On the Estimation of Nonrandom Signal Coefficients From Jittered Samples

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

Abstract—This paper examines the problem of estimating the parameters of a bandlimited signal from samples corrupted by random jitter (timing noise) and additive, independent identically distributed (i.i.d.) Gaussian noise, where the signal lies in the span of a finite basis. For the presented classical estimation problem, the Cramér–Rao lower bound (CRB) is computed, and an Expectation-Maximization (EM) algorithm approximating the maximum likelihood (ML) estimator is developed. Simulations are performed to study the convergence properties of the EM algorithm and compare the performance both against the CRB and a basic linear estimator. These simulations demonstrate that by postprocessing the jittered samples with the proposed EM algorithm, greater jitter can be tolerated, potentially reducing on-chip ADC power consumption substantially.

Index Terms—Analog-to-digital converters, Cramér–Rao bound, EM algorithm, jitter, maximum likelihood estimator, sampling, timing noise.

I. INTRODUCTION

A N analog-to-digital converter (ADC) processes a real signal $x(t)$ to generate a sequence of observations (samples) $\{y_n\}$ at times $\{t_n\}$:

$$y_n = [s(t) * x(t)]_{t=t_n} + w_n \qquad (1)$$

where $s(t)$ is the sampling prefilter and w_n is an additive noise term that lumps together quantization, thermal noise, and other effects. For standard sampling applications, it is assumed that the sample times are uniformly spaced by some period T_s ($t_n = nT_s$, $n \in \mathbb{Z}$), where the period is small enough that the total bandwidth of $x(t)$ is less than the sampling rate $F_s = 1/T_s$. Jitter $\{z_n\}$, also known as timing noise, perturbs the sample times:

$$t_n = nT_s + z_n. \qquad (2)$$

This paper focuses on the mitigation of random jitter in a non-Bayesian estimation framework. A simplified block diagram for the overall system is illustrated in Fig. 1.

The generally accepted practice is to design clocks with low enough phase noise that the effect of jitter is negligible. The
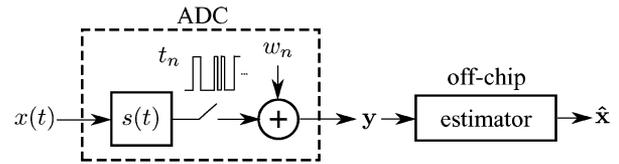
Fig. 1. Abstract block diagram of an ADC with off-chip postprocessing. The signal $x(t)$ is filtered by the sampling prefilter $s(t)$ and sampled at time t_n. These samples are corrupted by additive noise w_n to yield y_n. The postprocessor estimates the parameters $\mathbf{x}$ of $x(t)$ using the vector of N samples $\mathbf{y}$ from the ADC.

maximum allowable jitter is set such that the effect of jitter on a sinusoid of maximum frequency and maximum amplitude is at most one-half the least significant bit level [1]. While making jitter negligible obviates the mitigation of jitter, this may not be possible or desirable from an overall system design perspective, because requiring jitter to have a negligible effect may mandate high power consumption in the clock circuitry. (A model relating jitter and power is given below.) Technological trends suggest that eventually it will be worthwhile to allow nontrivial jitter and compensate through digital postprocessing: the digital portions of mixed-signal systems like sensors and wireless transceivers continue to shrink, so the analog portions of such systems, including the ADC and its clock generator, dominate the size and power consumption of these chips. The ability to use more power-efficient analog circuitry would enable substantial new capabilities in diverse applications like implantable medical devices and remote sensors. One motivation of our study is to contribute to understanding the trade-off between accuracy in analog circuitry versus complexity of off-chip digital postprocessing of samples.

The power consumed by a typical ADC design is approximately proportional to the desired accuracy and sampling rate [2], so lower-power circuitry would produce clock signals with more jitter. Specifically, it is shown in [3] that

$$\text{Power} \propto \text{Speed} \times (\text{Accuracy(rms)})^2. \qquad (3)$$

Furthermore, the analyses in [4] and [5] suggest that in the large-jitter domain, every doubling of the standard deviation of the jitter reduces the effective number of bits $(\text{ENOB} = \log_2 \text{ accuracy (rms)})$ by one. Thus, precompensating for the expected jitter in the design requires increasing the power consumption of the ADC by a factor of four for every doubling of the jitter standard deviation (e.g., by adding an additional level of comparators). In this paper, we instead propose block postprocessing the jittered samples using classical estimation techniques off-chip. In addition to mitigating random jitter, this work may also be adapted to compensate for frequency modulated

and spread-spectrum clocks, which produce lower EMI and radiation [6]. Note that this block postprocessing method is intended to be performed off-chip, where power consumption of an implementation of the algorithm is not important.

A. Problem Formulation

While more sophisticated signal models may be more appropriate for some applications, we concern ourselves with a signal that lies in the span of a finite basis $\{h_k(t)\}$, with K basis functions. We further restrict the basis to be uniform shifts of a single smooth bandlimited function $h(t)$:

$$h_k(t) = h(t - kT). \qquad (4)$$

Denote the unknown weighting parameters x_k; there are K of them. The signal $x(t)$ then equals

$$x(t) = \sum_{k=0}^{K-1} x_k h(t - kT). \qquad (5)$$

Without loss of generality, assume that T equals the critical sampling period of $x(t)$, and assume this period is unity. In this work, the signal parameters $\{x_k\}_{k=0}^{K-1}$ are unknown deterministic quantities.

While there are many possible choices of $h(t)$, when in need of a specific example in this work, we choose the sinc interpolating function $\text{sinc}(t) = \sin(\pi t)/\pi t$. This basis satisfies $x(k) = x_k$, for all $k = 0, \ldots, K-1$. In general, we choose $h(t)$ to be appropriate for the class of input signals we wish to sample; we choose the sinc function because bandlimitedness is a common assumption in the context of signal processing. It is sufficient, but not necessary, in this work that $h(t)$ be analytic and bounded.

When sampling the signal $x(t)$, we will assume that the sampling prefilter is an ideal anti-aliasing filter with bandwidth F_s, so $[s(t) * x(t)]_{t=t_n} = x(t_n)$ for appropriately bandlimited inputs. The signal's critical sampling period is assumed to be one, but to accommodate oversampling by a factor of M into our model, the ideal sample times are spaced $1/M$ time units apart. We acquire N jittered samples with additive noise, $y_0, \ldots, y_{N-1}$, at this rate:

$$y_n = x(n/M + z_n) + w_n. \qquad (6)$$

In this paper, we assume that the jitter and additive noise are independent identically distributed (i.i.d.) zero-mean Gaussian, with known variances equal to σ_z^2 and σ_w^2, respectively. We assume that these variances can be measured reasonably accurately through in-factory calibration, although we expect the variances to vary naturally over time due to environmental effects.

Combining the signal and observation models yields

$$y_n = \sum_{k=0}^{K-1} h(n/M + z_n - k)x_k + w_n. \qquad (7)$$

This relationship can be expressed as a semilinear system of equations:

$$\mathbf{y} = \mathbf{H}(\mathbf{z})\mathbf{x} + \mathbf{w}, \qquad (8)$$

where $\mathbf{y} = [y_0, \ldots, y_{N-1}]^T$, $\mathbf{x} = [x_0, \ldots, x_{K-1}]^T$, $\mathbf{z} = [z_0, \ldots, z_{N-1}]^T$, and $\mathbf{w} = [w_0, \ldots, w_{N-1}]^T$. For notational convenience, denote the nth (zero-indexed) row of $\mathbf{H}(\mathbf{z})$ by $\mathbf{h}_n^T(z_n)$.

