Bayesian Post-Processing Methods for Jitter Mitigation in Sampling

The MIT Faculty has made this article openly available. *Please share* how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1109/tsp.2011.2108289">http://dx.doi.org/10.1109/tsp.2011.2108289</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td>Version</td>
<td>Author's final manuscript</td>
</tr>
<tr>
<td>Accessed</td>
<td>Mon Jan 07 14:41:36 EST 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/72601">http://hdl.handle.net/1721.1/72601</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Creative Commons Attribution-Noncommercial-Share Alike 3.0</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td><a href="http://creativecommons.org/licenses/by-nc-sa/3.0/">http://creativecommons.org/licenses/by-nc-sa/3.0/</a></td>
</tr>
</tbody>
</table>
Bayesian Post-Processing Methods for Jitter Mitigation in Sampling

Daniel S. Weller*, Student Member, IEEE, and Vivek K Goyal, Senior Member, IEEE

Abstract—Minimum mean squared error (MMSE) estimators of signals from samples corrupted by jitter (timing noise) and additive noise are nonlinear, even when the signal prior and additive noise have normal distributions. This paper develops a stochastic algorithm based on Gibbs sampling and slice sampling to approximate the optimal MMSE estimator in this Bayesian formulation. Simulations demonstrate that this nonlinear algorithm can improve significantly upon the linear MMSE estimator, as well as the EM algorithm approximation to the maximum likelihood (ML) estimator used in classical estimation. Effective off-chip post-processing to mitigate jitter enables greater jitter to be tolerated, potentially reducing on-chip ADC power consumption.

Index Terms—sampling, timing noise, jitter, analog-to-digital conversion, Markov chain Monte Carlo, Gibbs sampling, slice sampling

I. INTRODUCTION

Reducing the power consumption of analog-to-digital converters (ADCs) would improve the capabilities of power-constrained devices like medical implants, wireless sensors, and cellular phones. Clock circuits that produce jittered (noisy) sample times naturally consume less power than those with low phase noise, so allowing high phase noise is one avenue to reduce power consumption. However, increasing jitter in an ADC reduces the effective number of bits (ENOB) (rms accuracy on a dyadic scale) by one for every doubling of the jitter standard deviation, as described in [1] and [2]. Compensating for the reduced ENOB by designing more accurate comparators increases power consumption by a factor of four for every lost bit of accuracy [3]. Thus, to achieve reduced on-chip power consumption, the lost bits should be recovered in a different manner.

In [4], the authors post-process the jittered samples, employing an EM algorithm to perform classical maximum likelihood (ML) estimation of the signal parameters. This nonlinear classical estimation technique is capable of tolerating between 1.4 and 2 times the jitter standard deviation that can be mitigated by linear estimation. In this work, nonlinear post-processing is extended to the Bayesian framework, where the signal parameters are estimated knowing their prior distribution. Here, we do not require that signal and noise variances are known a priori; our hierarchical Bayesian model includes prior distributions on these parameters. The technique presented here achieves significant improvement over linear estimation for a wider range of jitter variance than the EM algorithm from [4], improving the applicability of nonlinear post-processing.

The block post-processing of the jittered samples is intended to be performed off-chip (e.g. on a PC), so we do not attempt to optimize the total power consumption, including the digital post-processing. However, we are concerned with making prudent choices in algorithm design so that the computational complexity of the post-processing is reasonable. The problem of mitigating jitter also can be motivated by loosening manufacturing tolerances (hence reducing cost) or by problems in which spatial locations of sensors are analogous to sampling times [5].

A. Problem Formulation

Consider the shift-invariant subspace of $L^2(\mathbb{R})$ associated with a generating function $h(t)$ and a signal $x(t)$ in that subspace:

$$x(t) = \sum_{k \in \mathbb{Z}} x_k h(t/T - k).$$

(1)

Assuming $\{h(t/T - k) : k \in \mathbb{Z}\}$ is a Riesz basis for the subspace, $x(t)$ is in one-to-one correspondence with the sequence $\{x_k\}_{k \in \mathbb{Z}}$ and the sequence $\{x_k\}_{k \in \mathbb{Z}}$ is in $\ell^2(\mathbb{Z})$. Examples of $h(t)$ include the function $\sin(t) \triangleq \sin(\pi t) / \pi t$ used throughout this paper, as well as B-splines and wavelet scaling functions as discussed in [6]. While $h(t) = \sin(t)$ is used for simulations, the developments in this paper are not specialized to the form of $h(t)$ in any way. We only require that $h(t)$ satisfies the Riesz basis condition and that the sampling prefiltter $s(-t)$ satisfies the biorthogonality condition $\{h(t/T - k), s(t/T - \ell)\} = \delta_{k-\ell}$, for all $k, \ell \in \mathbb{Z}$. The Riesz basis condition allows bounding of $L^2$ error of $x(t)$ in terms of $\ell^2$ error of $x_k$; when $\{h(t/T - k) : k \in \mathbb{Z}\}$ is an orthogonal set, these errors are constant multiples. When $h(t) = \sin(t)$, the shift-invariant subspace is the subspace of signals with Nyquist sampling period $T$. Without loss of generality, we assume $T = 1$.

When observing the signal $x(t)$ through a sampling system like an ADC, the analog signal is prefilttered by $s(-t)$, and samples $y_n$ are taken of the result at jittered times $t_n = nT_s + z_n$. To model oversampled ADCs, we oversample the signal by a factor of $M$, so the sampling period is $T_s = 1/M$. The samples are also corrupted by additive noise $w_n$, which models auxiliary effects like quantization and thermal noise. For $h(t) = \sin(t)$, the dual $s(t) = \sin(t)$ is an ideal lowpass...
filter with bandwidth \(2\pi\). The observation model, depicted in Figure 1, is

\[
y_n = \left[ x(t) * s(-t) \right]_{t=\Delta t} + z_n + w_n. \tag{2}
\]

We aim to estimate a block of \(K\) coefficients, assuming the remaining coefficients are negligible:

\[
x(t) \approx \sum_{k=0}^{K-1} x_k h(t-k). \tag{3}
\]

This specializes the observation model to

\[
y_n = \sum_{k=0}^{K-1} x_k h \left( \frac{n}{M} + z_n - k \right) + w_n, \tag{4}
\]

Grouping the variables into vectors, let \(\mathbf{x} = [x_0, \ldots, x_{K-1}]^T\), \(\mathbf{y} = [y_0, \ldots, y_{N-1}]^T\), \(\mathbf{z} = [z_0, \ldots, z_{N-1}]^T\), and \(\mathbf{w} = [w_0, \ldots, w_{N-1}]^T\). Then, in matrix form,

\[
y = \mathbf{H}(\mathbf{x}) \mathbf{x} + \mathbf{w}, \tag{5}
\]

where \(\mathbf{H}(\mathbf{z})_{n,k} = h(\frac{n}{M} + z_n - k),\) for \(n = 0, \ldots, N-1\), and \(k = 0, \ldots, K-1\). Let \(h^T_n(z_n)\) be the \(n\)th row of \(\mathbf{H}(\mathbf{z})\). Also, denote the \(k\)th column of \(\mathbf{H}(\mathbf{z})\) by \(\mathbf{H}_k(\mathbf{z})\) and the matrix with the remaining \(K-1\) columns by \(\mathbf{H}_{\setminus k}(\mathbf{z})\). Similarly, let \(\mathbf{x}_{\setminus k} = [x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{K-1}]^T\) be the vector of all but the \(k\)th signal coefficient.

