}}\right) + \frac{\sqrt{2\pi}}{\Omega_B\sigma_z}\left(\frac{2}{3\sqrt{\pi}}\right)\left(\frac{\Omega_B\sigma_z}{\sqrt{2}}\right)^3 = \frac{1}{3}\Omega_B^2\sigma_z^2. \tag{4.13}$$

Note that in both cases, the squared error scales approximately with σ_z^2. The rest of this chapter will be devoted to developing algorithms that improve on this performance bound.

4.2 BLS Estimation Using Taylor Series

The BLS estimator can be expressed in terms of the likelihood function, as in (2.20). The likelihood function can be expressed in terms of $p(\mathbf{y}|\mathbf{z},\mathbf{x})$, resulting in

$$\hat{\mathbf{x}}_{BLS}(\mathbf{y}) = \frac{\int \mathbf{x} \left[\int \mathcal{N}(\mathbf{y};\mathbf{H}(\mathbf{z})\mathbf{x};\sigma_w^2\mathbf{I})\mathcal{N}(\mathbf{z};\mathbf{0},\sigma_z^2\mathbf{I})\,d\mathbf{z}\right] p(\mathbf{x})\,\mathbf{dx}}{\int \left[\int \mathcal{N}(\mathbf{y};\mathbf{H}(\mathbf{z})\mathbf{x};\sigma_w^2\mathbf{I})\mathcal{N}(\mathbf{z};\mathbf{0},\sigma_z^2\mathbf{I})\,d\mathbf{z}\right] p(\mathbf{x})\,\mathbf{dx}}. \tag{4.14}$$

This N-dimensional integration can be approximated in any number of ways. One method whose complexity does not scale exponentially with N involves using a partial Taylor series expansion of $f(\mathbf{z}) = \mathcal{N}(\mathbf{y};\mathbf{H}(\mathbf{z})\mathbf{x};\sigma_w^2\mathbf{I})$ about $\mathbf{z} = 0$. The first, second, and third mixed partial derivatives are listed below:

$$\frac{\partial f(\mathbf{z})}{\partial z_n} = f(\mathbf{z})\frac{1}{\sigma_w^2}(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \tag{4.15}$$

$$\begin{aligned}\frac{\partial^2 f(\mathbf{z})}{\partial z_n \partial z_m} = f(\mathbf{z}) \Bigg(&\frac{1}{\sigma_w^4}(y_m - \mathbf{h}_m(z_m)^T\mathbf{x})\;(\mathbf{h}_m'(z_m)^T\mathbf{x})(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \\ &+ \frac{\delta_{n-m}}{\sigma_w^2}(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n''(z_n)^T\mathbf{x}) - \frac{\delta_{n-m}}{\sigma_w^2}(\mathbf{h}_n'(z_n)^T\mathbf{x})^2\Bigg)\end{aligned} \tag{4.16}$$

$$\begin{aligned}\frac{\partial^3 f(\mathbf{z})}{\partial z_n \partial z_m \partial z_l} = f(\mathbf{z}) \Bigg(&\frac{1}{\sigma_w^6}(y_l - \mathbf{h}_l(z_l)^T\mathbf{x})(\mathbf{h}_l'(z_l)^T\mathbf{x})(y_m - \mathbf{h}_m(z_m)^T\mathbf{x}) \\ &\cdot (\mathbf{h}_m'(z_m)^T\mathbf{x})(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \\ &+ \frac{\delta_{m-l}}{\sigma_w^4}(y_m - \mathbf{h}_m(z_m)^T\mathbf{x})(\mathbf{h}_m''(z_m)^T\mathbf{x})(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \\ &- \frac{\delta_{m-l}}{\sigma_w^4}(\mathbf{h}_m'(z_m)^T\mathbf{x})(\mathbf{h}_m'(z_m)^T\mathbf{x})(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \\ &+ \frac{\delta_{n-l}}{\sigma_w^4}(y_m - \mathbf{h}_m(z_m)^T\mathbf{x})(\mathbf{h}_m'(z_m)^T\mathbf{x})(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n''(z_n)^T\mathbf{x}) \\ &- \frac{\delta_{n-l}}{\sigma_w^4}(y_m - \mathbf{h}_m'(z_m)^T\mathbf{x})(\mathbf{h}_m'(z_m)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x})(\mathbf{h}_n'(z_n)^T\mathbf{x}) \\ &+ \frac{\delta_{n-m}\delta_{n-l}}{\sigma_w^2}(y_n - \mathbf{h}_n(z_n)^T\mathbf{x})(\mathbf{h}_n^{(3)}(z_n)^T\mathbf{x}) \\ &- \frac{\delta_{n-m}\delta_{n-l}}{\sigma_w^2}(\mathbf{h}_n'(z_n)^T\mathbf{x})(\mathbf{h}_n''(z_n)^T\mathbf{x}) \\ &- \frac{2\delta_{n-m}\delta_{n-l}}{\sigma_w^2}(\mathbf{h}_n'(z_n)^T\mathbf{x})(\mathbf{h}_n''(z_n)^T\mathbf{x})\Bigg)\end{aligned} \tag{4.17}$$

The 2nd-order Taylor series expansion is equal to

$$f(\mathbf{z}) = f(\mathbf{0}) + \nabla f(\mathbf{0})\mathbf{z} + \frac{1}{2}\mathbf{z}^T\nabla^2 f(\mathbf{0})\mathbf{z} + \sum_{n,m,l=0}^{N-1} \frac{\partial^3 f(\xi_{n,m,l})}{\partial z_n \partial z_m \partial z_l} z_n z_m z_l \frac{1}{(n,m,l)!} \qquad (4.18)$$

for some $\xi_{n,m,l}$, where ∇f and $\nabla^2 f$ are the gradient (row) vector and Hessian matrix, respectively, and $(n, m, l)!$ is an extension of the factorial operator for mixed-partial derivatives:

$$(n,m,l)! = \begin{cases} 6 & n = m = l; \\ 2 & n = m \neq l; n = l \neq m; m = l \neq n; \\ 1 & \text{otherwise} \end{cases} \cdot \qquad (4.19)$$

Using the following lemma, as well as the triangle inequality, the partial third derivatives in (4.17) can be (loosely) bounded.

Lemma 1 *Let $f(t)$ be a real-valued periodic function bandlimited to frequency $\pm\pi/T$, with period KT. Also, let $f_k = f(t_k)$, where $t_k = t_0 + kT$ and t_0 is arbitrary. Then, $\sum_{k=0}^{K-1} |f_k|^2$ does not depend on the value of t_0.*

Proof. This lemma follows directly from Parseval's Theorem: suppose $f(t)$ has the Fourier transform $F_c(\Omega)$. The sampled f_k has discrete-time Fourier transform $F(\omega) = e^{-j\Omega t_0} F_c(\Omega)$ where $\Omega = \omega/T$. Then,

$$\sum_{k=0}^{K-1} |f_k|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |F(\omega)|^2\, d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} |e^{-j\omega t_0/T}||F_c(\omega/T)|^2\, d\omega. \qquad (4.20)$$

Since $|e^{-j\omega t_0/T}| = 1$, the energy of the signal does not depend on t_0. □

Applying Lemma 1 to bound $||h_n(z_n)||_2$ and its derivatives, the following quantities can be bounded by fixed constants $\bar{h}^{(0)}, \bar{h}^{(1)}, \bar{h}^{(2)}, \bar{h}^{(3)}$:

$$|y_n - \mathbf{h}_n(z_n)^T \mathbf{x}| \leq |y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2 \tag{4.21}$$

$$|\mathbf{h}_n'(z_n)^T \mathbf{x}| \leq \bar{h}^{(1)} ||\mathbf{x}||_2 \tag{4.22}$$

$$|\mathbf{h}_n''(z_n)^T \mathbf{x}| \leq \bar{h}^{(2)} ||\mathbf{x}||_2 \tag{4.23}$$

$$|\mathbf{h}_n^{(3)}(z_n)^T \mathbf{x}| \leq \bar{h}^{(3)} ||\mathbf{x}||_2 \tag{4.24}$$

Together with the triangle inequality, and the fact that $|f(\mathbf{z})| \leq 1/(2\pi\sigma_w^2)^{N/2}$, the third mixed partial derivatives $\frac{\partial^3 f(\mathbf{z})}{\partial z_n \partial z_m \partial z_l}$ can be bounded by $F_{n,m,l}$:

$$\begin{aligned}
F_{n,m,l} \triangleq \frac{1}{(2\pi\sigma_w^2)^{N/2}} \Bigg(& \frac{1}{\sigma_w^6} (|y_l| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2) \\
& \cdot (|y_m| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2)(|y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{m-l}}{\sigma_w^4} (|y_m| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(2)} ||\mathbf{x}||_2)(|y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{m-l}}{\sigma_w^4} (\bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2)(|y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{n-l}}{\sigma_w^4} (|y_m| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2)(|y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(2)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{n-l}}{\sigma_w^4} (|y_m| + \bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(1)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{n-m}\delta_{n-l}}{\sigma_w^2} (|y_n| + \bar{h}^{(0)} ||\mathbf{x}||_2)(\bar{h}^{(3)} ||\mathbf{x}||_2) \\
& + \frac{\delta_{n-m}\delta_{n-l}}{\sigma_w^2} (\bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(2)} ||\mathbf{x}||_2) + \frac{2\delta_{n-m}\delta_{n-l}}{\sigma_w^2} (\bar{h}^{(1)} ||\mathbf{x}||_2)(\bar{h}^{(2)} ||\mathbf{x}||_2) \Bigg).
\end{aligned} \tag{4.25}$$

Now, consider integrating $f(\mathbf{z})\mathcal{N}(\mathbf{z}; \mathbf{0}, \sigma_z^2 \mathbf{I})$ using the 2nd-order partial Taylor series expansion $f^*(\mathbf{z})$ ((4.18) minus the error terms). Then, the integration error can be bounded using (4.25):

$$\left| \int (f(\mathbf{z}) - f^*(\mathbf{z})) p(\mathbf{z}) \, d\mathbf{z} \right| \leq \sum_{n,m,l} \int \left| \frac{\partial^3 f(\xi_{n,m,l})}{\partial z_n \partial z_m \partial z_l} \frac{z_n z_m z_l}{(n,m,l)!} \right| p(\mathbf{z}) \, d\mathbf{z} \tag{4.26}$$

$$\leq \sum_{n,m,l} \frac{F_{n,m,l}}{(n,m,l)!} \int |z_n z_m z_l| p(\mathbf{z}) \, d\mathbf{z}. \tag{4.27}$$

It can be shown using integration by parts that

$$\int |z_n|^3 \mathcal{N}(\mathbf{z}; \mathbf{0}, \sigma_z^2 \mathbf{I})\, d\mathbf{z} = 2\sqrt{\frac{2}{\pi}}\sigma_z^3 \tag{4.28}$$

$$\int |z_n| z_m^2 \mathcal{N}(\mathbf{z}; \mathbf{0}, \sigma_z^2 \mathbf{I})\, d\mathbf{z} = \sqrt{\frac{2}{\pi}}\sigma_z^3 \tag{4.29}$$

$$\int |z_n z_m z_l| \mathcal{N}(\mathbf{z}; \mathbf{0}, \sigma_z^2 \mathbf{I})\, d\mathbf{z} = \frac{2}{\pi}\sqrt{\frac{2}{\pi}}\sigma_z^3. \tag{4.30}$$

Integrating the partial Taylor series expansion itself with respect to the probability measure $p(\mathbf{z})$ uses the moments of the Gaussian. The resulting expression can be plugged back into (4.14) to yield

$$\hat{\mathbf{x}}_{BLS}(\mathbf{y}) \approx \frac{\int \mathbf{x} \left(f(\mathbf{0}) + \frac{\sigma_z^2}{2} \operatorname{tr}(\nabla^2 f(\mathbf{0})) \right) p(\mathbf{x})\, \mathbf{dx}}{\int \left(f(\mathbf{0}) + \frac{\sigma_z^2}{2} \operatorname{tr}(\nabla^2 f(\mathbf{0})) \right) p(\mathbf{x})\, \mathbf{dx}}. \tag{4.31}$$

where $f(\mathbf{0})$ and $\operatorname{tr}(\nabla^2 f(\mathbf{0}))$ depend on $\mathbf{x}$ as a polynomial times a Gaussian kernel. In particular, $f(\mathbf{0})$ is simply the Gaussian kernel $\mathcal{N}(\mathbf{y}; \mathbf{H}(\mathbf{0})\mathbf{x}, \sigma_w^2 \mathbf{I})$, and $\operatorname{tr}(\nabla^2 f(\mathbf{0}))$, which is described in (4.16), is a 4th-degree polynomial in $\mathbf{x}$, multiplied by the same Gaussian kernel.