To keep notation compact, denote the probability density function (pdf) of $\mathbf{a}$ by $p(\mathbf{a})$, the pdf of $\mathbf{a}$ parameterized by the nonrandom vector $\mathbf{b}$ by $p(\mathbf{a}; \mathbf{b})$, and the pdf of $\mathbf{a}$ conditioned on the random variable $\mathbf{c}$ by $p(\mathbf{a}|\mathbf{c})$. The pdf is made explicit using subscripts only when necessary to avoid ambiguity. Expectations are written similarly. Also, denote the d-dimensional multivariate normal distribution by

$$\mathcal{N}(\mathbf{a}; \boldsymbol{\mu}, \boldsymbol{\Lambda}) \triangleq |2\pi\boldsymbol{\Lambda}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{a}-\boldsymbol{\mu})^T \boldsymbol{\Lambda}^{-1}(\mathbf{a}-\boldsymbol{\mu})\right\}. \qquad (9)$$

The primary objective of classical (non-Bayesian) estimation is to derive an estimator $\hat{\mathbf{x}}$ that minimizes a desired cost function $C(\hat{\mathbf{x}}; \mathbf{x})$ of unknown nonrandom parameters $\mathbf{x}$. One such cost function is the mean-square error (MSE):

$$C(\hat{\mathbf{x}}; \mathbf{x}) = \mathbb{E}_{\mathbf{y}}\left[\|\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x}\|_2^2\right] \qquad (10)$$

where $\mathbb{E}_{\mathbf{y}}[\cdot]$ is the expectation with respect to $\mathbf{y}$, $\hat{\mathbf{x}}(\mathbf{y})$ is the estimate of the unknown parameters $\mathbf{x}$ based on the samples $\mathbf{y}$, and the observations $\mathbf{y}$ are implicitly a function of $\mathbf{x}$. However, except in certain cases, computing the estimate that minimizes this cost function, which is called the minimum MSE (MMSE) estimator, is impossible without prior knowledge about $\mathbf{x}$. Instead, it is essential to derive an estimator that relies only on the observation model and the actual observations $\mathbf{y}$. One such estimator is the maximum likelihood (ML) estimator, which maximizes the likelihood function $\ell(\mathbf{x}; \mathbf{y}) \triangleq p(\mathbf{y}; \mathbf{x})$. The likelihood function corresponding to the signal parameter observation model in (8) is

$$\ell(\mathbf{x}; \mathbf{y}) = \int \mathcal{N}\left(\mathbf{y}; \mathbf{H}(\mathbf{z})\mathbf{x}, \sigma_w^2\mathbf{I}\right) \mathcal{N}\left(\mathbf{z}; \mathbf{0}, \sigma_z^2\mathbf{I}\right) d\mathbf{z}. \qquad (11)$$

Using the assumptions that the jitter and additive noise are i.i.d., and the fact that the nth row of $\mathbf{H}(\mathbf{z})$ only depends on one z_n, the multivariate normal distributions in (11) are separable over $\mathbf{z}$. Thus, $p(\mathbf{y}; \mathbf{x})$ is also separable, with $p(\mathbf{y}; \mathbf{x}) = \prod_n p(y_n; \mathbf{x})$, and the likelihood function is the product of N univariate integrals

$$\ell(\mathbf{x}; \mathbf{y}) = \prod_{n=0}^{N-1} \int \mathcal{N}\left(y_n; \mathbf{h}_n^T(z_n)\mathbf{x}, \sigma_w^2\right) \mathcal{N}\left(z_n; 0, \sigma_z^2\right) dz_n. \qquad (12)$$

Given the likelihood function, parameters M, σ_z^2, and σ_w^2, MSE cost function, and observations $\mathbf{y}$, the goal of this work is to use ML estimation to tolerate more jitter when estimating $\mathbf{x}$. Thus, the bulk of this paper is concerned with the evaluation of this likelihood function and the problem of maximizing it.

B. Related Work

The problem of mitigating jitter has been investigated since the early days of signal processing. The effects of jitter on the statistics of samples of a deterministic (nonrandom) bandlimited signal are briefly discussed in [7]; this work also is concerned

with stochastic signals and proposes an optimal linear reconstruction filter for the stochastic case. Much more work on analyzing the error and reconstructing stochastic signals from jittered samples can be found in [8] and [9]. However, the analysis of jittered samples of deterministic signals appears to be much more limited in the early literature.

When the sample times are irregularly spaced, but known, the problem greatly simplifies. Efficient techniques, as well as a mention of prior work, can be found in [10]. When the sample times are unknown, but belong to a known finite set, the jitter mitigation problem becomes a combinatorial one; [11] describes geometric and algebraic solutions to this problem of reconstructing discrete-time signals. Two block-based reconstruction methods for this finite location-set problem are described in [12].

However, when the set is infinitely large, or when the jitter is described by a continuous random distribution as seen here, a different approach is necessary. One contribution of this work is an Expectation-Maximization (EM) algorithm; in a similar context, [13] develops a similar EM algorithm for the related problem of mitigating unknown phase offsets between component ADCs in a time-interleaved ADC system. Some of the results summarized in this paper are described in greater detail in [14], which also provides further background material.

C. Outline

In Section II, numerical integration using Gauss quadrature and iteration using the EM algorithm are discussed. Section III presents and derives the Cramér–Rao lower bound (CRB) on the MSE for this estimation problem. Sections IV and V derive linear and ML estimators for the jitter mitigation problem; simulations comparing these estimators are discussed in Section VI. In conclusion, the results and contributions are summarized, and future research directions are introduced.

II. BACKGROUND

Except for certain limited choices for $h(t)$, the expression for the likelihood function in (12) has no closed form; however, various techniques exist to approximate it. One such powerful and general technique is that of quadrature, which refers to the method of approximating an integral with a finite weighted summation. The trapezoidal and Simpson's rules are elementary examples of quadrature. In particular, due to the normal distribution assumption on the jitter z_n, Gauss–Hermite quadrature is a natural choice of quadrature rule. Gauss–Legendre quadrature and Romberg's method are also discussed below.

Computational problems also occur when deriving the ML estimator, due to the nonconcave and high-dimensional nature of the likelihood function. One local approximation technique called the EM algorithm can be used to locate local maxima in a computationally feasible manner. The EM algorithm is also reviewed in this section.

A. Numerical Integration

Consider approximating the integral $\int f(x)w(x)dx$ using the summation $\sum_{j=1}^{J} w_j f(x_j)$, where x_j and w_j are fixed abscissas (sampling locations) and weights. This type of approximation is known generally as quadrature. When the

abscissas are uniformly spaced, the summation is known as interpolatory quadrature; the trapezoidal and Simpson's rules, as well as Romberg's method [15], are of this type. Gauss quadrature seeks greater accuracy for a given number of function evaluations by allowing the abscissas to be spaced nonuniformly. An appropriate choice of abscissas and weights (called a rule) can be precomputed for a choice of $w(x)$ and J using a variety of methods, including a very efficient eigenvalue-based method derived in [16]. Orthogonal polynomials satisfy a three-term recursive relationship, which is used to form a tri-diagonal matrix, whose eigenvalues are the abscissas, and whose eigenvectors yield the weights. The eigendecomposition of a tri-diagonal matrix is very efficient, so quadrature rules are very inexpensive to compute, even for very large J. Quadrature is particularly attractive when $f(x)$ is smooth and has bounded derivatives. This method can be applied to multivariate integration as well, although in the absence of separability, the complexity scales exponentially with the number of variables.

One weighting function of particular interest in this work is the pdf of the normal distribution. For a standard normal distribution, the associated form of quadrature is known as Gauss-Hermite quadrature, since the abscissas and weights derive from Hermite polynomials. Using elementary changes of variables, this method can be generalized to normal distributions with arbitrary mean μ and variance σ^2:

$$\int_{-\infty}^{\infty} f(x)\mathcal{N}(x; \mu, \sigma^2)dx \approx \sum_{j=1}^{J} w_j f(\sigma x_j + \mu) \quad (13)$$

where w_j and x_j are the weights and abscissas for Gauss-Hermite quadrature with a standard normal weighting function. As mentioned in [17], the approximation error for Gauss-Hermite quadrature is bounded by the function's derivatives:

$$\left| \int_{-\infty}^{\infty} f(x)\mathcal{N}(x; \mu, \sigma^2)dx - \sum_{j=1}^{J} w_i f(\sigma x_j + \mu) \right|$$
$$\leq \frac{J!\sigma^{2J}}{(2J)!} \max_x \left| f^{(2J)}(x) \right|. \quad (14)$$

As long as $f(x)$ is sufficiently smooth, the $(2J)!$ term in the denominator dominates the above expression for large J, and the approximation error goes to zero superexponentially fast. While general conditions for convergence are difficult to isolate for arbitrary $f(x)$, a sufficient condition for convergence mentioned in [17] is that

$$\lim_{x \to \infty} \frac{|f(x)| \, |x|^{1+\rho}}{e^{x^2}} \leq 1, \quad \text{for some } \rho > 0. \quad (15)$$