In this paper, we assume both the jitter and additive noise are random, independent of each other and the signal \(x(t)\). Specifically, \(z_n\) and \(w_n\) are assumed to be iid zero-mean Gaussian, with variances equal to \(\sigma_z^2\) and \(\sigma_w^2\), respectively. In keeping with the Bayesian framework, we also choose a prior for the signal parameters. For convenience, we use an iid zero-mean Gaussian prior with variance \(\sigma_z^2\) because the observation model is linear in the parameters. Rather than assuming these parameters (variances) to be known, we treat them as random variables and assign a conjugate prior to these parameters. Thus, \(\sigma_z^2, \sigma_w^2,\) and \(\sigma_x^2\) are inverse Gamma distributed with hyperparameters \(\{\alpha_x, \beta_x\}, \{\alpha_w, \beta_w\},\) and \(\{\alpha_z, \beta_z\}\), respectively. These hyperparameters may be selected to be consistent with in-factory measurement of the noise variances or other information. The hierarchical Bayesian model is shown in Figure 2.

To simplify notation, the probability density function (pdf) of \(a\) is written as \(p(a)\), and the pdf of \(b\) conditioned on \(a\) is abbreviated as \(p(b \mid a)\) for random \(a\) and \(p(b; a)\) for nonrandom \(a\). The subscripts usually included outside the parentheses will be written only when needed to avoid confusion. Expectations will follow the same convention.

The uniform distribution is written in this paper as \(U(\set)\); for instance, \(U([a,b])\) is a uniform distribution over the interval \([a,b]\), and \(U(\{u: p(u) \geq c\})\) is a uniform distribution over the set \(\{u: p(u) \geq c\}\). Writing \(u \sim U(\set)\) means that \(u\) is a sample generated from this distribution; analogous notation is used for the other distributions in this paper. The inverse Gamma distribution has the density function

\[
I\Gamma(s; \alpha, \beta) \propto \beta^\alpha \Gamma(\alpha)^{-1} s^{\alpha-1} e^{-\beta/s}. \tag{6}
\]

The mean and variance of \(s\) are

\[
E[s] = \frac{\beta}{\alpha - 1}, \quad \text{and} \quad \text{var}(s) = \frac{\beta^2}{(\alpha - 1)^2(\alpha - 2)}. \tag{7}
\]

The density function of the \(d\)-dimensional normal distribution with mean \(\mu\) and covariance matrix \(\Lambda\) is written as

\[
\mathcal{N}(\mathbf{a}; \mu, \Lambda) \overset{\Delta}{=} \left|2\pi\Lambda\right|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{a} - \mu)^\top \Lambda^{-1} (\mathbf{a} - \mu)\right\}. \tag{8}
\]

When performing simulations, specific values are required for the \(\alpha's\) and \(\beta's\). For the signal variance \(\sigma_x^2\), consider an unbiased estimate of that variance from \(K > 1\) observations generated from a standard normal distribution: \(s_K = \frac{1}{K-1} \sum_{k=0}^{K-1} (x_k - \bar{x})^2\), where \(\bar{x} = \frac{1}{K} \sum_{k=0}^{K-1} x_k\) is the sample mean. Then, we fit the inverse Gamma prior hyperparameters \(\alpha_x\) and \(\beta_x\) to the mean and variance of \(s_K\) using (7).

\[
\frac{\beta_x}{\alpha_x - 1} = E[s_K] = 1; \quad \frac{\beta_x^2}{(\alpha_x - 1)^2(\alpha_x - 2)} = \text{var}(s_K) = \frac{2}{K - 1}. \tag{9}
\]

Solving,

\[
\alpha_x = \frac{K + 3}{2}; \quad \beta_x = \frac{K + 1}{2}. \tag{10}
\]
Similarly for the zero-mean jitter and additive noise variances, given \( N > 1 \) observations and expected noise variances \( \mathbb{E}[\sigma_z^2] \) and \( \mathbb{E}[\sigma_w^2] \),

\[
\alpha_z = \alpha_w = \frac{N + 3}{2}, \quad \beta_z = \frac{N + 1}{2} \mathbb{E}[\sigma_z^2], \quad \text{and} \quad \beta_w = \frac{N + 1}{2} \mathbb{E}[\sigma_w^2].
\]

For the examples in this paper, we use the same \( K \) and \( N \) as for our signal; in practical applications, \( K \) and \( N \) are prior observations performed at a factory (for the noise variances) or elsewhere (for the signal variance).

The objective of the algorithm presented in this paper is to find the estimator \( \hat{x} \) that minimizes the mean squared error (MSE) \( \mathbb{E}[\|\hat{x}(y) - x\|^2] \), where the observations \( y \) are implicitly functions of \( x \). Unlike in the classical estimation framework, we have a prior on \( x \), which allows us to formulate the minimum mean squared error (MMSE) estimator as the posterior expectation

\[
\hat{x}_{\text{MMSE}} \triangleq \mathbb{E}[x \mid y].
\]

The posterior distribution \( p(x \mid y) \) depends on the likelihood function \( p(y \mid x) \), which can be expressed as in [4] as a product of marginal likelihoods:

\[
p(y_n \mid x) = \int \int N(y_n; h_n^T(z_n)x, \sigma_w^2)N(z_n; 0, \sigma_z^2)I\mathcal{G}(\sigma_z^2; \alpha_z, \beta_z)I\mathcal{G}(\sigma_w^2; \alpha_w, \beta_w)\]

As neither the likelihood nor posterior distribution has a simple closed form, the majority of this paper is devoted to approximating these functions using numerical and stochastic methods.

### B. Related Work

Random jitter has been studied extensively throughout the early signal processing literature (see [7], [8], and [9]). However, much of the effort in designing reconstruction algorithms was constrained to linear transformations of the observations. These papers also analyze the performance of such algorithms; for example, [9] proves that when the jitter is Gaussian and small enough, the MSE is approximately \( \frac{1}{N^2} \Omega_0^2 \sigma_z^2 \), where the input PSD \( S_{xx}(j\Omega) = \frac{1}{\Omega_0^2} \) is flat. Due to the lack of attention to nonlinear post-processing, it is not readily apparent from the literature that these linear estimators are far from optimal. The effects of jitter on linear MMSE reconstruction of bandlimited signals are discussed in [10] and extended to the asymptotic \( K, N \to \infty \) case and multidimensional signals in [11].