Since $\mathbf{H}(\mathbf{0})^T \mathbf{H}(\mathbf{0}) = M\mathbf{I}$, as a result of the orthogonality of the psinc basis and Lemma 1, the exponent of $f(\mathbf{0})$, which is also found in its derivatives, can be easily re-expressed to form a Gaussian kernel for $\mathbf{x}$ instead of $\mathbf{y}$:

$$\frac{||\mathbf{y} - \mathbf{H}(\mathbf{0})\mathbf{x})||_2^2}{\sigma_w^2} = \frac{(\mathbf{y}^T\mathbf{y} + \mathbf{x}^T M \mathbf{x} - 2(\mathbf{H}(\mathbf{0})^T\mathbf{y})^T\mathbf{x})}{\sigma_w^2} \tag{4.32}$$

$$= \frac{M||\mathbf{x} - \frac{\mathbf{H}(\mathbf{0})^T}{M}\mathbf{y})||_2^2 + \mathbf{y}^T \left(\mathbf{I} - \frac{\mathbf{H}(\mathbf{0})\mathbf{H}(\mathbf{0})^T}{M} \right) \mathbf{y}}{\sigma_w^2}. \tag{4.33}$$

Thus,

$$\mathcal{N}(\mathbf{y}; \mathbf{H}(\mathbf{0})\mathbf{x}, \sigma_w^2 \mathbf{I}) = \frac{\exp\left[-\frac{\mathbf{y}^T \left(M\mathbf{I} - \mathbf{H}(\mathbf{0})\mathbf{H}(\mathbf{0})^T \right)\mathbf{y}}{2M\sigma_w^2} \right]}{(2\pi\sigma_w^2)^{(N-K)/2} M^{K/2}} \mathcal{N}\left(\mathbf{x}; \frac{\mathbf{H}(\mathbf{0})^T}{M}\mathbf{y}, \frac{\sigma_w^2}{M}\mathbf{I} \right). \tag{4.34}$$

The resulting polynomials in (4.31) are of degree five and four, in the numerator and denominator, respectively. The resulting integral also has a closed form, albeit in terms of the Gaussian error function, because the integral is over a finite interval instead, yielding incomplete moments of the Gaussian kernel $\mathcal{N}(\mathbf{x}; \frac{\mathbf{H}(\mathbf{0})^T}{M}\mathbf{y}, \frac{\sigma_w^2}{M}\mathbf{I})$.

Below are the coefficients that multiply the mixed terms in the polynomials in $\mathbf{x}$ for the top and bottom integrals in (4.31). Since the top is different from the bottom only in that the top has an extra factor of $\mathbf{x}$ multiplying everything, the coefficients will be indexed according to their place in the bottom polynomial. Indexing the coefficients as $c^{(0)}$ multiplying (1), $c_i^{(1)}$ multiplying x_i, $c_{i,j}^{(2)}$ multiplying $x_i x_j$ (considered different from $x_j x_i$), $c_{i,j,k}^{(3)}$ multiplying $x_i x_j x_k$, $c_{i,j,k,l}^{(4)}$ multiplying $x_i x_j x_k x_l$,

$$c^{(0)} = 1 \tag{4.35}$$

$$c_i^{(1)} = \frac{\sigma_z^2}{2\sigma_w^2} \sum_{n=0}^{N-1} y_n \mathbf{H}_{n,i}^{(2)} \tag{4.36}$$

$$c_{i,j}^{(2)} = \frac{\sigma_z^2}{2\sigma_w^2} \sum_{n=0}^{N-1} \left(\frac{y_n^2}{\sigma_w^2} - 1\right) \mathbf{H}_{n,i}^{(1)}\mathbf{H}_{n,j}^{(1)} - \mathbf{H}_{n,i}^{(0)}\mathbf{H}_{n,j}^{(2)} \tag{4.37}$$

$$c_{i,j,k}^{(3)} = -\frac{\sigma_z^2}{\sigma_w^4} \sum_{n=0}^{N-1} y_n \mathbf{H}_{n,i}^{(0)}\mathbf{H}_{n,j}^{(1)}\mathbf{H}_{n,k}^{(1)} \tag{4.38}$$

$$c_{i,j,k,l}^{(4)} = -\frac{\sigma_z^2}{2\sigma_w^4} \sum_{n=0}^{N-1} \mathbf{H}_{n,i}^{(0)}\mathbf{H}_{n,j}^{(0)}\mathbf{H}_{n,k}^{(1)}\mathbf{H}_{n,l}^{(1)}, \tag{4.39}$$

where $\mathbf{H}_{n,k}^{(m)} = \left[\frac{d^m}{dz^m} \operatorname{psinc}_K(z)\right]_{z=n/M-k}$.

Let $\hat{\mathbf{m}}_x = \frac{1}{M}\mathbf{H}(\mathbf{0})^T\mathbf{y}$ and $\hat{\sigma}_x^2 = \frac{1}{M}\sigma_w^2$. Then, the bottom integral in (4.31) is equal to

$$\frac{1}{(2\pi\sigma_w^2)^{(N-K)/2}M^{K/2}} \Bigg(c^{(0)} \int \mathcal{N}(\mathbf{x}; \hat{\mathbf{m}}_x, \hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx} + \sum_{i=0}^{K-1} c_i^{(1)} \int x_i \mathcal{N}(\mathbf{x}; \hat{\mathbf{m}}_x, \hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}$$
$$+ \sum_{j=0}^{K-1} c_{i,j}^{(2)} \int x_i x_j \mathcal{N}(\mathbf{x}; \hat{\mathbf{m}}_x, \hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx} + \sum_{k=0}^{K-1} c_{i,j,k}^{(3)} \int x_i x_j x_k \mathcal{N}(\mathbf{x}; \hat{\mathbf{m}}_x, \hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}$$
$$+ \sum_{l=0}^{K-1} c_{i,j,k,l}^{(4)} \int x_i x_j x_k x_l \mathcal{N}(\mathbf{x}; \hat{\mathbf{m}}_x, \hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}\Bigg), \tag{4.40}$$

and the top integral in (4.31) is equal to

$$\frac{1}{(2\pi\sigma_w^2)^{(N-K)/2}M^{K/2}}\left(\sum_{i=0}^{K-1} c^{(0)}\int x_i\mathcal{N}(\mathbf{x};\hat{\mathbf{m}}_x,\hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}\right.$$
$$+\sum_{j=0}^{K-1} c_j^{(1)}\int x_ix_j\mathcal{N}(\mathbf{x};\hat{\mathbf{m}}_x,\hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}+\sum_{k=0}^{K-1} c_{j,k}^{(2)}\int x_ix_jx_k\mathcal{N}(\mathbf{x};\hat{\mathbf{m}}_x,\hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}$$
$$+\sum_{l=0}^{K-1} c_{j,k,l}^{(3)}\int x_ix_jx_kx_l\mathcal{N}(\mathbf{x};\hat{\mathbf{m}}_x,\hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}$$
$$\left.+\sum_{m=0}^{K-1} c_{j,k,l,m}^{(4)}\int x_ix_jx_kx_lx_m\mathcal{N}(\mathbf{x};\hat{\mathbf{m}}_x,\hat{\sigma}_x^2\mathbf{I})\,\mathbf{dx}\right). \quad (4.41)$$

While this approximation for the BLS estimator does not require complicated numeric integrations, the complexity scales poorly with K. For $K = 10$, the approximation to the top integral alone involves over 10^5 terms. This poor scalability reduces the number of parameters we reasonably can estimate at a time, and in the event that the approximation is inexact, the scalability gets even worse as more derivatives from the Taylor series of $f(\mathbf{z})$ are added (four more powers of $\mathbf{x}$ for each additional even term in the Taylor series).

4.3 Approximating the BLS Estimator with Gibbs Sampling

Rather than attempt to solve the integral in (4.14) repeatedly for different sample values, augment the parameters with the jitter $\mathbf{z}$ and approximate the joint BLS estimate $\mathbb{E}[\mathbf{x},\mathbf{z}|\mathbf{y}]$ instead. The resulting estimate for $\mathbf{x}$ is equivalent to the BLS estimate $\mathbb{E}[\mathbf{x}|\mathbf{y}]$; the estimate for $\mathbf{z}$ can be ignored.

For this scenario, Gibbs sampling can be used, iteratively drawing samples from $p(\mathbf{x}|\mathbf{y},\mathbf{z})$ and $p(\mathbf{z}|\mathbf{y},\mathbf{x})$. The general algorithm consists of alternatingly sampling from each of these distributions in some (not necessarily natural) order and averaging all the samples for each parameter after a sufficient "burn-in" period has elapsed.