Many other Gauss quadrature rules exist; one simple rule also considered is called Gauss–Legendre quadrature and is defined for integrating over a finite interval $[a, b]$, with the weighting function $w(x) \equiv 1$:

$$\int_a^b f(x)dx \approx \sum_{j=1}^{J} w_j f(x_j). \quad (16)$$

The abscissas and weights for Gauss–Legendre quadrature can be computed using the eigenvalue-based method mentioned
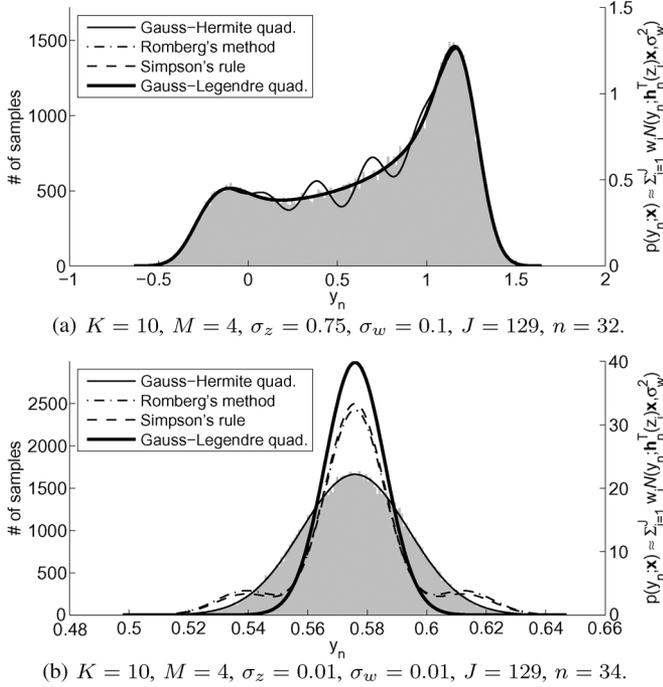
(a) $K = 10$, $M = 4$, $\sigma_z = 0.75$, $\sigma_w = 0.1$, $J = 129$, $n = 32$.

(b) $K = 10$, $M = 4$, $\sigma_z = 0.01$, $\sigma_w = 0.01$, $J = 129$, $n = 34$.

Fig. 2. The quadrature approximation to $p(y_n; \mathbf{x})$ is computed for a fixed, but randomly chosen, $\mathbf{x}$ on a dense grid of y_n and is compared against a histogram of 100 000 samples y_n generated using (7). Two cases are shown to illustrate the approximation quality of (a) Gauss–Legendre quadrature and (b) Gauss–Hermite quadrature; the worst-case n is chosen for each of these approximations. Note: the choice of $J = 129$ is used instead of $J = 100$ because Romberg's method evaluates the function $J = 2^j + 1$ times for j iterations. (a) $K = 10$, $M = 4$, $\sigma_z = 0.75$, $\sigma_w = 0.1$, $J = 129$, $n = 32$. (b) $K = 10$, $M = 4$, $\sigma_z = 0.01$, $\sigma_w = 0.01$, $J = 129$, $n = 34$.

above. Gauss–Legendre quadrature and other rules defined over a finite interval, including interpolatory quadrature methods like Simpson's rule and Romberg's method, can be extended to the infinite support case by remapping the variable of integration:

$$\int_{-\infty}^{\infty} f(x)w(x)dx = \int_{-\pi/2}^{\pi/2} f(\tan(y)) \, w(\tan(y)) \sec^2(y) dy. \quad (17)$$

When applied to the Gauss–Legendre quadrature rule, the new rule becomes

$$\int_{-\infty}^{\infty} f(x)w(x)dx \approx \sum_{j=1}^{J} w'_j f(x'_j) \quad (18)$$

where $x'_j = \tan(x_j)$, and $w'_j = w(x'_j)(1 + (x'_j)^2)w_j$.

To compare the effectiveness of these different quadrature-based methods for numerical integration, Gauss–Hermite quadrature and Gauss–Legendre quadrature are contrasted against two more general methods, Simpson's rule and Romberg's method, by comparing each method against the marginal likelihood function $p(y_n; \mathbf{x})$, for a fixed, but randomly chosen, value of $\mathbf{x}$. The marginal likelihood function is calculated from the empirical distribution of samples generated by the observation model in (7). As shown in Fig. 2, Gauss–Legendre quadrature approximates the likelihood function well when σ_z is relatively large, but when σ_z and σ_w are both small,

Gauss–Hermite quadrature is much more effective. However, other quadrature rules may be more accurate for different choices of signal basis functions $\{h_k(t)\}$.

B. EM Algorithm

The EM algorithm was introduced in [18]; a classic application of this algorithm is ML estimation in the presence of incomplete data. Consider the problem of maximizing the likelihood function $\ell(\mathbf{x}; \mathbf{y})$, where $\mathbf{y}$ depends on some latent random variables $\mathbf{z}$. The observations $\mathbf{y}$ are described as the incomplete data. We augment this incomplete data with some subset of latent (hidden) variables to form the complete data. The underlying assumption of the EM algorithm is that knowledge of the complete data makes the ML estimation problem easier to solve.

The EM algorithm consists of repeatedly maximizing the function

$$Q\left(\mathbf{x}; \hat{\mathbf{x}}^{(i-1)}\right) = \mathbb{E}\left[\log p(\mathbf{y}, \mathbf{z}; \mathbf{x}) | \mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right] \quad (19)$$

with respect to the desired parameters $\mathbf{x}$; the maximizing value becomes $\hat{\mathbf{x}}^{(i)}$, which is used in the next iteration. As long as the original likelihood function is bounded above, and some other mild conditions are satisfied, this algorithm is guaranteed to converge to a local maximum of the likelihood function [18].

Much has been written about the convergence rate of EM algorithms. In [19], the rate of convergence of the EM algorithm is related to the difference in the CRB using the incomplete data and the CRB using the complete data (incomplete data + latent variables). The supplemented EM algorithm in [20] also obtains Fisher information estimates, conditioned on the observations $\mathbf{y}$, which can be used to evaluate the quality of the resulting approximation to the ML estimate.

Since the likelihood function in (11) is not in general strictly concave, the presence of many critical points is a potential problem for any local algorithm. Simulated annealing [21] and other methods can be combined with the EM algorithm to improve robustness to getting trapped in local extrema.

III. APPROXIMATING THE CRB

Minimizing the MSE without access to prior information about the unknown parameters $\mathbf{x}$ may be impossible in the general case. However, even in situations when the MMSE estimator cannot be computed, the Cramér-Rao lower bound on the minimum achievable MSE by an unbiased estimator may be straightforward to compute. The $\text{CRB}_{\mathbf{y}}(\mathbf{x})$, where $\mathbf{x}$ is the variable to estimate from observations $\mathbf{y}$, is defined to be the trace of the inverse of the Fisher information matrix $\mathbf{I}_{\mathbf{y}}(\mathbf{x})$, which is defined as

$$\mathbf{I}_{\mathbf{y}}(\mathbf{x}) \triangleq \mathbb{E}\left[\left(\frac{\partial \log \ell(\mathbf{x}; \mathbf{y})}{\partial \mathbf{x}}\right)\left(\frac{\partial \log \ell(\mathbf{x}; \mathbf{y})}{\partial \mathbf{x}}\right)^T\right]. \quad (20)$$