More recently, [12] uses a second-order Taylor series approximation to perform weighted least-squares fitting of a jittered random signal. In [13], two post-processing methods are described for the case when the sample times are discrete (on a dense grid). Similar to the Gibbs sampler presented in this work, [14] uses a Metropolis-Hastings Markov chain Monte Carlo (MCMC) algorithm to estimate the jitter and jitter variance from a sequence of samples. Also, a maximum a posteriori (MAP)-based estimator is proposed in [15] to mitigate read-in and write-out jitter in data storage devices. Finally, a Gibbs sampler is developed in [16] to estimate the coefficients and locations of finite rate of innovation signals from noisy samples.

Preliminary versions of the algorithms and results presented in this work are also discussed, with further background material and references, in [17].

### II. BACKGROUND

In general, the likelihood function in the introduction is described in terms of an integration without a closed form. Fortunately, numerical methods such as Gauss quadrature, which approximates the integration in question with a weighted sum of the integrand evaluated at different locations (abscissas), are relatively accurate and efficient. A more detailed description of Gauss quadrature can be found in the background section of [4], or in [18] or [19]. This paper discusses using Gauss–Laguerre quadrature to approximate integration with respect to \( \sigma_z^2 \) and \( \sigma_w^2 \).

However, simply being able to evaluate (approximately) the likelihood function is insufficient to approximate the Bayes MMSE estimator. To approximate the expectation in (12), we propose using a Monte Carlo statistical method combining Gibbs sampling and slice sampling. Gibbs sampling and slice sampling are discussed below.

#### A. Numerical Integration

For integrals of the form \( \int_{-\infty}^{\infty} f(x)N(x; \mu, \sigma^2) \, dx \), techniques such as Gauss–Legendre and Gauss–Hermite quadrature, Romberg’s method, and Simpson’s rule, are described in [4]. Similarly, Gauss–Laguerre quadrature can approximate integrals of the form \( \int_{0}^{\infty} f(x) x^\alpha e^{-x} \, dx \). The abscissas and weights for Gauss–Laguerre quadrature can be computed using the eigenvalue-based method derived in [20].

Let \( x_j \) and \( w_j \) be the abscissas and weights for the Gauss–Laguerre quadrature rule of length \( J \). Then, we can integrate against the pdf of the inverse Gamma distribution by observing

\[
\int_{0}^{\infty} f(x) I\mathcal{G}(x; \alpha, \beta) \, dx = \int_{0}^{\infty} \frac{\beta^\alpha}{\Gamma(\alpha)} f(x) x^{-(\alpha+1)} e^{-\beta/x} \, dx
\]

\[
= \int_{0}^{\infty} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{e^{-\beta/x}}{x} \frac{e^{-x}}{y} \frac{1}{\beta} \, dy
\]

\[
= \int_{0}^{\infty} \frac{1}{\Gamma(\alpha)} \left( \frac{\beta}{y} \right)^{\alpha+1} e^{-\beta/y} \, dy
\]

\[
= \sum_{j=1}^{J} w_j f(x_j),
\]

\[(14)\]
Gauss–Laguerre quadrature with $\sigma^2_{\text{Lag}}$ are shown in Figure 3. Based on these comparisons, we choose $\sigma$ using (4). Comparisons for unimodal and multimodal $x$ empirically, by fixing $y$ as a function of $x$.

To explore the accuracy of this approximation the choices of $\sigma$ on the value of $\sigma^2_{\text{Lag}}$. In this equation, the innermost quadrature (over $\alpha$’s and $\beta$’s) are computed from samples of $\sigma^2_{\text{Lag}}, \sigma^2_{\text{Herm}}$, and $w_n$ according to (4). The multimodal case (a) favors Gauss–Legendre quadrature; case (b) favors Gauss–Hermite quadrature; the worst-case for (4). Utilizing a combination of Gauss–Laguerre quadrature and Gauss–Hermite quadrature with $J_3 = 129$ when $\mathbb{E}[\sigma^2_{\text{Lag}}]$ is small ($< 0.01$) and Gauss–Legendre quadrature with $J_3 = 129$ when $\mathbb{E}[\sigma^2_{\text{Herm}}] > 0.01$. This hybrid quadrature also is used when computing the expectations in Section III and in the appendix.

B. Gibbs Sampling

The Gibbs sampler is a Markov chain Monte Carlo method developed in [21]. Details about the Gibbs sampler and its many variants, including Metropolis-within-Gibbs sampling, can be found in [22]. When implementing the Gibbs sampler, one must consider both the number of iterations until the Markov chain has approximately converged to its stationary distribution (the “burn-in time”) and the number of samples that should be taken after convergence to compute the MMSE estimate. According to [23], separating highly correlated variables slows convergence of the Gibbs sampler. The number of iterations after convergence is connected to both correlation between successive samples and the variance of the random variables distributed according to the stationary distribution.

To monitor convergence, heuristics such as the potential scale reduction factor (PSRF) and the inter-chain and intra-chain variances are developed in [24], [25]. Consider $C$ instances (chains) of the Gibbs sampler running simultaneously. Define the vector $\mathbf{a}_{c,i}$ to be the combined vector of all the samples for the $c$th chain at the $i$th iteration. For chain $c$, the average is $\bar{\mathbf{a}}_c = \frac{1}{i} \sum_{j=1}^{i} \mathbf{a}_{c,i}$. Across all chains, the average is $\bar{\mathbf{a}} = \frac{1}{C} \sum_{c=1}^{C} \bar{\mathbf{a}}_c$. Then, following the multivariate extension to the potential scale reduction factor (PSRF) derived in [25], define the intra-chain covariance

$$W_i \triangleq \frac{1}{(i-1)C} \sum_{c=1}^{C} \sum_{j=1}^{i} (\mathbf{a}_{c,i} - \bar{\mathbf{a}}_c)(\mathbf{a}_{c,i} - \bar{\mathbf{a}}_c)^T,$$

and the inter-chain covariance

$$B_i \triangleq \frac{1}{i-1} \sum_{j=1}^{i} (\bar{\mathbf{a}}_c - \bar{\mathbf{a}})(\bar{\mathbf{a}}_c - \bar{\mathbf{a}})^T.$$

The posterior variance $\hat{\mathbf{V}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{V}_i + C^{-1}B_i)$, and the PSRF $R^p \triangleq \frac{1}{C+1} + \frac{C+1}{C+2} \left\| \mathbf{V}_i^{-1} \mathbf{B}_i \right\|_2$, where $\| \cdot \|_2$ is the induced matrix 2-norm. Then, the Gibbs sampler’s Markov chain has converged when $R^p = 1$, and $\hat{\mathbf{V}}$ stabilizes. To measure the change in $\mathbf{V}$, we compute $\| \mathbf{V} \|_2^{-1}$.