If the underlying Markov chain is irreducibile and aperiodic, Gibbs sampling will eventually converge to a steady-state distribution that is the joint posterior distribution $p(\mathbf{x}, \mathbf{z}|\mathbf{y})$. In [13], Gelfand points out that as long as the joint pdf is continuous and bounded, and the support is connected, the Gibbs sampler will converge. However, the rate of convergence depends on how the variables are split. In [28], Smith and Roberts also discuss the role of correlation between the parameters; to maximize the rate of convergence, highly correlated variables should be sampled together at once. Once the Markov chain is sufficiently close to the stationary distribution, further samples are effectively taken from this joint posterior and can be used to approximate the BLS estimate of these parameters. In the algorithm, I_b represents the time until the Markov chain reaches its steady state, and I represents the number of samples taken afterward from the joint posterior. Experimentation is needed to determine appropriate values of these parameters.

```
Require: $\mathbf{y}, I, I_b$
  $\mathbf{z}^{(0)} = \mathbf{0}$
  $\mathbf{x}^{(0)} = \mathbf{0}$
  for $i = 1 : I + I_b$ do
      $z_0^{(i)} \sim p(\cdot|z_1^{(i-1)}, \ldots, z_{N-1}^{(i-1)}, \mathbf{x}^{(i-1)}, \mathbf{y})$
      $z_1^{(i)} \sim p(\cdot|z_0^{(i)}, z_2^{(i-1)}, \ldots, z_{N-1}^{(i-1)}, \mathbf{x}^{(i-1)}, \mathbf{y})$
      $\vdots$
      $z_{N-1}^{(i)} \sim p(\cdot|z_0^{(i)}, \ldots, z_{N-2}^{(i)}, \mathbf{x}^{(i-1)}, \mathbf{y})$
      $x_0^{(i)} \sim p(\cdot|\mathbf{z}^{(i)}, x_1^{(i-1)}, \ldots, x_{K-1}^{(i-1)}, \mathbf{y})$
      $x_1^{(i)} \sim p(\cdot|\mathbf{z}^{(i)}, x_0^{(i)}, x_2^{(i-1)}, \ldots, x_{K-1}^{(i-1)}, \mathbf{y})$
      $\vdots$
      $x_{K-1}^{(i)} \sim p(\cdot|\mathbf{z}^{(i)}, x_0^{(i)}, \ldots, x_{K-2}^{(i)}, \mathbf{y})$
  end for
  $\hat{\mathbf{x}} = \frac{1}{I} \sum_{i=I_b+1}^{I_b+I} \mathbf{x}^{(i)}$
  $\hat{\mathbf{z}} = \frac{1}{I} \sum_{i=I_b+1}^{I_b+I} \mathbf{z}^{(i)}$
  return $\hat{\mathbf{x}}$, $\hat{\mathbf{z}}$
```

For each parameter, the distribution of that parameter conditioned on the data and other parameters is proportional to the joint density of all the parameters and the data, with the appropriate values of the other parameters and data plugged-in to the expression, so it suffices to sample from the joint distribution instead (keeping the other parameter values fixed, of course). Thus,

$$z_n \sim \mathcal{N}(y_n; \mathbf{h}_n^T(z_n)\mathbf{x}, \sigma_w^2)\mathcal{N}(z_n; 0, \sigma_z^2), \tag{4.42}$$

where $h_n^T(z_n)$ is the nth row of $\mathbf{H}(\mathbf{z})$, and

$$x_k \sim \mathcal{N}(\mathbf{y}; \mathbf{H}(\mathbf{z})\mathbf{x}, \sigma_w^2\mathbf{I}). \tag{4.43}$$

Unfortunately, the density for z_n is not easy to sample from directly; however, a method like rejection sampling can reliably sample from this distribution by sampling from a suitable proposal distribution $q(z_n)$. Here, the propsal density used is $q(\cdot) \triangleq \mathcal{N}(\cdot; 0, \sigma_z^2)$, and $1/c = \sqrt{2\pi\sigma_w^2}$.

Let the vector $\mathbf{x}_{-k}$ represent of all but the kth parameter. Also, let $\mathbf{H}_k(\mathbf{z})$ and $\mathbf{H}_{-k}(\mathbf{z})$ be the kth column and all but the kth column, respectively, of the matrix $\mathbf{H}(\mathbf{z})$. The density for x_k, however, can be sampled using the cdf inversion method. Re-expressing the Normal distribution in (4.43) as a Normal distribution on x_k, the overall distribution becomes a truncated Normal distribution; i.e.

$$x_k \sim \mathcal{N}\left(x_k; \mu_k, \sigma_k^2\right) U(x_k; -1, 1) \tag{4.44}$$

where

$$\mu_k = \frac{\mathbf{H}_k(\mathbf{z})^T(\mathbf{y} - \mathbf{H}_{-k}(\mathbf{z})\mathbf{x}_{-k})}{\mathbf{H}_k(\mathbf{z})^T\mathbf{H}_k(\mathbf{z})} \tag{4.45}$$

$$\sigma_k^2 = \frac{\sigma_w^2}{\mathbf{H}_k(\mathbf{z})^T\mathbf{H}_k(\mathbf{z})} \tag{4.46}$$

Now, denote the zero-mean unit-variance version of x_k by χ_k. Then, χ_k has a truncated standard Normal distribution between $a_k = \frac{-1-\mu_k}{\sigma_k}$ and $b_k = \frac{1-\mu_k}{\sigma_k}$. Thus, this distribution can be sampled by generating a uniform random variable u between $\Phi(a)$ and $\Phi(b)$. Then, the inverse cdf can be used to convert u to the corresponding value of χ_k. When the mean μ_k lies within the finite support $[a_k, b_k]$, this method is very efficient and stable; however, when the support is mainly the tail of the Normal distribution, the cdf is flat throughout the region, and inversion fails or results in a poor approximation to the truncated Gaussian random variable. Robert suggests employing rejection sampling in this region and describes the optimal proposal distribution for the univariate standard normal case in [26]. Define the shifted exponential distribution $\text{Exp}(x; \alpha, \mu)$ to have the pdf

$$\text{Exp}(x; \alpha, \mu) = \begin{cases} \alpha e^{-\alpha(x-\mu)} & x \geq \mu, \\ 0 & x < \mu \end{cases}. \tag{4.47}$$

Then, the optimal proposal distribution for rejection sampling when the support interval $[a, b]$ lies to the right of zero is a shifted exponential distribution with shift $\mu = a$ and optimal scaling factor

$$\alpha^* = \frac{a + \sqrt{a^2 + 4}}{2}. \tag{4.48}$$

The "accept" condition for a sample $x \sim \text{Exp}(\alpha^*, a)$ here is two-fold: $x \leq b$, and $\rho(x) \triangleq \exp[-(x - \alpha^*)^2] \geq u \sim U(0, 1)$. When the support lies to the left of zero, the situation can be viewed as a mirror image of this problem. Also, Robert proves that it is better to instead use a uniform proposal distribution if the interval $[a, b]$ is sufficiently small; i.e. use the proposal distribution $U(x; a, b)$ if (for the case $0 < a < b$)

$$(b - a) \leq \frac{1}{\alpha^*} \exp\left[(a^2 - \alpha^* a + 1)/2\right]. \tag{4.49}$$

Once the samples χ_k are generated, the desired samples x_k can be recovered by undoing the linear transformation: $x_k = \sigma_k \chi_k + \mu_k$.

One might question why, since all the parameters $\mathbf{x}$ are highly correlated given the data $\mathbf{y}$, they are not sampled altogether to improve the rate of convergence. However, performing a linear transformation on the multivariate truncated normal distribution distorts the support of the distribution, so that the support is no longer a Cartesian product of intervals $[a_0, b_0] \times [a_1, b_1] \times \cdots \times [a_{K-1}, b_{K-1}]$. Thus, the multivariate distribution is difficult to sample from, and Gibbs sampling is recommended as an alternative in [26].

4.4 Slice Sampling to Approximate the BLS Estimator

Slice sampling is used to improve upon the Gibbs sampler developed in the previous section, as an inexpensive alternative to rejection sampling to create samples from the univariate distributions $p(z_n|\mathbf{y}, \mathbf{x})$. An alternative approach involving using slice sampling and Gauss-Hermite quadrature to sample from the full conditional distributions $p(x_k|\mathbf{y}, \mathbf{x}_{-k})$ yields another Gibbs sampling method that does not require samples of the jitter z_n. However, this approach is computationally very expensive, and since the quadrature yields only an approximation to the full conditional distribution, this other approach has a measurable probability of yielding a very inaccurate approximation to the BLS estimate.

In [25], Neal points out that performing Gibbs sampling using slice sampling to sample from each full conditional distribution allows slice sampling to be easily applied to complicated multivariate distributions. Using slice sampling on a multivariate distribution directly is possible; however, for each iteration, finding a reasonable approximation to the slice $\{\mathbf{x} : p(\mathbf{x}|\mathbf{y}) \geq a\}$, where a is the height of the slice (generated uniformly from the previous sample $\hat{\mathbf{x}}$), can be very difficult, and generally, the number of samples required to adequately represent a K-dimensional distribution is exponential in K.

The main application of slice sampling presented entails substituting slice sampling for the inefficient rejection sampling process used to generate samples from $p(\mathbf{z}|\mathbf{y}, \mathbf{x})$ with a slice sampler. Since $p(\mathbf{z}|\mathbf{y}, \mathbf{x})$ is separable, univariate slice sampling can be employed and generate samples of z_n in parallel for all n. In addition, the target density is proportional to $p(y_n|z_n, \mathbf{x})p(z_n)$, which has a closed form. Therefore, both finding the interval containing the slice and sampling from the portion of the interval that is part of the slice are expected to be relatively efficient. The below algorithm shows the adapted Gibbs sampler:

```
Require: $\mathbf{y}, I, I_b$
  $\mathbf{z}^{(0)} = \mathbf{0}$
  $\mathbf{x}^{(0)} = \mathbf{0}$
  for $i = 1 : I + I_b$ do
    for $n = 0 : N - 1$ do
      $u \sim U(0, p(\mathbf{y}, z_n^{(i-1)} | \mathbf{x}^{(i-1)}))$
      Use "shrinkage" method to obtain sample $z_n^{(i)}$ from slice.
    end for
    $\mathbf{x}^{(i)} \sim p(\cdot | \mathbf{z}^{(i)}, \mathbf{y})$
  end for
  $\hat{\mathbf{x}} = \frac{1}{I} \sum_{i=I_b+1}^{I_b+I} \mathbf{x}^{(i)}$
  $\hat{\mathbf{z}} = \frac{1}{I} \sum_{i=I_b+1}^{I_b+I} \mathbf{z}^{(i)}$
  return $\hat{\mathbf{x}}, \hat{\mathbf{z}}$
```

Since $u \leq p(\mathbf{y}, z_n | \mathbf{x}^{(i-1)})$, the range of possible z_n (and thus the extreme endpoints for search) is simply

$$|z_n| \leq \sigma_z \sqrt{-2 \log u - 2N \log(2\pi \sigma_w \sigma_z)} \tag{4.50}$$

Because the target distribution can be highly oscillatory within this interval, narrowing the interval to make sampling from the slice more efficient is not considered to preserve accuracy. Instead, the shrinkage method is applied to the entire bounded interval, which is not too large for u close to 1. This resulting algorithm can be faster than the original Gibbs sampler, and therefore, it can gain greater accuracy by approximating $\mathbb{E}[\mathbf{x}, \mathbf{z} | \mathbf{y}]$ with a greater number of iterations. However, the accuracy gained from additional iterations would have to be balanced by the fact that each iteration of slice sampling only produces an approximation to the target density, unlike rejection sampling. Therefore, the generated samples are less reliable, and the MSE for this algorithm can be expected to be greater in general than for the exact Gibbs sampler.

4.5 Simulations

As a baseline method, the LLS estmators for both the no-jitter (4.7) and random-jitter (4.4) cases are implemented, the latter using Gauss-Hermite quadrature to approximate $\mathbb{E}[\mathbf{H}(\mathbf{z})]$ and $\mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T]$. Also, approximations to the BLS estimate using a second-order Taylor series expansion, Gibbs sampling, and slice sampling are written in MATLAB.