Assuming the likelihood function satisfies the regularity condition

$$\mathbb{E}\left[\frac{1}{\ell(\mathbf{x}; \mathbf{y})} \frac{\partial^2 \ell(\mathbf{x}; \mathbf{y})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right] = \int_{-\infty}^{\infty} \frac{\partial^2 \ell(\mathbf{x}; \mathbf{y})}{\partial \mathbf{x} \partial \mathbf{x}^T} d\mathbf{y} = \mathbf{0}, \quad (21)$$

the Fisher information matrix can be expressed in terms of the Hessian of the log-likelihood:

$$[\mathbf{I_y}(\mathbf{x})]_{j,k} = \mathbb{E}\left[\frac{\partial \log \ell(\mathbf{x};\mathbf{y})}{\partial x_j}\frac{\partial \log \ell(\mathbf{x};\mathbf{y})}{\partial x_k}\right]$$

$$= \mathbb{E}\left[\frac{1}{\ell(\mathbf{x};\mathbf{y})^2}\frac{\partial \ell(\mathbf{x};\mathbf{y})}{\partial x_j}\frac{\partial \ell(\mathbf{x};\mathbf{y})}{\partial x_k}\right]$$

$$= \mathbb{E}\left[\frac{1}{\ell(\mathbf{x};\mathbf{y})^2}\frac{\partial \ell(\mathbf{x};\mathbf{y})}{\partial x_j}\frac{\partial \ell(\mathbf{x};\mathbf{y})}{\partial x_k}\right.$$
$$\left. - \frac{1}{\ell(\mathbf{x};\mathbf{y})}\frac{\partial^2 \ell(\mathbf{x};\mathbf{y})}{\partial x_j \partial x_k}\right]$$

$$= \mathbb{E}\left[\frac{\partial}{\partial x_k}\left(\frac{-1}{\ell(\mathbf{x};\mathbf{y})}\frac{\partial \ell(\mathbf{x};\mathbf{y})}{\partial x_j}\right)\right] \tag{22}$$

$$\mathbf{I_y}(\mathbf{x}) = -\mathbb{E}\left[\frac{\partial^2 \log \ell(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}\partial \mathbf{x}^T}\right]. \tag{23}$$

A sufficient condition for the regularity condition in (21) is that differentiation and integration interchange, since

$$\int_{-\infty}^{\infty}\frac{\partial^2 \ell(\mathbf{x};\mathbf{y})}{\partial \mathbf{x}\partial \mathbf{x}^T}d\mathbf{y} = \frac{\partial^2}{\partial \mathbf{x}\partial \mathbf{x}^T}\left[\int_{-\infty}^{\infty}\ell(\mathbf{x};\mathbf{y})d\mathbf{y}\right] = 0. \tag{24}$$

By basic analysis, uniform convergence of the integral in the likelihood function in (11) implies that the above regularity condition holds.

Since the likelihood function is separable, the log-likelihood function can be expressed as the summation of marginal log-likelihood functions; i.e.,

$$\log \ell(\mathbf{x};\mathbf{y}) = \sum_{n=0}^{N-1}\log p(y_n;\mathbf{x}), \tag{25}$$

which means that (23) can be rewritten as (again, assuming regularity conditions are satisfied)

$$\mathbf{I_y}(\mathbf{x}) = \sum_{n=0}^{N-1}\mathbb{E}\left[\left(\frac{\partial \log p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)\left(\frac{\partial \log p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)^T\right]. \tag{26}$$

The marginal pdf $p(y_n;\mathbf{x})$ can be computed numerically using quadrature:

$$p(y_n;\mathbf{x}) \approx \sum_{j=1}^{J}w_j\mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right) \tag{27}$$

where z_j and w_j are the abscissas and weights for the chosen quadrature rule. As depicted in Fig. 2, Gauss–Hermite quadrature is a good choice for small σ_z, and Gauss–Legendre quadra-

ture is more accurate for larger σ_z (>0.1). For all simulations in this paper, we use $J = 100$ unless otherwise specified.

The derivative of the marginal pdf can be approximated similarly:

$$\frac{\partial p(y_n;\mathbf{x})}{\partial \mathbf{x}} = \int\frac{\left(y_n - \mathbf{h}_n^T(z_n)\mathbf{x}\right)\mathbf{h}_n(z_n)}{\sigma_w^2}$$
$$\cdot \mathcal{N}\left(y_n;\mathbf{h}_n^T(z_n)\mathbf{x},\sigma_w^2\right)\mathcal{N}(z_n;0,\sigma_z^2)dz_n$$
$$\approx \sum_{j=1}^{J}w_j\frac{\left(y_n - \mathbf{h}_n^T(z_j)\mathbf{x}\right)\mathbf{h}_n(z_j)}{\sigma_w^2}$$
$$\times \mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right). \tag{28}$$

Since $(\partial/\partial \mathbf{x})[\log p(y_n;\mathbf{x})] = (1/p(y_n;\mathbf{x}))(\partial p(y_n;\mathbf{x})/\partial \mathbf{x})$, combining (27) and (28) yields a complicated approximation to the expression inside the expectation in (26) [see (29) shown at the bottom of the page]. For convenience, denote the above approximation $\mathbf{F}_n(y_n;\mathbf{x})$.

Now, consider computing the Fisher information matrix from this approximation:

$$\mathbf{I_y}(\mathbf{x}) \approx \sum_{n=0}^{N-1}\mathbb{E}\left[\mathbf{F}_n(y_n;\mathbf{x})\right]. \tag{30}$$

To compute this expectation, a numerical method is needed again. The expectation is with respect to the distribution $p(y_n;\mathbf{x})$, which is approximated in (27) with a Gaussian mixture, so Monte Carlo sampling is a convenient method to approximate this expectation. Generating S samples $y_{n,s}$ from the Gaussian mixture $\sum_{j=1}^{J}w_j\mathcal{N}(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2)$ and averaging the corresponding function values $\mathbf{F}_n(y_{n,s};\mathbf{x})$, the Fisher information matrix can be computed as

$$\mathrm{I_y}(\mathrm{x}) \approx \frac{1}{S}\sum_{n=0}^{N-1}\sum_{s=1}^{S}\mathbf{F}_n(y_{n,s};\mathbf{x}). \tag{31}$$

Once this matrix is computed and inverted, the trace gives the Cramér–Rao lower bound for that choice of parameter $\mathbf{x}$. Due to matrix inversion, to ensure an accurate CRB estimate, we use $J = 1000$ for the quadratures in (29).

How much does the CRB decrease when $\mathbf{z}$ is assumed given? Comparing the $\mathrm{CRB_y}(\mathbf{x})$ against the $\mathrm{CRB_{y,z}}(\mathbf{x})$ of the jitter-augmented data will be important later when analyzing the EM algorithm design. The Fisher information matrix in this case is equal to

$$\mathrm{I_{y,z}}(\mathrm{x}) = -\sum_{n=0}^{N-1}\mathbb{E}\left[\frac{\partial^2 \log p(y_n,z_n;\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^T}\right]. \tag{32}$$

$$\left(\frac{\partial \log p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)\left(\frac{\partial \log p(y_n;\mathbf{x})}{\partial \mathbf{x}}\right)^T \approx \left(\frac{\sum_{j=1}^{J}w_j\left(y_n - \mathbf{h}_n^T(z_j)\mathbf{x}\right)\mathbf{h}_n(z_j)\mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right)}{\sigma_w^2\sum_{j=1}^{J}w_j\mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right)}\right)$$
$$\cdot \left(\frac{\sum_{j=1}^{J}w_j\left(y_n - \mathbf{h}_n^T(z_j)\mathbf{x}\right)\mathbf{h}_n(z_j)\mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right)}{\sigma_w^2\sum_{j=1}^{J}w_j\mathcal{N}\left(y_n;\mathbf{h}_n^T(z_j)\mathbf{x},\sigma_w^2\right)}\right)^T. \tag{29}$$

Of course, since

$$-\log p(y_n, z_n; \mathbf{x}) = \frac{1}{2\sigma_w^2} \left(y_n - \mathbf{h}_n^T(z_n)\mathbf{x} \right)^2$$
$$+ \frac{1}{2\sigma_z^2} z_n^2 + \text{constant}, \quad (33)$$

and the Hessian matrix with respect to $\mathbf{x}$ is

$$\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^T} \log p(y_n, z_n; \mathbf{x}) = -\frac{1}{\sigma_w^2} \mathbf{h}_n(z_n)\mathbf{h}_n^T(z_n), \quad (34)$$

the jitter-augmented Fisher information matrix is

$$\mathrm{I}_{\mathrm{y,z}}(\mathrm{x}) = \frac{1}{\sigma_w^2} \sum_{n=0}^{N-1} \mathbb{E}\left[\mathbf{h}_n(z_n)\mathbf{h}_n^T(z_n) \right] \quad (35)$$

where the expectation can be approximated numerically using quadrature or Monte Carlo approximation. The jitter-augmented $\mathrm{CRB}_{\mathbf{y,z}}(\mathbf{x})$ is the trace of the inverse of this matrix. We will return to the question of the difference of the two CRBs later in Section VI, after we discuss ML estimation using an EM algorithm.