C. Slice Sampling

Slice sampling is a Markov chain Monte Carlo method described in [26] for generating samples from a distribution by instead sampling uniformly from the subgraph of the pdf and framing this sampling procedure as a two-stage Gibbs sampler, depicted in Figure 4.

The difficulty of slice sampling is in representing and sampling from the slice. In this problem, we show that any given slice is bounded, and therefore, an interval containing the slice can be constructed, and the “shrinkage” method described in [26] can be used. The shrinkage method is an accept-reject method, where given an interval $[L, R]$ containing part (or all)
of the slice, a sample is generated uniformly from the interval and accepted if the sample is inside the slice. If the sample is rejected, the interval shrinks to use the rejected sample as a new endpoint. Several variants, including shrinking to the midpoint of the interval instead of or in addition to the rejected sample, are also described in the rejoinder at the end of [20]. These variants are compared in the context of the jitter mitigation problem in Section [IV].

III. LINEAR BAYESIAN ESTIMATION

When block post-processing the samples, the linear Bayesian estimator with minimum MSE is called the linear MMSE (abbreviated LMMSE) estimator. The general form of the linear MMSE estimator is given in [27]. For estimating the random signal coefficients \( \mathbf{x} \) using the hierarchical Bayesian model in Section [IV],

\[
\mathbf{A}_{xy} = \frac{\beta_x}{\alpha_x - 1} \mathbb{E}[\mathbf{H}(\mathbf{z})]^T, \tag{18}
\]

\[
\mathbf{A}_y = \frac{\beta_x}{\alpha_x - 1} \mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\beta_w}{\alpha_w - 1} \mathbf{I}, \tag{19}
\]

and \( \boldsymbol{\mu}_y = \boldsymbol{\mu}_x = \mathbf{0} \). The LMMSE estimator for random jitter is

\[
\hat{x}_{\text{LMMSE}}(y) = \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \left( \mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\beta_w}{\alpha_w - 1} \mathbf{I} \right)^{-1} \mathbf{y}. \tag{20}
\]

The expectations in (20) can be computed off-line using Gauss quadrature. The error covariance of the LMMSE estimator is also derived in [27]; for this problem,

\[
\mathbf{A}_{\text{LMMSE}} = \frac{\beta_x}{\alpha_x - 1} \left( \mathbf{I} - \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \left( \mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\beta_w}{\alpha_w - 1} \mathbf{I} \right)^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})] \right)^T \mathbb{E}[\mathbf{H}(\mathbf{z})\mathbf{H}(\mathbf{z})^T] + \frac{\beta_w}{\alpha_w - 1} \mathbf{I} \right)^{-1} \mathbf{y}. \tag{21}
\]

When no jitter is assumed, the LMMSE estimator simplifies to

\[
\hat{x}_{\text{LMMSE}|\mathbf{z}=\mathbf{0}}(y) = \mathbf{H}(0)^T \left( \mathbf{H}(0)\mathbf{H}(0)^T + \frac{\beta_w}{\alpha_w - 1} \mathbf{I} \right)^{-1} \mathbf{y}. \tag{22}
\]

This linear estimator is the best linear transformation of the data that can be performed in the absence of jitter. Hence, the no-jitter LMMSE estimator is the baseline estimator against which the nonlinear Bayesian estimators derived later are measured. The error covariance of this estimator is

\[
\mathbf{A}_{\text{LMMSE}|\mathbf{z}=\mathbf{0}} = \sigma_x^2 \left( \mathbf{I} - \mathbf{H}(0)^T \left( \mathbf{H}(0)\mathbf{H}(0)^T + \frac{\beta_w}{\alpha_w - 1} \mathbf{I} \right)^{-1} \mathbf{H}(0) \right)^{-1} \tag{23}
\]

IV. NONLINEAR BAYESIAN ESTIMATION

To improve upon the LMMSE estimator, we expand our consideration to nonlinear functions of the data. The Bayes MMSE estimator, in its general form in (12), is the nonlinear function that minimizes the MSE. However, since the posterior density function for this problem does not have a closed form, this estimator can be difficult to compute. Since we are interested in the mean of the posterior pdf, finding the Bayes MMSE estimator is an obvious application of Monte Carlo statistical methods, especially the Gibbs sampler described in Section [III].

We propose using Gibbs sampling to produce a sequence of samples for the random parameters we wish to find, via traversing a Markov chain to its steady-state distribution, and average the samples to approximate the estimator. To this end, samples of \( \mathbf{z}, \mathbf{x}, \sigma_x^2, \sigma_w^2, \) and \( \zeta^2 \) are generated according to their full conditional distributions (i.e. the distribution of one random variable given all the others). To generate samples of \( \mathbf{z}, \) we apply slice sampling.

A. Generating \( \mathbf{z}_n \) using Slice Sampling

Consider generating samples \( \mathbf{z}_n \) from the distribution \( p(\cdot | \mathbf{z}_{\text{\textasciitilde}n}, \mathbf{x}, \sigma_x^2, \sigma_w^2, \alpha_x^2, \sigma_w^2, \zeta^2) \), where \( \mathbf{z}_{\text{\textasciitilde}n} \) is the random vector of all the jitter variables except \( \mathbf{z}_n \). Using Bayes rule and the independence of \( \mathbf{z}_n \) and \( w_n \),

\[
p(\mathbf{z}_n | \mathbf{z}_{\text{\textasciitilde}n}, \mathbf{x}, \sigma_x^2, \sigma_w^2, \zeta^2) = \frac{p(\mathbf{y} | \mathbf{z}_n, \mathbf{x}, \sigma_x^2, \sigma_w^2, \zeta^2)p(\mathbf{z}_n | \sigma_x^2)p(\mathbf{x} | \sigma_x^2)p(\sigma_x^2)p(\sigma_w^2)p(\zeta^2)}{p(\mathbf{y} | \mathbf{z}_{\text{\textasciitilde}n}, \mathbf{x}, \sigma_x^2, \sigma_w^2, \zeta^2)p(\mathbf{z}_{\text{\textasciitilde}n} | \sigma_x^2, \sigma_w^2, \zeta^2)}
\]

\[
\propto \mathcal{N}(\mathbf{y}; \mathbf{H}_n(\mathbf{z}_n)^T \mathbf{z}_n, \sigma_x^2, \sigma_w^2, \zeta^2)\mathcal{N}(\mathbf{z}_n; 0, \sigma_x^2). \tag{24}
\]

Slice sampling is used for generating realizations of \( \mathbf{z}_n \) by no tightly enveloping proposal density or other tuning is necessary; the ability to evaluate an unnormalized form of the target distribution is sufficient. Each iteration of slice sampling consists of two uniform sampling problems:

1) Choose a slice \( u \) uniformly from \( [0, \tilde{p}(z_n(i)) - y_i, \mathbf{z}_n, \sigma_x^2, \sigma_w^2, \zeta^2) \), where \( \tilde{p}(z_n(i)) \) is the unnormalized full conditional density function in (24).