Just as with the iterative ML estimators in Chapter 3, it is essential to verify that the stochastic approximations to the BLS estimate all converge in a reasonable time. All these Monte Carlo algorithms are guaranteed to converge to the true BLS estimate, but the rate of convergence depends on the distributions used. In addition, each algorithm's sensitivity to initial conditions needs to be evaluated. If the Markov chain of the Gibbs sampler is both irreducible and aperiodic, it should have exactly one stationary distribution, which will be the steady-state distribution regardless of the initial conditions [16]. However, if the chain is not irreducible, the chain may have many steady-state distributions, one for each recurrence class.

Once the convergence behavior and sensitivity to initial conditions have been studied, all of these BLS estimation algorithms are simulated against the optimal linear estimators for the random and zero-jitter cases. The goal of designing these algorithms is to tolerate the use of a less accurate sampling clock; the hypothesis is that by incorporating jitter into the observation model, algorithms can be constructed that take advantage of the more accurate model to yield better MSE performance.

4.5.1 Convergence of stochastic methods

The general convergence properties of a Gibbs sampler are well-known. The slice sampling variant results in a slightly more complicated Markov chain, but the algorithm should still converge. While the rate of convergence is approximately linear in the number of sampling iterations, simulations are needed to determine how many iterations are required to be sufficiently close to the true BLS estimate. In addition, the choice of initial conditions can influence the steady-state distribution, as well as the rate of convergence.

Both sampling-based approximations to the BLS estimator share similar convergence characteristics. The effects of changing the oversampling factor, jitter, and additive noise on the convergence rates are shown for each algorithm in Figures 4-1 and 4-2. The key message to get out of these plots is that convergence accelerates with increased oversampling rate or decreased additive noise or jitter variance.

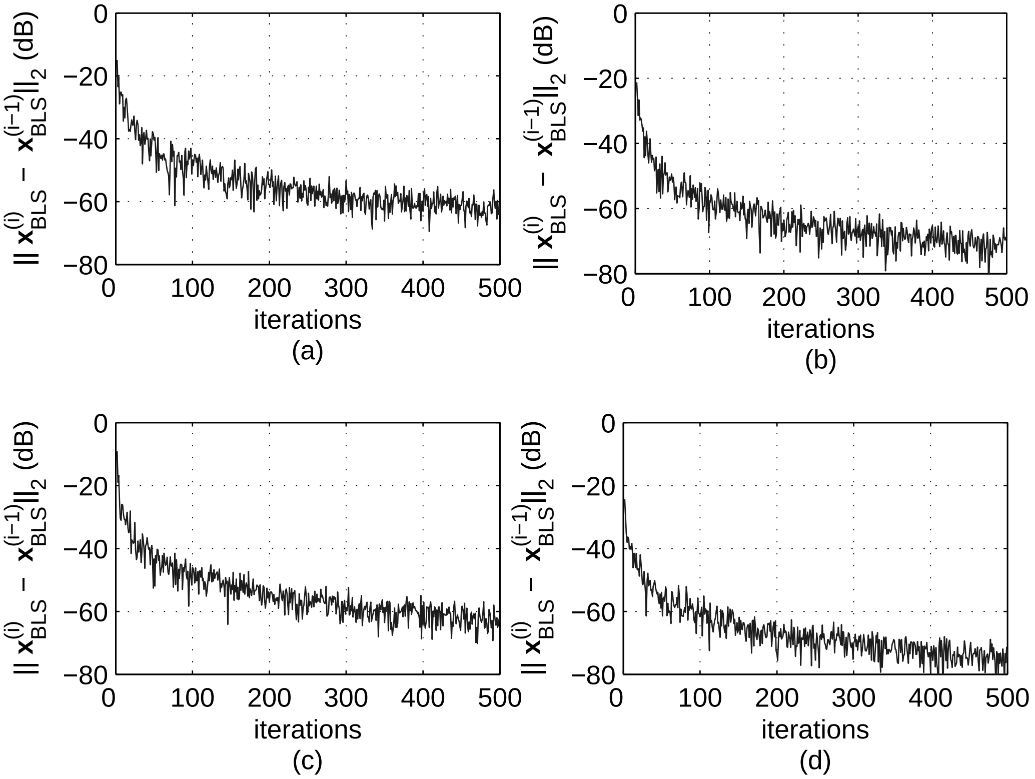


Figure 4-1: Gibbs sampler (with rejection sampling for z_n) convergence of BLS estimate over 500 iterations for $K = 5$, (a) $M = 2$, $\sigma_z = 0.1$, $\sigma_w = 0.25$, (b) $M = 16$, $\sigma_z = 0.1$, $\sigma_w = 0.25$, (c) $M = 2$, $\sigma_z = 0.001$, $\sigma_w = 0.25$, (d) $M = 2$, $\sigma_z = 0.1$, $\sigma_w = 0.05$.

However, these algorithms have a limited usefulness when the jitter variance is too large. The examples in Figures 4-3 and 4-4 demonstrate that when σ_w is much smaller than σ_z, or not enough oversampling is used, the algorithms may fail to converge to the same value, or may converge very slowly.

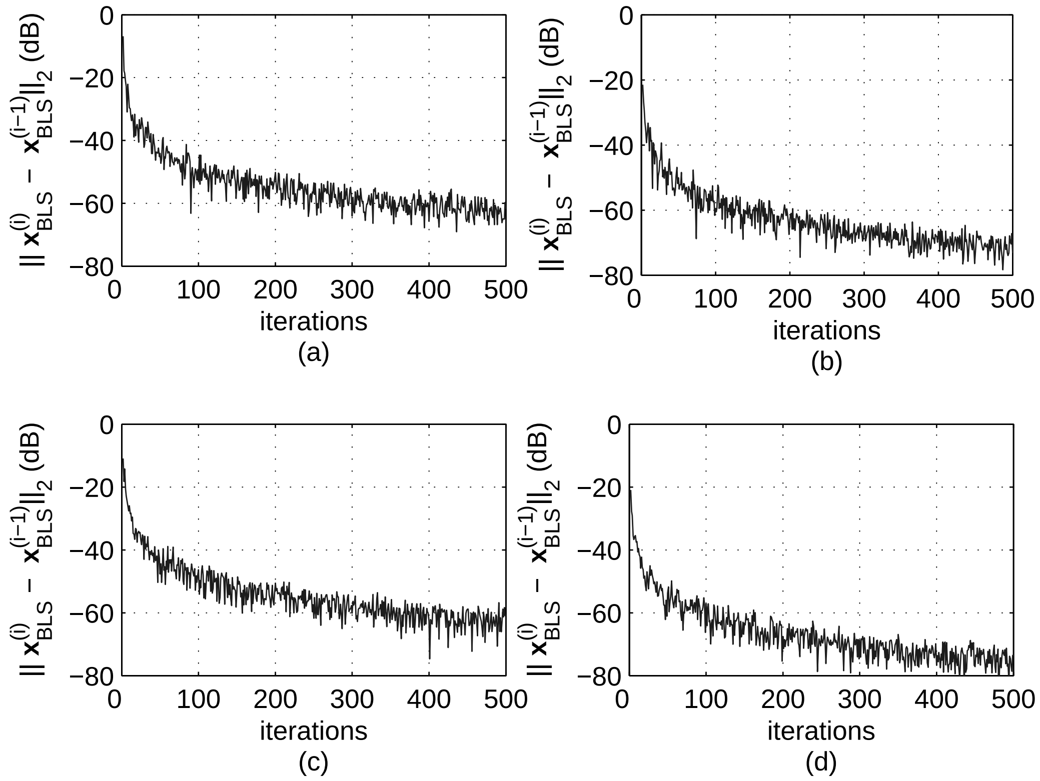


Figure 4-2: Gibbs sampler (with slice sampling for z_n) convergence of BLS estimate over 500 iterations for $K = 5$, (a) $M = 2$, $\sigma_z = 0.1$, $\sigma_w = 0.25$, (b) $M = 16$, $\sigma_z = 0.1$, $\sigma_w = 0.25$, (c) $M = 2$, $\sigma_z = 0.001$, $\sigma_w = 0.25$, (d) $M = 2$, $\sigma_z = 0.1$, $\sigma_w = 0.05$.

4.5.2 Linear and BLS methods compared

Having studied the convergence properties of the stochastic methods and analyzed their sensitivity to initial conditions, the performance of these algorithms now can be compared against the optimal linear estimate without jitter to confirm that accounting for the jitter more realistically in the observation model can yield better MSE performance.

The Taylor series approximation to the BLS estimate is connected to the error bound in (4.27). Essentially, to minimize the Taylor series approximation error, the additive noise variance should be maximized relative to σ_z. This constraint counterbalances the idea that the additive noise variance cannot be too great to risk the additive noise drowning out the effect of the jitter. To test the performance, the

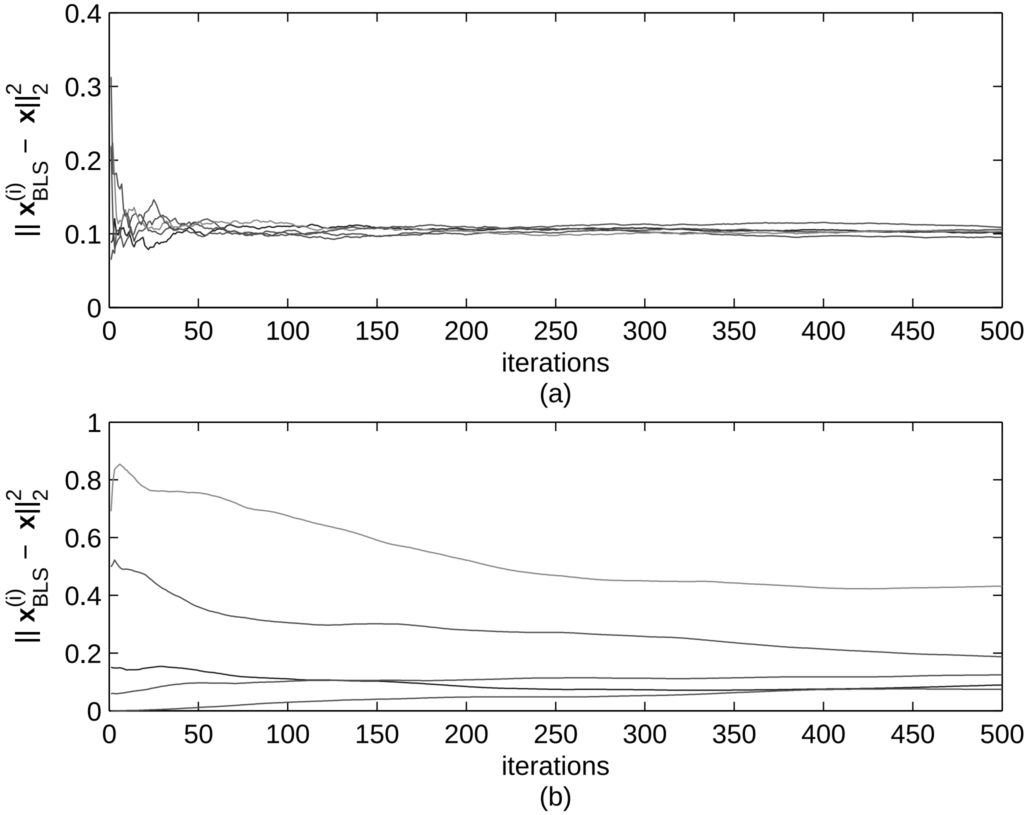
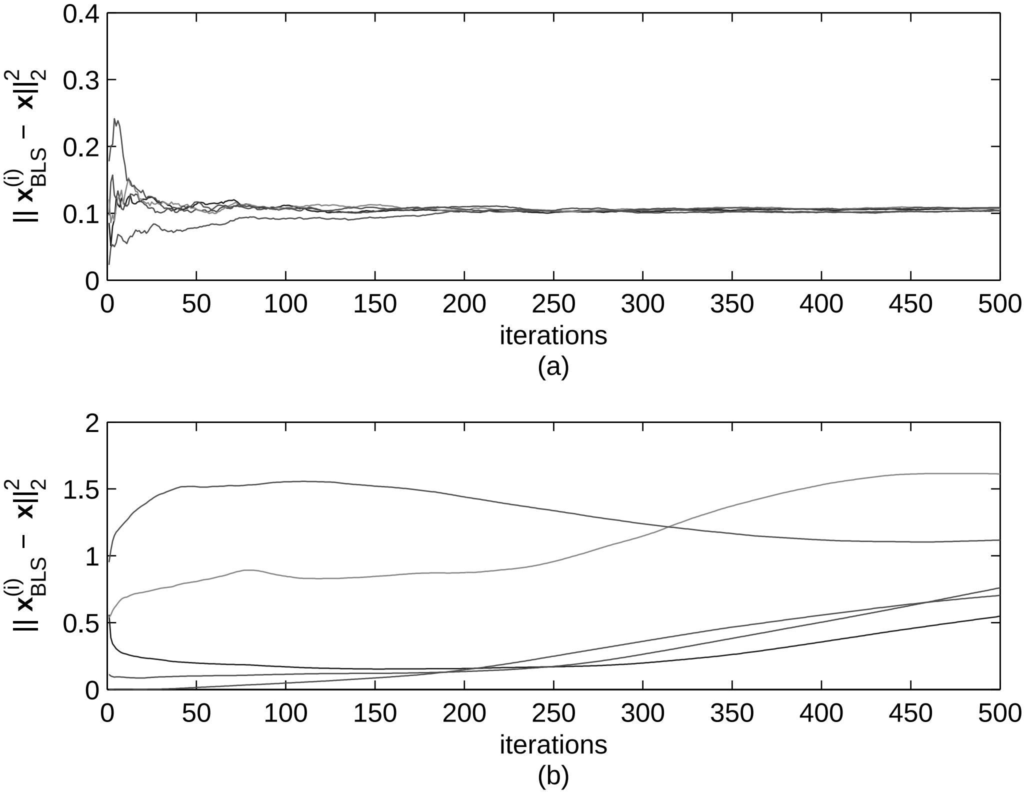