IV. LINEAR ESTIMATION

In this paper, an estimator is said to be linear if it is a linear function of the observations; such an estimator has the form

$$\hat{\mathbf{x}}_{\mathrm{L}}(\mathbf{y}) = \mathbf{A}\mathbf{y} \quad (36)$$

where the matrix $\mathbf{A}$ is fixed.[1]

For the semilinear observation model in (8), a linear estimator is unbiased if and only if $\mathbf{A}\mathbb{E}[\mathbf{H}(\mathbf{z})] = \mathbf{I}$. Since $\mathbb{E}[\mathbf{H}(\mathbf{z})]$ is a tall matrix, assuming it has full column rank, one possible linear unbiased estimator is

$$\hat{\mathbf{x}}_{\mathrm{L}}(\mathbf{y}) = \mathbb{E}[\mathbf{H}(\mathbf{z})]^\dagger \mathbf{y} \quad (37)$$

where $\mathbb{E}[\mathbf{H}(\mathbf{z})]^\dagger \triangleq \left(\mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbb{E}[\mathbf{H}(\mathbf{z})] \right)^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})]^T$ is the left pseudoinverse. This estimator is only one such linear unbiased estimator; more generally, any matrix that lies in the nullspace of $\mathbb{E}[\mathbf{H}(\mathbf{z})]$ can be added to the pseudoinverse and yield an unbiased estimator.

The question then remains of how to obtain the best linear unbiased estimator (BLUE), in the MMSE sense. In the context of a simple linear observation model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$, with Gaussian noise w, the BLUE is elementary to find (see [22]), and it is also the ML and efficient minimum variance unbiased estimator (MVUE). If we choose $\mathbf{z} = \mathbf{0}$ to be deterministic (no jitter) in the observation model, the corresponding BLUE/efficient estimator would be

$$\hat{\mathbf{x}}_{\mathrm{eff}|\mathbf{z}=\mathbf{0}}(\mathbf{y}) = \left(\mathbf{H}(0)^T \mathbf{H}(0) \right)^{-1} \mathbf{H}(0)^T \mathbf{y}. \quad (38)$$

[1]Sometimes, affine estimators $\hat{\mathbf{x}}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{b}$ are considered to be linear as well. However, as we will concern ourselves with unbiased estimators, $\mathbf{b}$ would turn out to be necessarily zero.
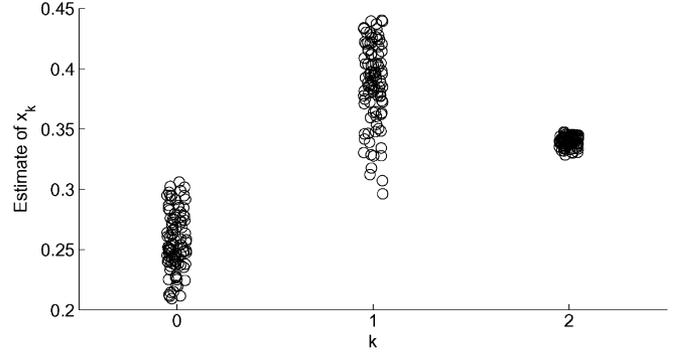

Fig. 3. The best linear unbiased estimator (BLUE) is computed for different choices of $\mathbf{x}$ (holding $\mathbf{y}$ fixed), using (39). In this example, $K = 3$, $M = 2$, and $\sigma_z = \sigma_w = 0.25$.

The performance of this estimator when applied to the proper (jittered) observation model will be used as one baseline for MSE improvement for the proposed estimators.

As derived previously in [14], the BLUE for the semilinear model (8) is

$$\hat{\mathbf{x}}_{\mathrm{BLUE}}(\mathbf{y}) = \left(\mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbf{\Lambda}_{\mathbf{y}}^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})] \right)^{-1} \mathbb{E}[\mathbf{H}(\mathbf{z})]^T \mathbf{\Lambda}_{\mathbf{y}}^{-1} \mathbf{y}$$
$$(39)$$

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

$$\mathbf{\Lambda}_{\mathbf{y}} = \mathbb{E}\left[\mathbf{H}(\mathbf{z})\mathbf{x}\mathbf{x}^T\mathbf{H}(\mathbf{z})^T \right] - \mathbb{E}[\mathbf{H}(\mathbf{z})]\mathbf{x}\mathbf{x}^{\mathbf{T}}\mathbb{E}[\mathbf{H}(\mathbf{z})]^T + \sigma_w^2 \mathbf{I}.$$
$$(40)$$

The BLUE estimator, in general, is not a valid estimator, since it depends on the true value of the unknown $\mathbf{x}$. Two sufficient conditions for the estimator to be valid are: $\mathbf{\Lambda}_{\mathbf{y}}$ is a scalar matrix, in which case, the covariance matrix commutes across multiplication, or $\mathbf{\Lambda}_{\mathbf{y}}$ does not depend on $\mathbf{x}$. Since z_m and z_n are independent for $m \neq n$, off-diagonal elements of $\mathbf{\Lambda}_{\mathbf{y}}$ are zero. For the covariance matrix to be a scalar matrix that commutes over matrix multiplication for all $\mathbf{x}$, $\mathrm{cov}(\mathbf{h}_n(z_n))$ must be equal for all n. However, this equality generally does not hold due to oversampling. Also, the covariance matrix clearly depends on $\mathbf{x}$ when $\mathrm{cov}(\mathbf{h}_n(z_n))$ is nonzero for some n. When the covariance matrix $\mathbf{\Lambda}_{\mathbf{y}}$ is a scalar matrix, the BLUE estimator is equal to $\hat{\mathbf{x}}_{\mathrm{L}}(\mathbf{y})$.

To conclusively demonstrate that the BLUE is not a valid estimator, the estimator is computed for a fixed value of $\mathbf{y}$ and varying $\mathbf{x}$; the results are shown in Fig. 3. Clearly, since the estimates of x_k vary depending on the value of $\mathbf{x}$ used in (39), the estimator is not valid. Thus, an MSE-optimal linear estimator does not exist for this problem, and we will utilize the estimator in (37).

V. ML ESTIMATION

Given a semilinear model as in (8), we would not expect the optimal MMSE estimator to have a linear form. Indeed, as shown in the previous section, for most signal models and priors on the jitter, the BLUE does not even exist. To improve upon linear estimation, and reduce the MSE, we move to maximum likelihood estimation.

Consider the problem of maximizing the likelihood function in (12); since the logarithm is an increasing function, we can perform the optimization by maximizing the log-likelihood:

$$\hat{\mathbf{x}}_{\mathrm{ML}}(\mathbf{y}) = \arg\max_{\mathbf{x}} \sum_{n=0}^{N-1} \log p(y_n; \mathbf{x}). \qquad (41)$$

However, since the marginal pdf does not have a closed form, and neither do its derivatives, performing the necessary optimization is difficult. Numerical techniques may be applied directly to (41), and various general-purpose methods have been studied extensively throughout the literature. An iterative joint maximization method proposed in [23] attempts to approximate the ML estimate by alternating between maximizing $p(\mathbf{z}|\mathbf{y}; \mathbf{x})$ with respect to $\mathbf{z}$ and $p(\mathbf{y}|\mathbf{z}; \mathbf{x})$ with respect to $\mathbf{x}$. One method that explicitly takes advantage of the special structure in (41) is the EM algorithm.

A. ML Estimation Using the EM Algorithm

Consider the function $Q(\mathbf{x}; \hat{\mathbf{x}}^{(i-1)})$ in (19). The expression for $\log p(y_n, z_n; \mathbf{x})$ is in (33), and summing these together (without the minus sign) gives

$$\log p(\mathbf{y}, \mathbf{z}; \mathbf{x}) = -\frac{1}{2\sigma_w^2} \|\mathbf{y} - \mathbf{H}(\mathbf{z})\mathbf{x}\|_2^2 - \frac{1}{2\sigma_z^2} \|\mathbf{z}\|_2^2 + \text{constants}. \qquad (42)$$

Expanding and substituting into the expectation in (19) yields

$$Q\left(\mathbf{x}; \hat{\mathbf{x}}^{(i-1)}\right) = \frac{-1}{2\sigma_w^2} \left(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T \mathbb{E}\left[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right] \mathbf{x} \right.$$
$$\left. + \mathbf{x}^T \mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right] \mathbf{x} \right)$$
$$- \frac{1}{2\sigma_z^2} \mathbb{E}\left[\mathbf{z}^T\mathbf{z}|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right] + \text{constants}. \qquad (43)$$

We want to find the value of $\mathbf{x}$ that maximizes this expression. Noticing that (43) is quadratic in $\mathbf{x}$, the candidate value $\mathbf{x}$ satisfies the linear system

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right] \mathbf{x} = \mathbb{E}\left[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right]^T \mathbf{y}. \qquad (44)$$

Also, the Hessian matrix is negative-definite, so (43) is strictly concave, and the candidate point $\mathbf{x}$ is the unique maximum $\hat{\mathbf{x}}^{(i)}$. All that remains to specify the EM algorithm is to approximate the expectations in (44).