Sample \( \mathbf{z}_n \) uniformly from the slice \( S \triangleq \{ \mathbf{z}_n : \tilde{p}(z_n(i)) \geq u \} \).

The first step is trivial, since we are sampling from a single interval. The second step is more difficult. However, since \( u \leq \tilde{p}(z_n(i)) | y, x, \sigma_x^2, \sigma_w^2, \zeta^2 \) for all \( z_n \) in the slice,

\[
\log u \leq -\frac{(y_n - \mathbf{H}_n(\mathbf{z}_n)x_i)^2}{2\sigma_x^2} - \frac{z_n^2}{2\sigma_z^2} - \log(2\pi \sigma_x \sigma_w) \leq -\frac{z_n^2}{2\sigma_z^2} - \log(2\pi \sigma_x \sigma_w). \tag{25}
\]
Solving for \( z_n \), the range of possible \( z_n \) is bounded:

\[
|z_n| \leq \sigma_x \sqrt{-2 \log u - 2 \log(2\pi \sigma_w \sigma_z)}.
\] (26)

Using these extreme points for the initial interval containing the slice, and the “shrinkage” method specified in [26] to sample from the slice by repeatedly shrinking the interval, slice sampling becomes a relatively efficient method. The “shrinkage” method decreases the size of the interval exponentially fast, on average. To see this, consider one iteration of shrinkage, where the initial point \( x_0 \) from the previous step of slice sampling lies in the interval \([L, R]\). This initial point is guaranteed to be in the slice by construction. The expected size of the new interval \([L', R']\), from choosing a new point \( x' \), is

\[
\mathbb{E}[R' - L' \mid R, L, x_0] = \frac{1}{R - L} \left( \int_L^{x_0} (R - x') dx' + \int_{x_0}^{R} (x' - L) dx' \right)
\]

\[
= \frac{R^2 - 2RL + L^2}{2(R-L)} + \frac{x_0(R+L-x_0) - RL}{R-L}
\]

\[
= \frac{R - L}{2} + \frac{x_0(R+L-x_0) - RL}{R-L}.
\] (27)

This expectation is quadratic in \( x_0 \), so the maximum occurs at the extreme point \( x_0 = (R+L)/2 \). The maximum value is

\[
\max_{x_0} \mathbb{E}[R' - L' \mid R, L, x_0] = \frac{R - L}{2} + \frac{(R+L)/2(R+L) - (R+L)/2 - RL}{R-L}
\]

\[
= \frac{R - L}{2} + \frac{(R+L)^2/4 - RL}{R-L} = \frac{3}{4}(R-L).
\] (28)

Concavity implies that the minima are at the two endpoints \( x_0 = L \) and \( x_0 = R \). In both cases, the expected size of the interval is \((R-L)/2\). Therefore,

\[
\frac{1}{4}(R-L) \leq \mathbb{E}[R' - L' \mid R, L, x_0] \leq \frac{3}{4}(R-L),
\] (29)

which implies that at worst, the size of the interval shrinks to 3/4 its previous size per iteration, on average. Then, given the initial interval \([L_0, R_0]\) and previous point \( x_0 \), the expected size of the interval \([L_1, R_1]\) after \( I \) iterations of the shrinkage algorithm is

\[
\mathbb{E}[R_I - L_I \mid R_0, L_0, x_0] = \mathbb{E}[\mathbb{E}[R_I - L_I \mid R_0, L_0, \ldots, R_{I-1}, L_{I-1}, x_0] \leq \left( \frac{3}{4} \right)^I (R_0 - L_0).
\] (30)

If the target distribution \( p(x) \) is continuous, the algorithm is guaranteed to terminate once the search interval is small enough. Since the interval size shrinks exponentially fast, on average, the number of “shrinkage” iterations is approximately proportional to the log of the fraction of the initial interval contained in the slice.

In the rejoinder at the end of [26], an alternative binary search-like midpoint shrinkage algorithm is proposed that can converge faster on the slice than the original shrinkage algorithm, at the cost of increasing correlation between successive samples, which reduces the overall Gibbs sampler convergence speed. In an effort to mitigate the increased correlation, a hybrid method is proposed in [26] that always shrinks to the rejected sample, then shrinks to the midpoint of the remaining interval only if the probability of the rejected sample is sufficiently small (the threshold) relative to the slice. These algorithms are applied to both unimodal and multimodal posterior distributions \( p(z_n \mid x, \sigma_z^2, \sigma_w^2, y_n) \) in Figure 5.

To compare these methods in the context of the jitter mitigation problem, consider the convergence of the complete Gibbs sampler, using the shrinkage methods described above. While the combined method obviously shrinks the slice much faster than the original method, the hybrid method’s increased speed must offset any increased correlation in the accepted samples in order to be useful. In Figure 6, the convergence metrics PSRF\[^{1/2}\] and \( \frac{\|V\|^2}{\|\hat{V}\|^2} \) are plotted as a function of the total number of shrinkage iterations performed. The convergence rate of the two shrinkage methods are very similar, but in some cases, as shown in Figure 6a, the hybrid method outperforms the original shrinkage method.

To summarize, pseudocode of the slice sampling algorithm using either shrinkage method to generate realizations of \( z_n \) is written in Algorithm 1.
(a) $K = 10$, $M = 16$, $\sigma_z = 0.1$, $\sigma_w = 0.05$, $n = 80$, original (left) and thresholding-based (right) methods.

(b) $K = 10$, $M = 4$, $\sigma_z = 0.5$, $\sigma_w = 0.075$, $n = 20$, original (left) and thresholding-based (right) methods.

Fig. 5. Comparisons between the original shrinkage method and the modified thresholding-based shrinkage method for unimodal \((\text{a})\) and multimodal \((\text{b})\) posterior distributions. The unnormalized distribution $\tilde{p}(z_n^{(i-1)} | \mathbf{x}, \sigma_x^2, \sigma_z^2, \sigma_w^2, y_n)$ is evaluated for the previous Gibbs sampler iteration’s $z_n^{(i-1)}$ and the slice level shown is selected uniformly from $[0, \tilde{p}(z_n^{(i-1)} | \ldots)]$. The shrinkage methods proceed according to Algorithm [1]. The midpoint threshold shown corresponds to $\tau = 25$. Rejected samples are marked with a “×”, and the final accepted sample is marked with a “o”. In both cases, the thresholding-based method reduces the size of the interval more quickly than the original method. Especially in the unimodal case, the accepted sample $z_n^{(i)}$ is much closer to the previous iterate $z_n^{(i-1)}$ than would otherwise be expected from the size of the slice.