Figure 4-3: Gibbs sampler (with rejection sampling for z_n) sensitivity to initial conditions of BLS estimate over 500 iterations for $K = 5$: (a) $M = 4$, $\sigma_z = 0.1$, $\sigma_w = 0.25$ is relatively stable, while (b) $M = 2$, $\sigma_z = 0.5$, $\sigma_w = 0.01$ does not converge to the same value. The plots display the MSE (relative to the true value) for each iteration, starting the algorithms in five different ways: initializing $\mathbf{x}$ and $\mathbf{z}$ to zero; initializing $\mathbf{z}$ to zero and using the (no-jitter) LLS estimate for $\mathbf{x}$; initializing both $\mathbf{z}$ and $\mathbf{x}$ to the true value; and initializing $\mathbf{z}$ to a random value and using the fixed-jitter LLS estimate for $\mathbf{x}$, done for two different random values of $\mathbf{z}$.

jitter deviation is fixed to $\sigma_z = 0.001$ and M and σ_w are allowed to vary. The average improvement in MSE with $K = 10$ and 100 tests for every parameter combination is shown in Table 4.1.

Evidently, $\sigma_w = 0.5$ appears to be a good choice to get consistent, substantial improvement $(4 - 8\%)$. Table 4.2 shows the average MSE improvement for $\sigma_w = 0.5$ and varying σ_z.

For σ_z large enough relative to σ_w, the second order Taylor series expansion becomes a poor representation of the actual distribution, so the behavior, not unexpect-



(a)

(b)

Figure 4-4: Gibbs sampler (with slice sampling for z_n) sensitivity to initial conditions of BLS estimate over 500 iterations for $K = 5$: (a) $M = 4$, $\sigma_z = 0.1$, $\sigma_w = 0.25$ is relatively stable, while (b) $M = 2$, $\sigma_z = 0.5$, $\sigma_w = 0.01$ does not converge to the same value. The plots display the MSE (relative to the true value) for each iteration, starting the algorithms in five different ways: initializing $\mathbf{x}$ and $\mathbf{z}$ to zero; initializing $\mathbf{z}$ to zero and using the (no-jitter) LLS estimate for $\mathbf{x}$; initializing both $\mathbf{z}$ and $\mathbf{x}$ to the true value; and initializing $\mathbf{z}$ to a random value and using the fixed-jitter LLS estimate for $\mathbf{x}$, done for two different random values of $\mathbf{z}$.

edly, becomes rather chaotic. However, the performance improvement for smaller σ_z is encouraging.

In Figures 4-5, 4-6, 4-7, and 4-8, the Gibbs sampler is compared against the no-jitter linear least squares estimator to demonstrate the improvement that can be achieved using Gibbs sampling. Just as with the iterative estimators for the non-random case, the Gibbs sampler yields the greatest improvement over the linear estimator when the jitter is large relative to the additive noise, but not when the jitter is so large that the Gibbs sampler becomes an inaccurate approximation to the

Table 4.1: Average (%) MSE Improvement for approximate BLS vs. LLS estimation ($\sigma_z = 0.001$)

σ_w	$M = 2$	3	4	6	8	16
2.5	0.001	0.019	0.22	0.54	0.39	0.96
1	0.9	2.8	4.3	2.9	7.4	7.5
0.5	5.6	8.2	6.1	4.6	8.3	7.6
0.1	5.5	6.0	3.0	2.8	2.9	2.9
0.025	1.9	2.3	1.7	0.56	0.88	0.24

Table 4.2: Average (%) MSE Improvement for approximate BLS vs. LLS estimation ($\sigma_w = 0.5$)

σ_z	$M = 2$	3	4	6	8	16
0.001	5.6	8.2	6.1	4.6	8.3	7.6
0.01	5.6	8.2	6.1	4.6	8.3	7.6
0.1	5.8	7.6	5.9	4.2	7.9	5.6
0.5	0.49	-0.24	37.4	94.2	99.6	91.4

BLS estimate.

The Gibbs sampler using slice sampling for z_n has similar overall performance to the Gibbs sampler (compare to Figures 4-9, 4-10, 4-11, and 4-12). Notice the performance improvement given by using a slice sampler instead of rejection sampling in the case of high-oversampling (e.g. $M = 16$) and small additive noise variance. This performance boost can be traced to the difficulty of rejection sampling of z_n for these parameter choices. Essentially, when there is little additive noise to perturb the observations and high oversampling constrains the choice of signal parameters, the regions of z_n with high probability are very small and spaced far apart; therefore, rejection sampling can fail to find them, especially when the proposal distribution is a unimodal Normal distribution, and the target distribution is highly oscillatory, as in Figure 4-13.

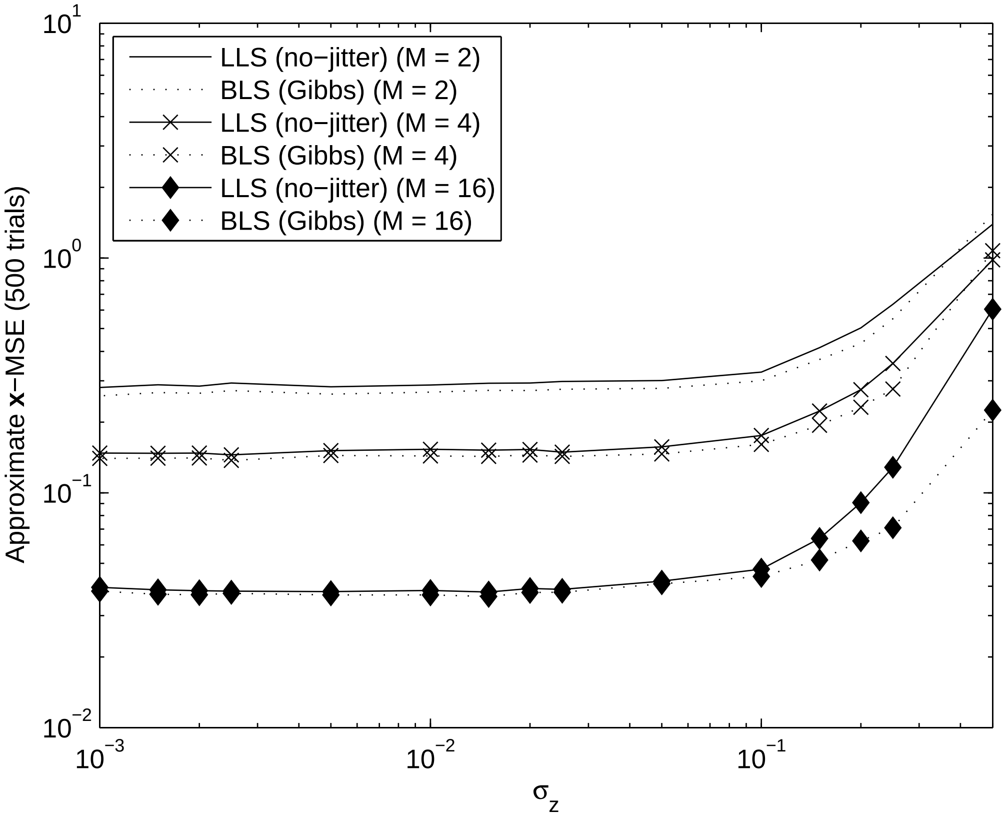


Figure 4-5: Gibbs sampler (with rejection sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.25$, and various choices of oversampling M and jitter σ_z.