Using Bayes' rule and the separability of both $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$ and $p(\mathbf{y}; \mathbf{x})$, the posterior distribution of the jitter is also separable:

$$p\left(\mathbf{z}|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right) = \prod_{n=0}^{N-1} \frac{p\left(y_n|z_n; \hat{\mathbf{x}}^{(i-1)}\right) p(z_n)}{p\left(y_n; \hat{\mathbf{x}}^{(i-1)}\right)}$$
$$= \prod_{n=0}^{N-1} p\left(z_n|y_n; \hat{\mathbf{x}}^{(i-1)}\right). \qquad (45)$$

Thus, the expectations are also separable into univariate expectations:

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right]$$
$$= \sum_{n=0}^{N-1} \mathbb{E}\left[\mathbf{h}_n(z_n)\mathbf{h}_n^T(z_n)|y_n; \hat{\mathbf{x}}^{(i-1)}\right]; \qquad (46)$$

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right]_{n,:}$$
$$= \mathbb{E}\left[\mathbf{h}_n^T(z_n)|y_n; \hat{\mathbf{x}}^{(i-1)}\right]. \qquad (47)$$

The subscript after the left-side expectation in (47) denotes the nth (zero-indexed) row of the matrix. The distribution $p(y_n; \hat{\mathbf{x}}^{(i-1)})$ is constant with respect to z_n, and can be evaluated using quadrature, as in (27). Approximating each of the univariate expectations in (46) and (47) with quadrature yields

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})^T\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right]$$
$$\approx \sum_{n=0}^{N-1} \frac{1}{p\left(y_n; \hat{\mathbf{x}}^{(i-1)}\right)} \sum_{j=1}^{J} w_j \mathbf{h}_n(z_j)\mathbf{h}_n^T(z_j) p\left(y_n|z_j; \hat{\mathbf{x}}^{(i-1)}\right); \qquad (48)$$

$$\mathbb{E}\left[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}\right]_{n,:}$$
$$\approx \frac{1}{p\left(y_n; \hat{\mathbf{x}}^{(i-1)}\right)} \sum_{j=1}^{J} w_j \mathbf{h}_n^T(z_j) p\left(y_n|z_j; \hat{\mathbf{x}}^{(i-1)}\right). \qquad (49)$$

The complexity of each iteration of this algorithm appears to be linear in the number of samples, although the rate of convergence (and thus, the number of iterations required) may also vary with the number of samples, or other factors. The convergence rate, as well as susceptibility to initial conditions (since the EM algorithm only guarantees local convergence), are the subject of simulations in this work and in [14].

The EM algorithm for ML estimation is summarized in Algorithm 1.

Algorithm 1: Pseudocode for the EM Algorithm for Computing ML Estimates for the Unknown Signal Parameters $\mathbf{x}$

Require: $p(\mathbf{y}, \mathbf{z}; \mathbf{x})$, $\mathbf{y}$, $\hat{\mathbf{x}}^{(0)}$, I, J, δ, ϵ

$\quad i \leftarrow 0$

$\quad$ Compute J-term quadrature rule (using e.g., the eigendecomposition method in [16]) for use in below approximations (use $J = 100$).

$\quad$ **repeat**

$\quad\quad i \leftarrow i + 1$

$\quad\quad$ **for** $n = 0$ to $N - 1$ **do**

$\quad\quad\quad$ Approximate $p(y_n; \hat{\mathbf{x}}^{(i-1)})$ using (27).

$\quad\quad\quad$ Compute $\mathbb{E}[\mathbf{h}_n(z_n)\mathbf{h}_n^T(z_n)|y_n; \hat{\mathbf{x}}^{(i-1)}]$ using (48).

$\quad\quad\quad$ Approximate $\mathbb{E}[\mathbf{H}(\mathbf{z})|\mathbf{y}; \hat{\mathbf{x}}^{(i-1)}]_{n,:}$ using (49).

$\quad\quad$ **end for**

$\quad\quad$ Solve for $\hat{\mathbf{x}}^{(i)}$ using (44) and the above approximations.

$\quad$ **until** $i = I$ **or** $\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(i-1)}\|_2 < \delta$ **or** $|\log \ell(\hat{\mathbf{x}}^{(i)}; \mathbf{y}) - \log \ell(\hat{\mathbf{x}}^{(i-1)}; \mathbf{y})| < \epsilon$

$\quad$ **return** $\hat{\mathbf{x}}^{(i)}$

VI. SIMULATION RESULTS

The objectives of the simulations presented here are to a) analyze the behavior of the proposed EM algorithm for approximating the ML estimator, and to b) compare the performance of this estimator to both the Cramér–Rao bound and that of linear
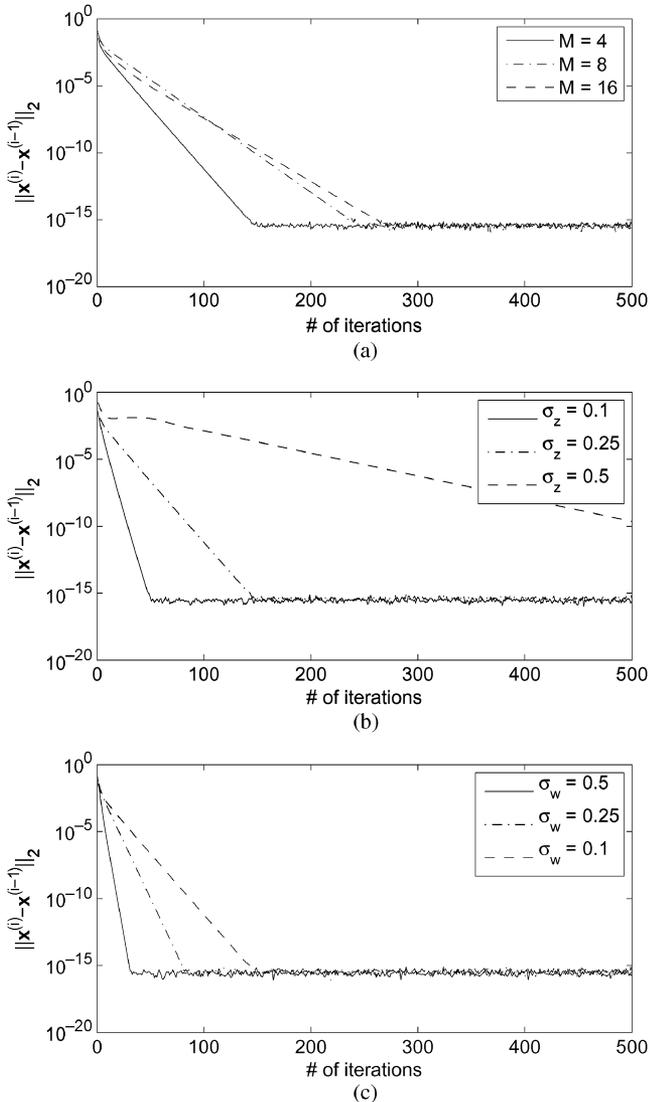
Fig. 4. The convergence of the EM algorithm depends on the choice of parameters M, σ_z, and σ_w, as demonstrated in the above plots. (a) $K = 10$, $\sigma_z = 0.25$, $\sigma_w = 0.1$, M varies. (b) $K = 10$, $M = 4$, $\sigma_w = 0.1$, σ_z varies. (c) $K = 10$, $M = 4$, $\sigma_z = 0.25$, σ_w varies.
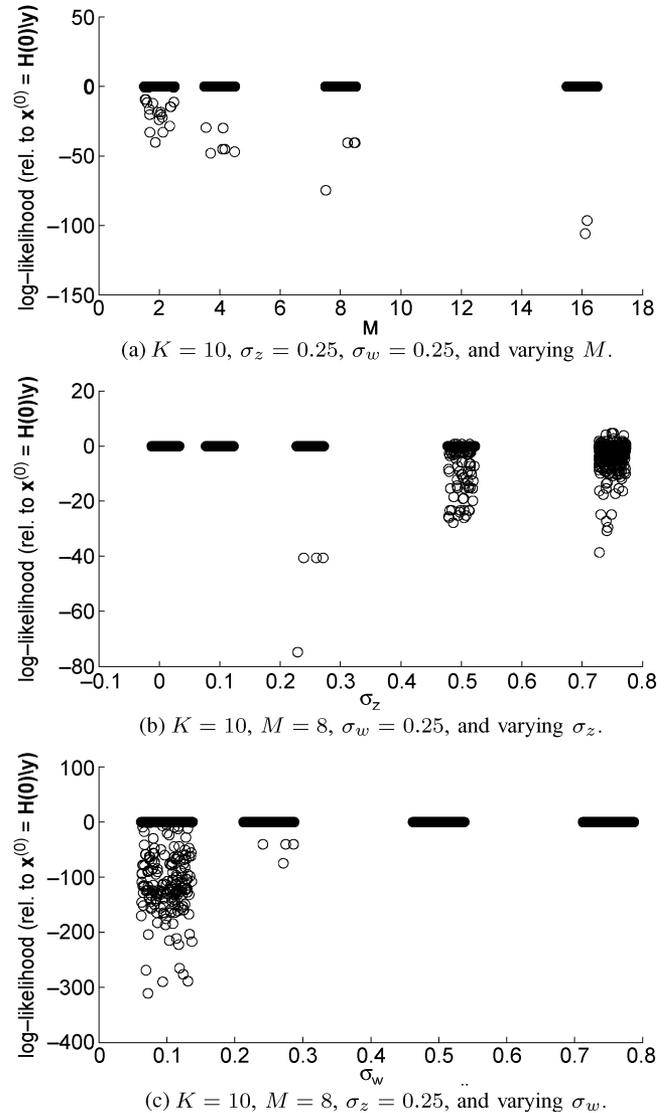