B. Generating $\mathbf{x}$, $\sigma_x^2$, $\sigma_z^2$, and $\sigma_w^2$

The full conditional distribution on $x_k$ does depend on the other signal parameters $\mathbf{x}_{\backslash k}$:

$$p(x_k | \mathbf{x}_{\backslash k}, z, \sigma_x^2, \sigma_w^2, \sigma_z^2, y) = \frac{p(y | z, \mathbf{x}, \sigma_x^2)p(z | \sigma_z^2)p(x | \sigma_x^2)p(\sigma_x^2)p(\sigma_z^2)}{p(x_{\backslash k}, z, \sigma_x^2, \sigma_w^2, \sigma_z^2, y)} \propto \mathcal{N}(y; \mathbf{H}(z)\mathbf{x}, \sigma_w^2 \mathbf{I})\mathcal{N}(x_k; 0, \sigma_x^2) \mathcal{N}(z; 0, \sigma_z^2 \mathbf{I})$$  \hspace{1cm} (31)

Grouping correlated variables accelerates Gibbs sampler convergence, and the random vector $\mathbf{x}$ can still be generated in one simple step since

$$p(\mathbf{x} | z, \sigma_x^2, \sigma_w^2, \sigma_z^2, y) \propto \mathcal{N}(y; \mathbf{H}(z)\mathbf{x}, \sigma_w^2 \mathbf{I})\mathcal{N}(\mathbf{x}; 0, \sigma_x^2 \mathbf{I})$$  \hspace{1cm} (32)

implies the posterior distribution of $\mathbf{x}$ is just multivariate normal with mean

$$\mu_x = \Lambda_x \frac{\mathbf{H}(z)^T y}{\sigma_w^2}$$  \hspace{1cm} (33)

and covariance matrix

$$\Lambda_x = \sigma_x^2 [\mathbf{H}(z)^T \mathbf{H}(z) + \frac{\sigma_z^2}{\sigma_w^2} \mathbf{I}]^{-1}.$$  \hspace{1cm} (34)
is as simple as taking the inverse of realizations of a gamma
\[ \sigma \]
outperforms the original rejection-based shrinkage method in (a) and performs
shrinkage iterations is measured by the square root of the PSRF and
Fig. 6. One-hundred chains of the Gibbs/slice sampler are run for
1000 iterations each, and the convergence as a function of the total number of
shrinkage iterations is measured by the square root of the PSRF and \( \lVert V \rVert_{1/2} \).
The values of \( \alpha_x, \beta_x, \alpha_z, \beta_z, \alpha_w, \) and \( \beta_w \) are determined for the expected values of \( \sigma_x^2, \sigma_z^2, \) and \( \sigma_w^2 \) using (10) and (11). For a given number of
shrinkage iterations, the hybrid rejection-midpoint-threshold method (\( \tau = 25 \))
outperforms the original rejection-based shrinkage method in [6] and performs
equally well in [6].

\[ \propto \mathcal{N}(x; 0, \sigma_x^2) \mathcal{IG}(\sigma_x^2; \alpha_x, \beta_x). \]  (36)

and

\[ p(\sigma_w^2 | x, z, y, \sigma_x^2, \sigma_z^2) = p(\sigma_w^2 | x, z, y) = \frac{p(y | x, z, \sigma_w^2)p(x | y, z)}{p(x | y, z)} \propto \mathcal{N}(y; H(x)z, \sigma_w^2) \mathcal{IG}(\sigma_w^2; \alpha_w, \beta_w). \]  (37)

The inverse Gamma distribution is the conjugate prior for the
variance parameter of a Normal distribution (see (28)). Therefore, the posterior distribution is also an inverse Gamma distribution. Specifically,
\[ p(\sigma_x^2 | x, z, y, \sigma_z^2, \sigma_w^2) = \mathcal{IG}(\sigma_x^2; \alpha', \beta'), \]
where
\[ \alpha' = \alpha_x + \frac{K}{2}; \quad \beta' = \beta_x + \frac{\|x\|_2^2}{2}. \]  (38)

Similarly the hyperparameters for the posterior inverse Gamma
distributions on \( \sigma_x^2 \) and \( \sigma_w^2 \) are
\[ \alpha' = \alpha_x + \frac{N}{2}; \quad \beta' = \beta_x + \frac{\|z\|^2}{2}; \]  (39)
\[ \alpha'_w = \alpha_w + \frac{N}{2}; \quad \beta'_w = \beta_w + \frac{\|y - H(x)z\|^2}{2}. \]  (40)

Thus, generating realizations of \( \sigma_x^2 \) or \( \sigma_w^2 \) using such a prior
is as simple as taking the inverse of realizations of a gamma
distribution with the proper choice of hyperparameters. For
those who prefer a non-informative prior, the Jeffrey's priors
for \( \sigma_x^2, \sigma_z^2, \) and \( \sigma_w^2 \) are
\[ p(\sigma_x^2) = 1/\sigma_x^2, \quad p(\sigma_z^2) = 1/\sigma_z^2, \]  and
\[ p(\sigma_w^2) = 1/\sigma_w^2. \] Although these priors are improper
distributions, they are equivalent to inverse Gamma distributions
with \( \alpha = \beta = 0, \) so the associated posterior distributions
are proper inverse Gamma distributed with the parameters
described above.

Once enough samples have been taken so that the current
description of the Markov chain is sufficiently close to the steady
state, the Gibbs sampling theory tells us that further samples
are drawn from the chain can be treated as if they were drawn from
the joint posterior distribution directly. Thus, these additional
samples can be averaged to approximate the Bayes MMSE
estimator. In the complete Gibbs sampler in Algorithm (2) \( I_b \)
represents the “burn-in time,” the number of iterations until
the Markov chain has approximately reached its steady state,
and \( I \) represents the number of samples to generate after
convergence, which are averaged to form the MMSE estimates.

V. SIMULATION RESULTS

In this section, both the convergence behavior and the
performance of the Gibbs/slice sampler are analyzed. Using
Matlab, a \( K \)-parameter signal and \( N = K M \) samples of that
signal are generated with pseudo-random jitter and additive
noise; \( M \) is the oversampling factor. Then, implementations of
the Gibbs/slice sampler, as well as the linear MMSE estimator in (20),
the no-jitter linear estimator in (22), and the EM
algorithm developed in (14) for approximating the ML estimator
are applied to the samples. The adaptation of the EM algorithm
to random \( \sigma_w^2 \) and \( \sigma_z^2 \) is described in the appendix; however,
the EM algorithm with known \( \sigma_w^2 \) and \( \sigma_z^2 \) is used in these
simulations because adapting to random variances dramatically
increases the computational cost, and the difference in MSE is
negligible. These algorithms are studied in detail for periodic
bandlimited signals with uniformly distributed signal parameters
in [17], and in this work, a similar analysis is performed
to analyze the convergence and sensitivity to initial conditions
of the proposed algorithms. This analysis is also similar to

Algorithm 2 Pseudocode for the Gibbs sampler modified to
use slice sampling for the \( z_n \)’s.