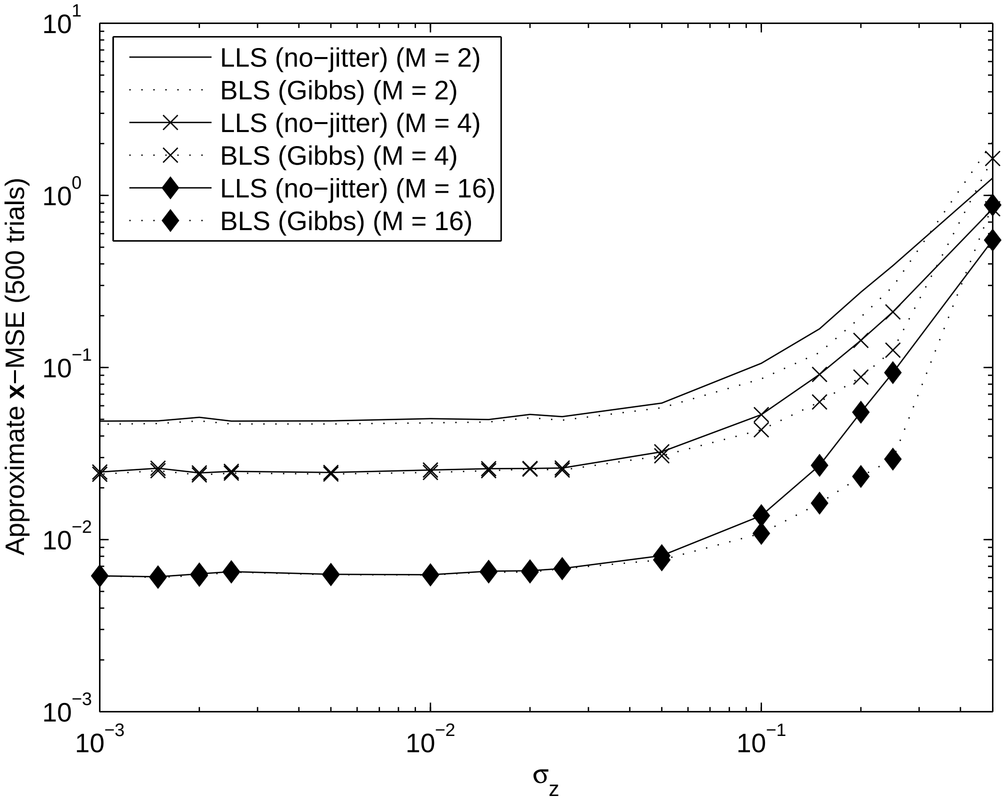


Figure 4-6: Gibbs sampler (with rejection sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.1$, and various choices of oversampling M and jitter σ_z.

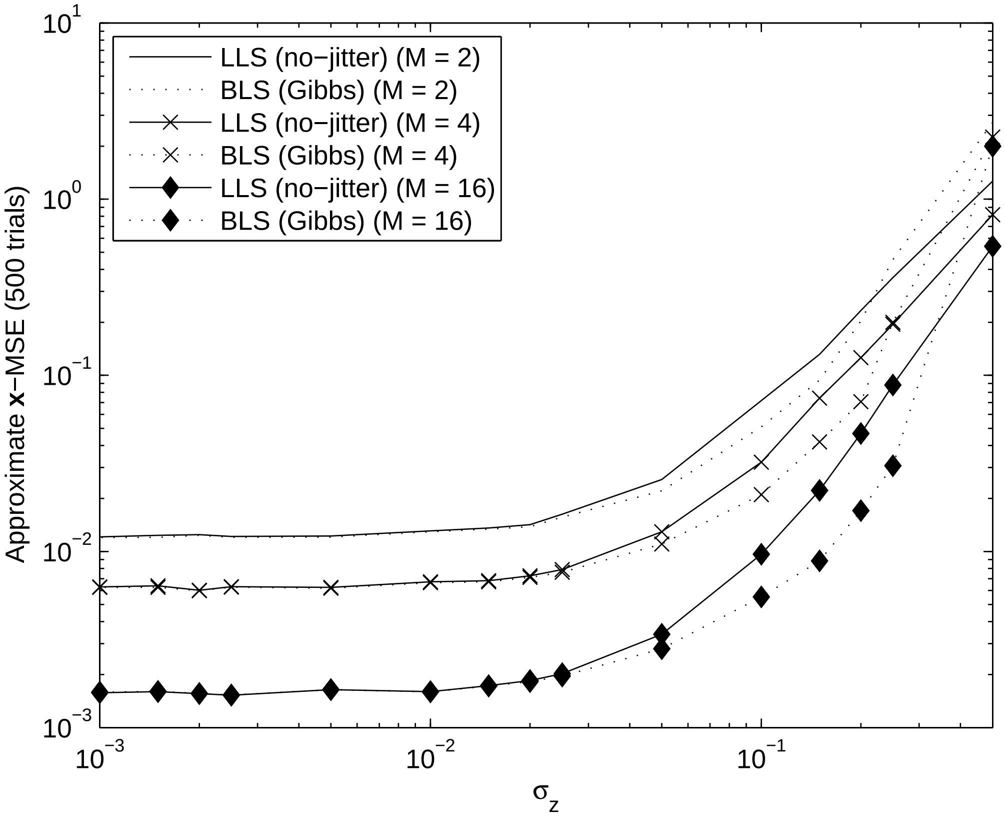


Figure 4-7: Gibbs sampler (with rejection sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.05$, and various choices of oversampling M and jitter σ_z.

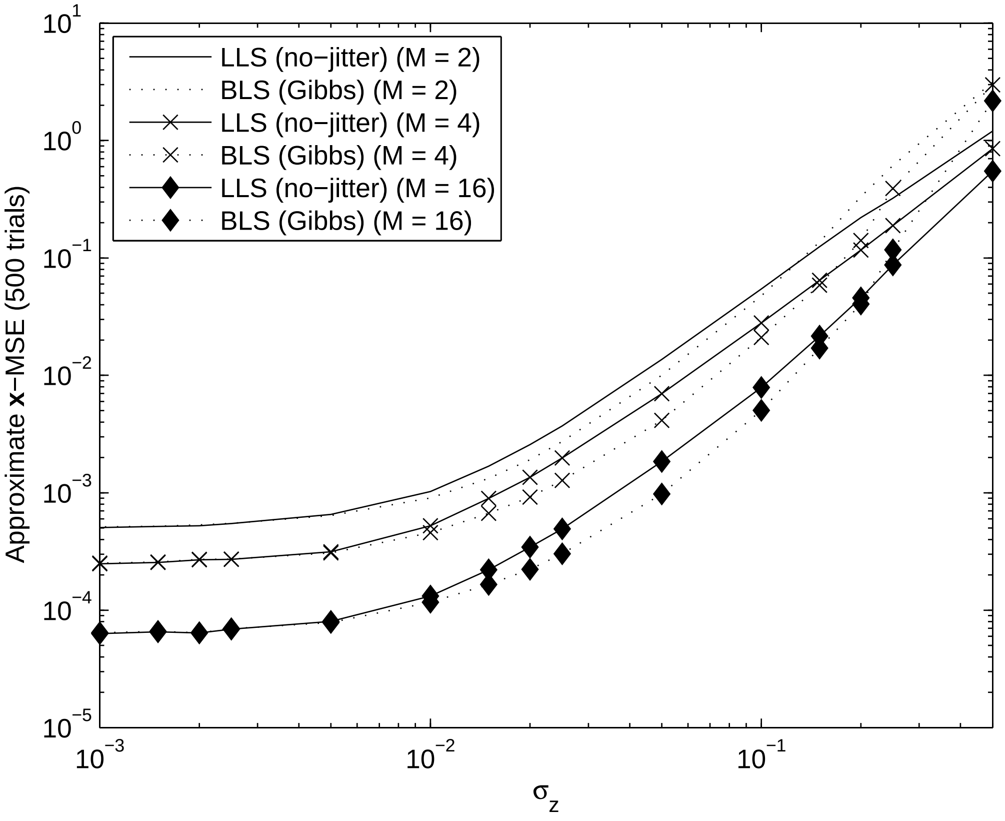


Figure 4-8: Gibbs sampler (with rejection sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.01$, and various choices of oversampling M and jitter σ_z.

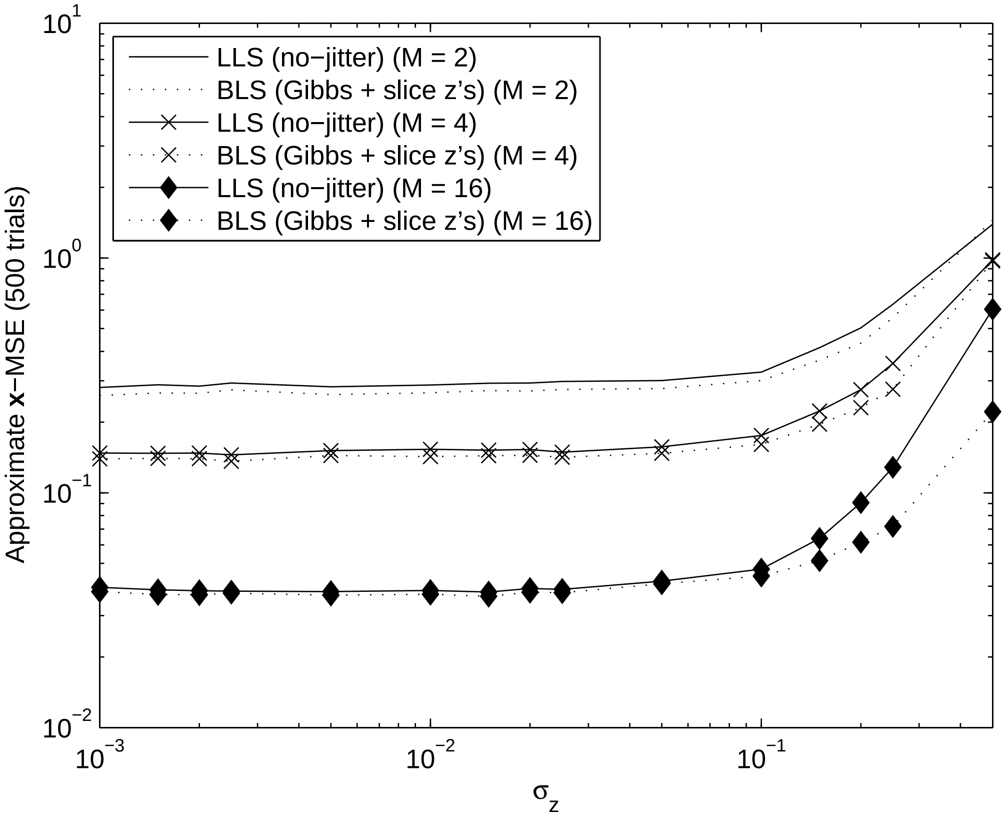


Figure 4-9: Gibbs sampler (with slice sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.25$, and various choices of oversampling M and jitter σ_z.

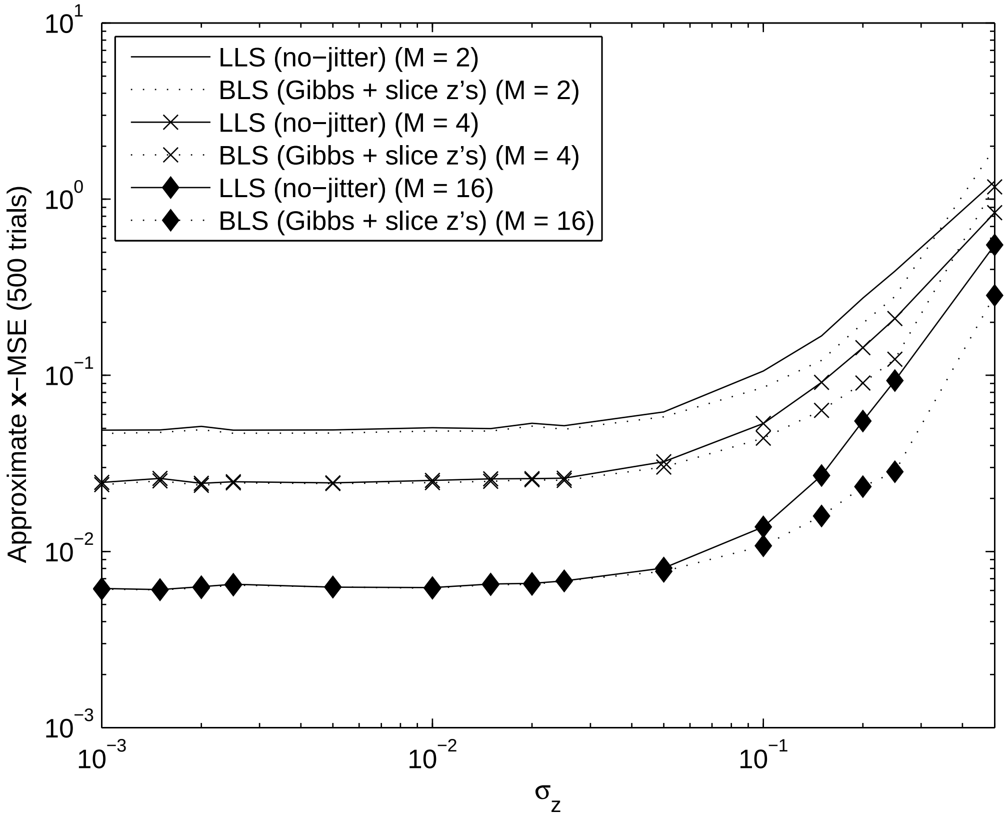


Figure 4-10: Gibbs sampler (with slice sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.1$, and various choices of oversampling M and jitter σ_z.