Fig. 5. The effects of varying initial conditions of the EM algorithm as a function of (a) oversampling factor, (b) jitter variance, and (c) additive noise variance are studied by computing the log-likelihoods of the EM algorithm results, for multiple initial conditions, across 50 trials. The log-likelihood of the EM algorithm results are displayed relative to the result for zero-jitter initialization, so that the log-likelihood of the result for $\hat{\mathbf{x}}^{(0)} = \mathbf{H}(\mathbf{0})^\dagger \mathbf{y}$ is zero. (a) $K = 10$, $\sigma_z = 0.25$, $\sigma_w = 0.25$, and varying M. (b) $K = 10$, $M = 8$, $\sigma_w = 0.25$, and varying σ_z. (c) $K = 10$, $M = 8$, $\sigma_z = 0.25$, and varying σ_w.

parameter estimation. The convergence behavior is studied in detail in [14] for periodic bandlimited signals. In this work, experiments to determine convergence behavior and sensitivity to initial conditions are conducted for the sinc basis signal model described in Section I. In all experiments, we utilize Matlab to generate signals with pseudo-random parameters and noise and apply the algorithms described to the samples of these signals. For a factor of M oversampling, we generate $N = KM$ samples.

A. Convergence Analysis

While guaranteed to converge, the EM algorithm would be of little use if it did not converge quickly. The rate of convergence of the EM algorithm is studied for several choices of M, σ_z, and σ_w, and trends are presented in Fig. 4. The rate of convergence is exponential, and the rate decreases with increasing M, increasing σ_z, and decreasing σ_w.

As mentioned in Section II, the rate of convergence of the EM algorithm is related to the difference between the CRBs of

the complete and the incomplete data. As shown later in Fig. 6, the difference between the CRBs for the complete data and incomplete data increases exponentially with σ_z. This relationship coincides with the convergence behavior observed in Fig. 4(b). Although these experiments evaluated 500 iterations of the EM algorithm, the results suggest that 100 iterations would suffice as long as the jitter standard deviation σ_z is not too large. Also, 10^{-8} is chosen as a reasonable stopping criterion for change in $\mathbf{x}^{(i)}$ and change in log-likelihood between iterations (δ and ϵ in Algorithm 1, respectively).

B. Sensitivity to Initial Conditions

The likelihood function described in (11) is generally non-concave, so maximizing the function via a hill-climbing method
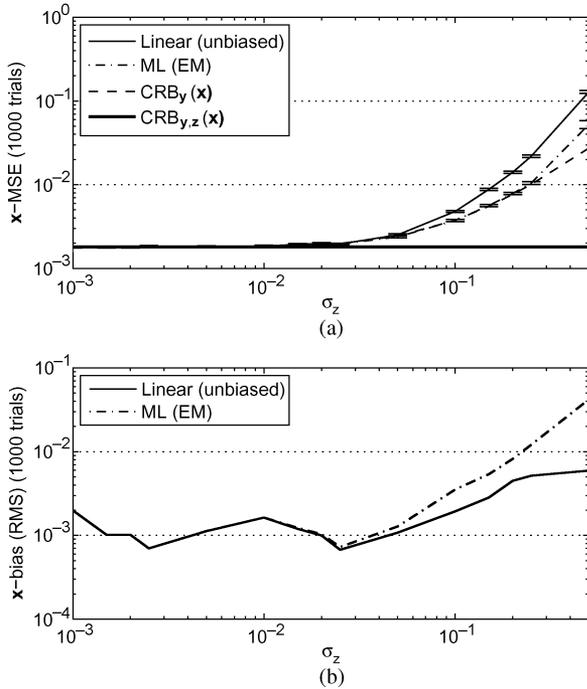
Fig. 6. The approximate performances of the linear unbiased estimator and ML estimator (EM algorithm) are plotted in (a) against the $\mathrm{CRB}_{\mathbf{y}}(\mathbf{x})$ and the complete data $\mathrm{CRB}_{\mathbf{y},\mathbf{z}}(\mathbf{x})$, as a function of σ_z for $K = 10$, $M = 16$, $\sigma_w = 0.05$, and a fixed random choice of $\mathbf{x}$. The linear and ML estimator biases are plotted in (b), using the root-mean-squared (RMS) values of the bias vectors. The bars above and below each data point for the linear and ML estimators in (a) delineate the 95% confidence intervals for those data points.

like the EM algorithm is only guaranteed to yield a local maximum. The ability of the algorithm to converge to the global maximum depends on the nonconcavity of the likelihood function. To demonstrate the sensitivity of the EM algorithm, as a function of M, σ_z, and σ_w, the empirical distribution of the log-likelihood of the optimal values reached from multiple initial conditions is evaluated over numerous trials for different choices of these parameters. In this experiment, the true value of $\mathbf{x}$, the no-jitter linear estimator (38), $\mathbf{x} = \mathbf{0}$, and ten random choices, are used as initial conditions for each trial. As suggested by the spread of the samples shown in Fig. 5, the variability of the EM algorithm increases with σ_z and decreasing σ_w. Even when the EM algorithm appears sensitive to initial conditions, using the no-jitter linear estimate (38) results in a relatively small deviation from the best observed log-likelihood value. In situations when such initialization does fail to produce consistent results, methods such as the deterministic annealing EM algorithm described in [24] may improve consistency.