Require: \( y, I, I_b \)
\[ z^{(0)} \rightarrow 0; \quad x^{(0)} \rightarrow \hat{x}_{LMMSE}|x=0(y) \text{ from (22)}; \quad \sigma_x^{(0)} \rightarrow 1; \]
\[ \sigma_z^{(0)} \rightarrow 0.01; \quad \sigma_w^{(0)} \rightarrow 0.01 \]
for \( i = 1 : I + I_b \) do
for \( n = 0 : N - 1 \) do
Generate \( z_n^{(i)} \) using slice sampling in Algorithm (11)
end for
Generate \( x^{(i)} \) from \( \mathcal{N}(\mu_x, \Lambda_x) \) using (33) and (34).
Generate \( \sigma_x^{(i)} \) from \( \mathcal{IG}(\alpha'_x, \beta'_x) \) using (38).
Generate \( \sigma_z^{(i)} \) from \( \mathcal{IG}(\alpha'_z, \beta'_z) \) using (39).
Generate \( \sigma_w^{(i)} \) from \( \mathcal{IG}(\alpha'_w, \beta'_w) \) using (40).
end for
\[ \hat{x} \rightarrow \frac{1}{I} \sum_{i=I_b+1}^{I} x^{(i)} \]
\[ \hat{z} \rightarrow \frac{1}{I} \sum_{i=I_b+1}^{I} z^{(i)} \]
\[ \hat{\sigma}_x^2 \rightarrow \frac{1}{I} \sum_{i=I_b+1}^{I} \sigma_x^{(i)} \]
\[ \hat{\sigma}_z^2 \rightarrow \frac{1}{I} \sum_{i=I_b+1}^{I} \sigma_z^{(i)} \]
\[ \hat{\sigma}_w^2 \rightarrow \frac{1}{I} \sum_{i=I_b+1}^{I} \sigma_w^{(i)} \]
return \( \hat{x}, \hat{z}, \hat{\sigma}_x^2, \hat{\sigma}_z^2, \hat{\sigma}_w^2 \).
that performed in [4] for the EM algorithm approximation to the ML estimator of the non-Bayesian version of this paper’s problem formulation.

A. Convergence Analysis

As a Markov chain Monte Carlo method, the Gibbs/slice sampler converges to the appropriate posterior distribution under certain conditions (see [22]); as long as the sequence generated by sampling from the steady-state distribution \( p(x, z, \sigma_x^2, \sigma_z^2, \sigma_w^2 | y) \) is ergodic, the samples can be averaged to approximate the Bayes MMSE estimate of the signal parameters. In addition, the steady-state distribution of an irreducible chain is unique, so the choice of initialization should not impact the final estimate generated from the steady-state samples. Of course, since the chain only converges to the steady-state in the limit, small transient effects from the initial conditions are evaluated.

The rate of convergence of the Gibbs/slice sampler, as measured by the \( ||V||^{1/2} \) and the square root of the PSRF, is shown in Figure 7. The results suggest that increasing the oversampling factor \( M \) or the jitter variance \( \sigma_z^2 \) or decreasing the additive noise variance \( \sigma_w^2 \) slows the rate of convergence. In most cases, the Markov chain appears to reach a steady state within 500 iterations; thus, we set \( I_0 = 500 \) iterations (see Algorithm 2) for the tests that follow.

To establish the number of iterations \( I \) needed after burn-in, we observe the squared error \( \| \hat{x}_I - x^* \|^2 \), where \( \hat{x}_I \) is the \( I \)th estimate of \( x \), as a function of \( I \), for \( I \) up to 1000, and \( x^* \) is the true value of \( x \). Examining the plots in Figure 8 approximately 500 iterations are sufficient to achieve a squared error within 0.5 dB of the asymptotic MSE (as measured by \( I = 1000 \)) for all cases.

The sensitivity to initial conditions of the Gibbs/slice sampler is shown in Figure 9 for \( I_0 = I = 500 \). For 50 trials, the squared error of the Bayes MMSE estimates are measured for ten different choices of initial conditions. The ten choices of initial conditions used are (1) \( \sigma_x^{(0)} = 1, \sigma_z^{(0)} = \sigma_w^{(0)} = 0.1 \), and all \( x^{(0)} \) and \( z^{(0)} \) equal to zero, (2) \( \sigma_x^{(0)} = 1, \sigma_z^{(0)} = \sigma_w^{(0)} = 0.1, z^{(0)} \) equal to zero, and the no-jitter LMMSE estimate for \( x^{(0)} \), (3) the true values of \( \sigma_x^2, \sigma_z^2, \sigma_w^2, z \), and \( x \), and (4-10) seven choices of random values of \( \sigma_x^2, \sigma_z^2, \sigma_w^2, z \) and the corresponding fixed-jitter LMMSE estimates for \( x \). The squared errors displayed are normalized so that the squared error for the no-jitter LMMSE estimate starting point equals one. Although the Gibbs/slice sampler becomes more sensitive to initial conditions as \( \sigma_z \) increases, in all cases, the squared errors for the majority of initial conditions are close to one. Thus, even though the algorithms are still sensitive to initial conditions after the burn-in period, especially for larger jitter variance, the choice of no-jitter LMMSE estimate is about average.

B. Performance Comparisons

In Figure 10 the performance of the Gibbs/slice sampler is compared against the linear MMSE and no-jitter linear MMSE estimators and the EM algorithm approximation to the ML estimator derived in [4]. The MSE performances are plotted for different values of \( M, \sigma_z \), and \( \sigma_w \) to demonstrate the effect of increasing \( M \), increasing \( \sigma_z \), or decreasing \( \sigma_w \) on the relative MSE performances. Comparing the Gibbs/slice sampler Bayes MMSE estimate against the linear estimator, the Gibbs/slice sampler outperforms the linear MMSE estimator for a large range of \( \sigma_z \), a difference that becomes more pronounced with higher oversampling \( M \). In addition, the results suggest that the Gibbs/slice sampler outperforms classical estimation, especially for higher jitter variances.

We also compare computation times for the EM algorithm and the Gibbs/slice sampler. Both converge more slowly for higher jitter and lower additive noise, and greater oversampling also lengthens computation. In the case of \( K = 10, M = 16, \) \( \mathbb{E}[\sigma_z^2] = 0.5^2 \), and \( \mathbb{E}[\sigma_w^2] = 0.025^2 \), the EM algorithm with known \( \sigma_z^2 \) and \( \sigma_w^2 \) requires 1.6 seconds per trial on average, the EM algorithm for random noise variances requires 24 seconds, and the Gibbs/slice sampler requires 3.1 seconds on average.
In only an eighth the time, the Gibbs/slice sampler achieves greater MSE performance than the EM algorithm.