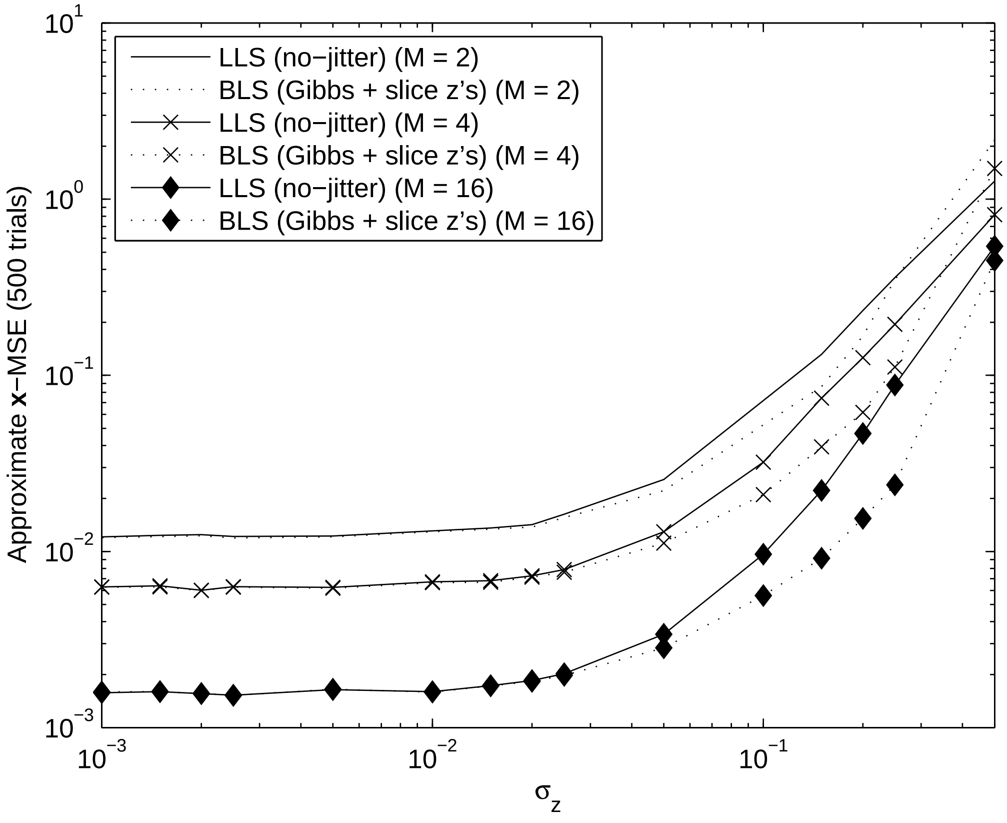


Figure 4-11: Gibbs sampler (with slice sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.05$, and various choices of oversampling M and jitter σ_z.

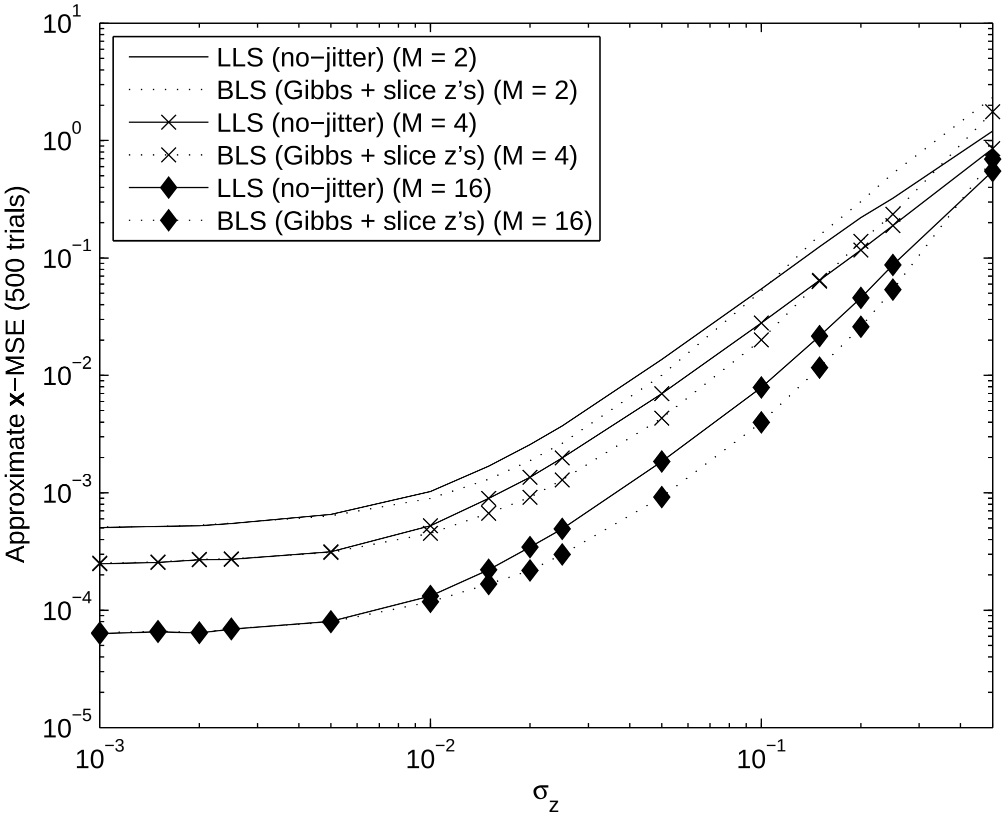


Figure 4-12: Gibbs sampler (with slice sampling for z_n) performance compared to no-jitter linear least squares estimate over 500 trials for $K = 10$, $\sigma_w = 0.01$, and various choices of oversampling M and jitter σ_z.

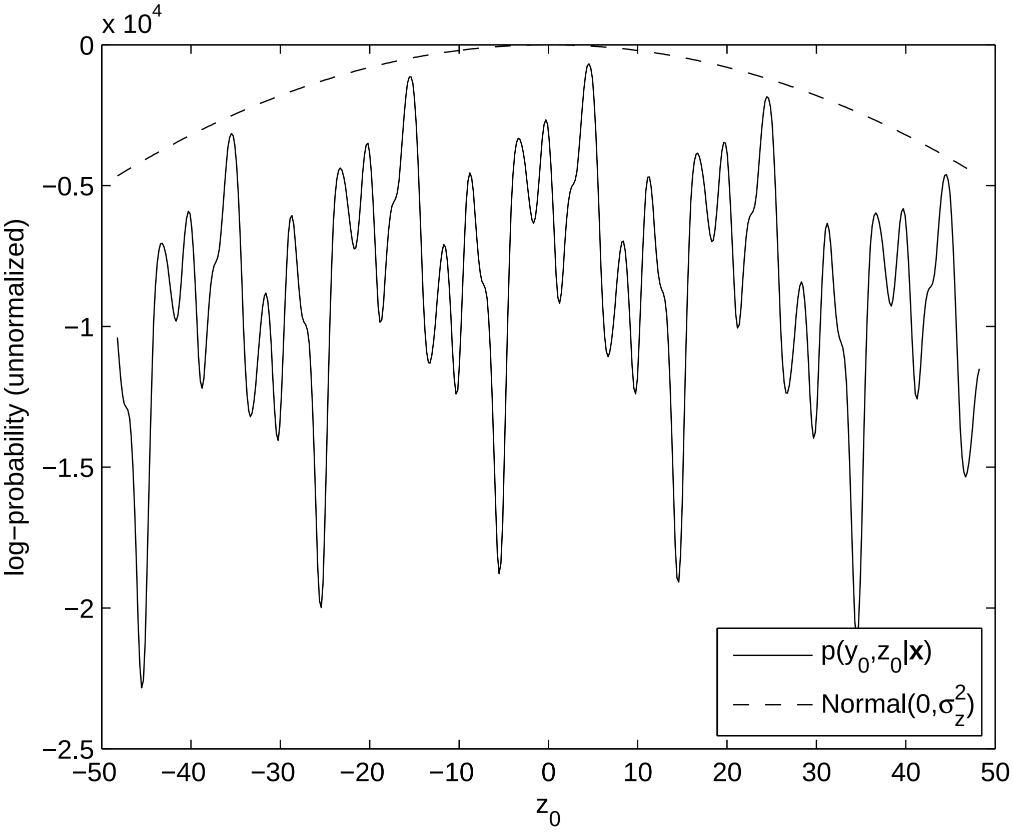


Figure 4-13: Example plot of the pdf $p(y_0, z_0|\mathbf{x})$ with respect to z_0 (holding y_0 fixed), for $K = 10$, $\sigma_w = 0.01$, $\sigma_z = 0.5$, and $M = 16$. This joint conditional pdf is used to generate samples of z_0, given the data y_0 and the previous estimate of the parameters $\mathbf{x}$, in the Gibbs sampler. The proposal distribution for rejection sampling (a zero-mean normal distribution with variance σ_z^2) is also shown for reference. Note that for z_0 close to zero, the joint conditional pdf has extremely low probability, so rejection sampling may reject an extraordinarily large number of samples before accepting a sample.

Chapter 5

Closing Remarks

The problem of estimating the parameters of a signal sampled in the presence of both jitter and additive noise is revisited, and new methods for achieving substantial improvement in MSE performance for both the non-random and Bayesian parameter cases are proposed. In particular, simulations confirm that the iterative EM algorithm is an attractive alternative to the linear estimation methods commonly employed today, and the Gibbs sampler and its variants show similar promise for the Bayesian case. These exciting results have numerous implications for the design of modern analog-to-digital converters.

As mentioned in the introduction, the growing market for ultra-low power electronics is stimulating the desire to make analog-to-digital converters more power-efficient. In addition, analog-to-digital converters have to be smaller and more inexpensive to be able to satisfy the demands of implantable medical instruments and other micro-devices. For instance, mobile wireless radio transceivers for cellular phones require extraordinary design constraints, where longer battery life, smaller size, and lower cost are primary design goals. In [5], Brannon cites how disruptive even 0.1 picoseconds of jitter can be to a GSM receiver/decoder. Thus, the ADCs currently used in common devices must incorporate clocks with extremely low phase noise, especially at high frequencies. At the Microsystem Technologies Laboratory at MIT, a single-chip ultra-wideband (UWB) system is proposed in [4], in which the flash ADC uses approximately a third of the chip area and over 86 mW of the chip's 271 mW power budget. Many other examples of the ADC's role in chip size and power consumption can be found throughout the electronics industry.