To understand the effectiveness of these methods in mitigating jitter, the difference in jitter variance as a function of target MSE is computed based on the performance results and the maximum observed differences (for $E[\sigma^2_w]/25 \geq 1/2 E[\sigma^2_w]/25$, to avoid the region where the MSE plots are flat) are compared for different values of $M$ and $E[\sigma^2_w]/25$. The resulting trends portrayed in Figure 8 demonstrate that greater improvement is achievable with increased oversampling $M$, and small additive noise variance $E[\sigma^2_w]$. In addition, the Gibbs/slice sampler outperforms the classical ML estimator (as approximated by the EM algorithm in [4]) at high jitter, increasing the factor of improvement, especially in the high oversampling and low additive noise variance regimes.

VI. CONCLUSION

The results displayed in this paper suggest that post-processing jittered samples with a nonlinear algorithm like Gibbs/slice sampling mitigates the effect of sampling jitter on the total sampling error. In particular, the expected jitter standard deviation can be increased by as much as a factor of 2:2, enabling substantial power savings in the analog circuitry when compared against linear post-processing or classical nonlinear post-processing (the EM algorithm). Such power savings may enable significant improvements in battery life for implantable cardiac pacemakers and enable the inclusion of ADCs in ultra-low power devices.

Like the EM algorithm proposed in [4], the Gibbs/slice sampler proposed here suffers from relatively high computational complexity and an iterative nature, which may be unsuitable for embedded applications. Developments in polynomial estimators, such as the Volterra filter-like polynomial estimators described in [29], may yield similar performance to the Gibbs/slice sampler proposed here, at least for low levels of oversampling, without such high online computational cost. Further investigation is warranted in developing these and similar approaches for post-processing jittered samples in ADCs. Nevertheless, for off-chip post-processing of jittered samples, the nonlinear Bayesian Gibbs/slice sampler presented here outperforms both linear MMSE estimator and the nonlinear classical EM algorithm approximation to the ML estimator.
Fig. 10. The MSE performance of the Bayes MMSE estimator as computed using the Gibbs/slice sampler is compared against both the unbiased linear MMSE estimator (20) and the no-jitter linear MMSE estimator (22), as well as the EM algorithm approximation to the ML estimator from (4). The values of $\alpha$, $\beta_2$, $\beta_3$, $\beta_4$, and $\beta_5$ are determined for the average $\sigma_z^2$, $\sigma_w^2$, and $\sigma_n^2$ using (40) and (41). The EM algorithm uses the true values of $\sigma_2$, $\sigma_n^2$, and $\sigma_w^2$, while the linear estimators and the Gibbs/slice sampler treat $\sigma_2$, $\sigma_n^2$, and $\sigma_w^2$ as random variables. The error bars above and below each data point for the estimators delineate the 95% confidence intervals for those data points.

**APPENDIX**

**ML Estimation with Random Variances**

In (4), the EM algorithm approximation to the ML estimator is derived in the classical setting for known variances $\sigma_z^2$ and $\sigma_w^2$. To adapt the method for random variances, we introduce $\sigma_2^2$ and $\sigma_n^2$ as latent variables:

$$\hat{x}^{(i)} = \arg \max_x \mathbb{E} \left[ \log p(y, z, \sigma_z^2, \sigma_n^2, \sigma_w^2 | x; \hat{x}^{(i-1)} \right].$$

(41)

By conditional independence,

$$p(y, z, \sigma_z^2, \sigma_n^2, \sigma_w^2 | x) = p(y | z, \sigma_z^2, \sigma_n^2, \sigma_w^2) p(z | \sigma_z^2, \sigma_n^2, \sigma_w^2)$$

$$= \mathcal{N}(y; H(z) x, \sigma_z^2 \sigma_n^2) \mathcal{N}(z | \sigma_z^2, \sigma_n^2, \sigma_w^2).$$

(42)

The terms not involving $x$ are unnecessary, since we are differentiating with respect to $x$ in the next step. The derivative of the expectation in (41) is

$$\mathbb{E} \left[ -\frac{2H^T(z)(H(z)x - y)}{2\sigma_w^2} | y; \hat{x}^{(i-1)} \right].$$

(43)

Setting the derivative equal to zero yields a linear system in $x$:

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

(44)

As is done in (4), the expectations in (44) become:

$$\mathbb{E} \left[ \frac{H^T(z)H(z)}{\sigma_w^2} | y, \hat{x}^{(i-1)} \right] = \sum_{n=0}^{N-1} \mathbb{E} \left[ \frac{h_n(z_n) h_n^T(z_n)}{\sigma_w^2} | y_n, \hat{x}^{(i-1)} \right];$$

(45)

$$\mathbb{E} \left[ \frac{H(z)}{\sigma_w^2} | y, \hat{x}^{(i-1)} \right] = \mathbb{E} \left[ \frac{h_n(z_n)}{\sigma_w^2} | y_n, \hat{x}^{(i-1)} \right];$$

(46)
The hybrid quadrature method discussed in Section II can be used to compute the expectations in (45) and (46):

\[
\mathbb{E} \left[ \frac{\mathbf{h}_n(z_n)L_n^T(z_n)}{\sigma_w^2} \right] | y_n, \mathbf{x}^{(i-1)} \approx \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} \frac{w_{j_1} w_{j_2} w_{j_3} \mathbf{h}_n(z_n)}{\sigma_w^2 p(y_n, \mathbf{x}^{(i-1)})} 
\]

(47)

Hybrid quadrature is also used to compute \( p(y_n, \mathbf{x}^{(i-1)}) \) (see (15)). Then, the EM algorithm becomes iteratively solving (44) for \( \mathbf{x}^{(i)} \), using the above hybrid quadrature formulas. However, due to the three-dimensional nature of the hybrid quadrature formulas, computational cost can increase dramatically.

Due to the increased computational cost of adapting the EM algorithm to random variances, we compare the MSE performance of both EM algorithms for the same choices of parameters used in the performance plots in [4] (1000 trials, \( J_1 = J_2 = 9, J_3 = 129 \)). The MSE performance for both algorithms are almost identical, up to only 0.54 dB apart. Thus, to reduce computation time when comparing performance against the Gibbs/slice sampler, the EM algorithm with known variances is used as a proxy for the EM algorithm with random variances.

ACKNOWLEDGMENT

The authors thank V. Y. F. Tan for valuable discussions on Gibbs sampling and J. Kusuma for asking stimulating questions about sampling and applications of jitter mitigation. The authors also thank Z. Zvonar at Analog Devices and G. Frantz at Texas Instruments for their insights and support.

REFERENCES