The iterative ML estimation and stochastic BLS estimation algorithms introduced in Chapter 3 and 4 can be used to reduce the need for low-noise clock signals in the above applications. By relaxing the stringent requirement on the clock jitter, designers employing these algorithms can use smaller analog-to-digital converters that consume less power, without sacrificing system-level performance. Since these post-processing algorithms do not require real-time access to the signal, the jitter mitigation process can be performed off-chip to minimize power consumption and complexity. As an alternative, the oversampling factor can be decreased instead of the jitter, thus decreasing the clock frequency, and hence, ADC power consumption.

Where to next? While I demonstrate that these algorithms yield considerable improvement over a wide range of jitter variance, these algorithms are by no means perfect. For very large jitter, all these algorithms become too inaccurate to be of much use. Developing methods to improve the accuracy of these algorithms would increase their useful range considerably, allowing even more jitter to be tolerated. In addition, these sampling methods are very inefficient computationally, since many samples must be generated and thrown away before an approximation for the BLS estimate even can be computed. Minimizing the so-called "burn-in" time of these algorithms would yield significant computational savings and perhaps improve the quality of the BLS estimate. Finally, more work is needed to extend the iterative EM algorithm and the Gibbs sampler for when the jitter is correlated or the signal parameters are generated from a non-trivial random process. However, care must be taken to avoid dramatically increasing the complexity of each iteration. These methods can all be modified for useful classes of non-bandlimited signals, such as splines.

Beyond analog-to-digital converters, jitter mitigation algorithms have a place in a variety of signal-processing systems. For instance, when applied to the frequency domain, timing noise translates into uncertainty in frequency; thus, jitter mitigation can yield a more accurate Fourier transform device for systems like spectrum analyzers. Also, mitigating jitter can reduce the potential for intersymbol interference in communication systems where symbols are packed closely together in time or frequency.

Mitigating clock jitter or timing/frequency noise is not the only direct application of this work. In the spatial domain, location uncertainty is analogous to jitter, and philosophically, such errors can be corrected using the same approach. Imagine a camera where the spacing of the pixels in the photosensor can be inconsistent. The cost of such components should fall dramatically if manufacturing tolerances can be relaxed. Location-jitter mitigation may even be combined with image de-mosaicing algorithms to better align the RGB color channels of the digital image.

The benefits of jitter mitigation are well worth the effort. Numerous iterative and stochastic algorithms for performing jitter mitigation have been presented and their performance simulated against simple linear algorithms. Several avenues for future exploration have been outlined, and the implications for analog-to-digital conversion, ultra-low power devices, and other areas just described. The author of this thesis hopes that this work will set the stage for the further exploration of jitter mitigation and the application of such algorithms in novel devices.

Appendix A

Derivation of the BLUE

Suppose the observation model is of the form in (1.9). Then, a linear unbiased estimator, as shown in (3.2), must satisfy

$$\mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbf{a}_i = \mathbf{d}_i, \quad \forall i, \tag{A.1}$$

where $\mathbf{a}_i^T$ is the ith row of $\mathbf{A}$ and $\mathbf{d}_i$ is a column vector whose only nonzero element is a one in the ith position.

The Best Linear Unbiased Estimator (BLUE) is the estimator that satisfies (A.1) and minimizes the MSE. Since the $\mathbf{a}_i^T$ rows can be chosen separately for each estimator, the estimator variance (which for unbiased estimators is also the error variance and MSE), can be minimized separately for each unknown parameter. The variance for the estimate of one such parameter is given by $\mathbf{a}_i^T \mathbf{\Lambda}_\mathbf{y} \mathbf{a}_i$, where $\mathbf{\Lambda}_\mathbf{y}$ is the covariance matrix of $\mathbf{y}$, which depends on $\mathbf{x}$. Being a covariance matrix, $\mathbf{\Lambda}_\mathbf{y}$ is both positive definite and symmetric (since the data is not deterministic, assume that the covariance matrix is not degenerate). Therefore, performing the constrained optimization problem

$$\min_{\mathbf{a}_i} \mathbf{a}_i^T \mathbf{\Lambda}_\mathbf{y} \mathbf{a}_i \tag{A.2}$$

subject to the ith constraint in (A.1) is equivalent to minimizing $\|\mathbf{w}_i\|_2^2$, where $\mathbf{w}_i = {\mathbf{\Lambda}_\mathbf{y}}^{1/2}\mathbf{a}_i$. This new minimization problem can be solved as in [8]: it has a particular solution $\breve{\mathbf{w}}_i$ found using the observation that since $\mathbf{W} \triangleq \mathbb{E}[\mathbf{H}(\mathbf{z})]^T {\mathbf{\Lambda}_\mathbf{y}}^{-1/2}$ has full row rank, $\mathbf{W}\mathbf{W}^T$ is invertible:

$$\breve{\mathbf{w}}_i = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}d_i. \tag{A.3}$$

This solution can be projected onto $\mathbf{W}$ and decomposed into its component in the range of $\mathbf{W}$ and the orthogonal component in the null-space of $\mathbf{W}$; denote these components $\mathbf{w}_W$ and $\mathbf{w}_{W\perp}$, respectively. Then, using the standard orthogonality argument,

$$\mathbf{w}_W = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{d}_i = \breve{\mathbf{w}}. \tag{A.4}$$

Since $\mathbf{w}^T\mathbf{w} = \mathbf{w}_W^T\mathbf{w}_W + \mathbf{w}_{W\perp}^T\mathbf{w}_{W\perp}$, selecting $\mathbf{w}$ to be in the range of $\mathbf{W}$ minimizes the 2-norm, with $\mathbf{w}_{W\perp} = \mathbf{0}$. Thus, plugging back into the expression for $\mathbf{a}_i$,

$$\mathbf{a}_i = \mathbf{\Lambda_y}^{-1}\mathbb{E}[\mathbf{H}(\mathbf{z})](\mathbb{E}[\mathbf{H}(\mathbf{z})]^T\mathbf{\Lambda_y}^{-1}\mathbb{E}[\mathbf{H}(\mathbf{z})])^{-1}\mathbf{d}_i. \tag{A.5}$$

Stacking the rows together yields the familiar expression for $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} -\mathbf{a}_1^T- \\ \vdots \\ -\mathbf{a}_K^T- \end{bmatrix} = (\mathbb{E}[\mathbf{H}(\mathbf{z})]^T\mathbf{\Lambda_y}^{-1}\mathbb{E}[\mathbf{H}(\mathbf{z})])^{-1}\mathbb{E}[\mathbf{H}(\mathbf{z})]^T\mathbf{\Lambda_y}^{-1}. \tag{A.6}$$

The resulting estimator is the Best Linear Unbiased estimate for $\mathbf{x}$. When $\mathbf{H}$ is a known matrix, this result simplifies to the one derived in [18]. However, Kay reaches this result differently, using Lagrange multipliers.

Bibliography

[1] A. V. Balakrishnan. On the problem of time jitter in sampling. *IRE Trans. Inform. Th.*, 8(3):226–236, April 1962.

[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Math. Statist. (Inst. Math. Stat.)*, 41(1):164–171, February 1970.

[3] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, U. C. Berkeley, International Computer Science Institute, Berkeley, California, April 1998.

[4] R. Blázquez, P. P. Newaskar, F. S. Lee, and A. P. Chandrakasan. A baseband processor for impulse ultra-wideband communications. *IEEE J. Solid-State Circuits*, 40(9):1821–1828, September 2005.

[5] B. Brannon. Aperture uncertainty and ADC system performance. Technical Report AN-501, Analog Devices, September 2000.

[6] W. M. Brown. Sampling with random jitter. *Journal of the Soc. Industrial and Appl. Math.*, 11(2):460–473, June 1963.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, Hoboken, New Jersey, second edition, 2006.

[8] M. Dahleh, M. A. Dahleh, and G. Verghese. Lectures on dynamic systems and control. Lecture notes for 6.241 Dynamic Systems and Control. Available at http://ocw.mit.edu.

[9] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, Orlando, 1984.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statist. Soc. Series B (Method.)*, 39(1):1–38, 1977.

[11] V. Divi and G. Wornell. Signal recovery in time-interleaved analog-to-digital converters. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 593–596, May 2004.

[12] R. Eckhardt. Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science (special issue)*, pages 131–137, 1987.

[13] A. E. Gelfand. Gibbs sampling. *Journal of the Amer. Statist. Assoc.*, 95(452):1300–1304, December 2000.

[14] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741, 1984.

[15] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Math. Comp. (Amer. Math. Soc.)*, 23(106):221–230, April 1969.

[16] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford, 2001.

[17] C. Herzet and L. Vandendorpe. Prediction of the EM-algorithm speed of convergence with Cramer-Rao bounds. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 805–808, April 2007.

[18] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, volume 1 of *Prentice-Hall Signal Processing Series*. Prentice Hall, Upper Saddle River, NJ, 1993.

[19] J. Kusuma. *Economical Sampling of Parametric Signals*. PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, August 2006.

[20] J. Kusuma and V. K. Goyal. Signal parameter estimation in the presence of timing noise. In *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, pages 984–989, March 2006.

[21] T. H. Lee and A. Hajimiri. Oscillator phase noise: A tutorial. *IEEE J. Solid-State Circuits*, 35(3):326–336, March 2000.

[22] Y. Lee and R. Mittra. Electromagnetic interference mitigation by using a spread-spectrum approach. *IEEE Trans. Electromagn. Compat.*, 44(2):380–385, May 2002.

[23] F. Lin and D. Y. Chen. Reduction of power supply EMI emission by switching frequency modulation. *IEEE Trans. Power Electron.*, 9(1):132–137, January 1994.

[24] B. Liu and T. P. Stanley. Error bounds for jittered sampling. *IEEE Trans. Autom. Control*, 10(4):449–454, October 1965.

[25] R. Neal. Slice sampling. *Annals of Statist. (Inst. Math. Stat.)*, 31(3):705–767, 2003.

[26] C. P. Robert. Simulation of truncated normal variables. *Statistics and Computing*, 5(2):121–125, June 1995.

[27] W. Rudin. *Principles of Mathematical Analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill, New York, third edition, 1976.

[28] A. F. M. Smith and G. O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statist. Soc. Series B (Method.)*, 55(1):3–23, 1993.

[29] J. C. Spall and S. D. Hill. Least-informative Bayesian prior distributions for finite samples based on information theory. In *Proceedings of the 28th IEEE Conference on Decision and Control*, volume 3, pages 2567–2569, December 1989.

[30] K. Uyttenhove and M. S. J. Steyaert. Speed-power-accuracy tradeoff in high-speed CMOS ADCs. *IEEE Trans. Circuits Syst. II*, 49(4):280–287, April 2002.

[31] R. H. Walden. Analog-to-digital converter survey and analysis. *IEEE J. Sel. Areas Commun.*, 17(4):539–550, April 1999.