C. Performance of the EM Algorithm

In the first performance experiment, the Cramér–Rao lower bound is compared to the unbiased linear estimator (37) and the EM algorithm of the ML estimator to measure the efficiency of the algorithms. The Cramér–Rao lower bound for the complete data is also presented for reference. Although computational difficulties prevent a complete comparison for every possible value of $\mathbf{x}$, carrying out a comparison for a few randomly chosen values of $\mathbf{x}$ provide a measure of the quality of the algorithms.
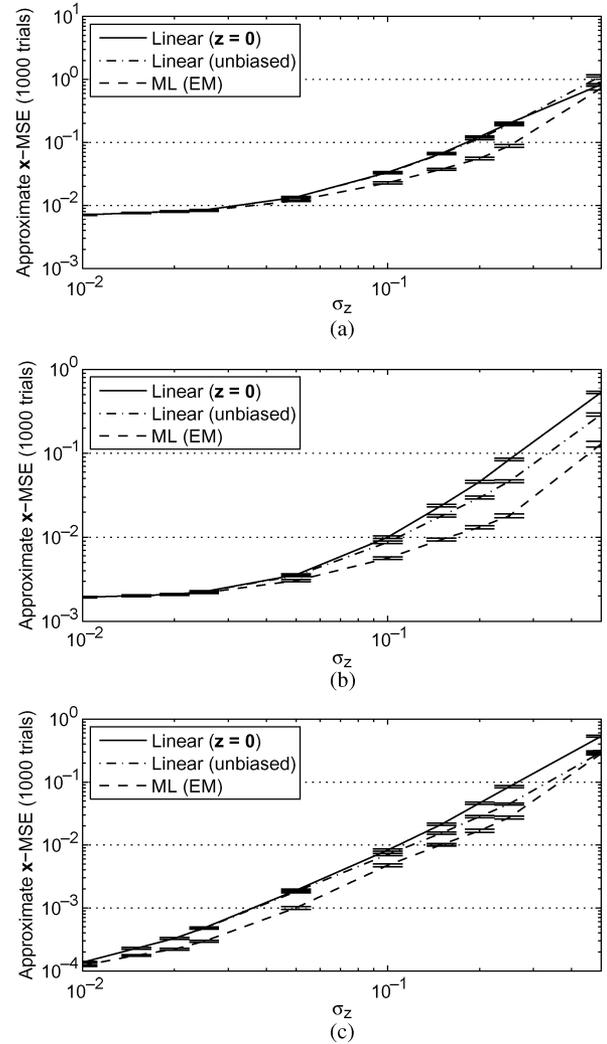

Fig. 7. The MSE performance of the ML estimator (EM algorithm) is compared against both the unbiased linear estimator (37) and the no-jitter BLUE (38), as a function of σ_z. The bars above and below each data point for the linear and ML estimators delineate the 95% confidence intervals for those data points. (a) $K = 10$, $M = 4$, $0.01 \leq \sigma_z \leq 0.5$, $\sigma_w = 0.05$. (b) $K = 10$, $M = 16$, $0.01 \leq \sigma_z \leq 0.5$, $\sigma_w = 0.05$. (c) $K = 10$, $M = 16$, $0.01 \leq \sigma_z \leq 0.5$, $\sigma_w = 0.01$.

As the curves in Fig. 6 demonstrate for one such random choice of $\mathbf{x}$, both algorithms are approximately efficient for small σ_z, but the EM algorithm continues to be efficient for larger values of σ_z than the linear estimator. In addition, the bias shown for the linear and ML estimators is approximately the same for $\sigma_z < \sigma_w$; this bias may be due to the small error in the numerical integration. Note how this error becomes larger with σ_z.

In Fig. 7, the EM algorithm is compared against two linear estimators. First, to demonstrate the MSE improvement attainable through nonlinear estimation, the EM algorithm is pitted against the linear unbiased estimator. Since a major motivating factor for developing these algorithms is to reduce the power consumption due to clock accuracy, the EM algorithm also can achieve the same MSE as the linear estimator for a substantially larger jitter variance, reducing the clock's power consumption.

When the additive noise dominates the jitter ($\sigma_z \ll \sigma_w$), the improvement can be expected to be minimal, since the system
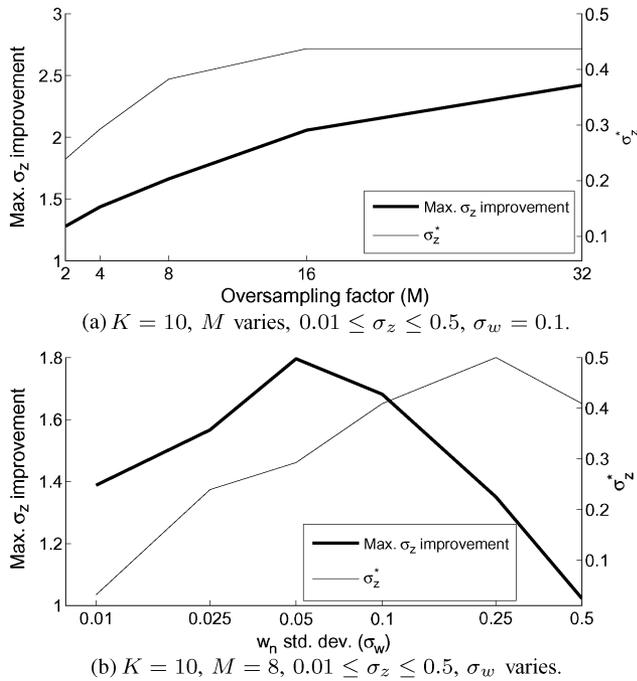
Fig. 8. These graphs show the maximum factor of improvement in jitter tolerance, measured by σ_z, achievable by the EM algorithm (relative to linear reconstruction). Holding σ_w fixed, (a) shows the trend in maximum improvement as M increases, and (b) shows the trend in maximum improvement as σ_w increases while holding M fixed. The jitter standard deviation σ_z^* corresponding to this maximum improvement for the ML estimator is plotted on the same axes. (a) $K = 10$, M varies, $0.01 \leq \sigma_z \leq 0.5$, $\sigma_w = 0.1$. (b) $K = 10$, $M = 8$, $0.01 \leq \sigma_z \leq 0.5$, σ_w varies.

is nearly linear, and the jitter is statistically insignificant. As the amount of jitter increases, the density function $p(\mathbf{z}|\mathbf{y}; \mathbf{x})$ used in each iteration of the EM algorithm becomes more nonlinear in $\mathbf{z}$, and the quadrature becomes less accurate for a given number of terms. Therefore, the EM algorithm generally takes longer to converge, and the result should be a less accurate approximation to the true ML estimator. This behavior is observed in Fig. 7, where the EM algorithm is compared against both the linear unbiased estimator and the no-jitter linear estimator. The EM algorithm generally has lower MSE than either linear estimator, and the performance gap is more pronounced for higher oversampling factor M.

To answer the question of how much more jitter can be tolerated for the same desired MSE using the EM algorithm, the maximum proportional increase is plotted as a function of M and σ_w in Fig. 8. The maximum proportional increase for a choice of M and σ_w is computed by approximating log-log domain MSE curves, like those in Fig. 7, with piece-wise linear curves and interpolating the maximum distance between them over the range of $\sigma_z \geq \sigma_w$. The range of $\sigma_z < \sigma_w$ is ignored since the linear and nonlinear reconstructions perform similarly when the additive noise dominates (as expected). The proportion of improvement increases linearly as M increases. As σ_w increases, the level of improvement stays approximately the same for $\sigma_w < \sigma_z$. However, when σ_w increases beyond σ_z, the level of improvement decreases substantially as expected, since the additive noise dominates, and the optimal estimator is approxi-

mately linear. A maximum σ_z improvement factor of two corresponds to power savings of up to 75 percent.

VII. CONCLUSION

The results presented in Section VI are very encouraging from a power-consumption standpoint. A maximum improvement of between 1.4 to 2 times the jitter translates to a two-to-fourfold decrease in power consumption by the clock, according to (3). To put the magnitude of such an improvement in context, consider the digital baseband processor for ultra-wideband communication in [25]. This processor incorporates an ADC and a PLL, which consume 86 and 45 mW, respectively, out of a 271 mW budget for the chip. Reducing by a factor of two the power consumed by the ADC alone would decrease the total power consumption of the chip by almost sixteen percent.

While effective, the EM algorithm is computationally expensive. One benefit of digital postprocessing is that these algorithms can be performed off-chip, on a computer or other system with less limited computational resources. For real-time on-chip applications, Kalman filter-like versions of the EM algorithm would be more practical; this extension is a topic for further investigation. Related to real-time processing is developing streaming algorithms for the infinite-dimensional case, extending this work for general real-time sampling systems. Another future direction involves modifying these algorithms for correlated or periodic jitter.

Sampling jitter mitigation is actually just one application of these new algorithms. In the frequency domain, jitter maps to uncertainty in frequency; using algorithms such as these should produce more reliable Fourier transforms for systems like spectrum analyzers. In higher dimensions, timing noise becomes location jitter in images or video. Greater tolerance of the locations of pixels in images would allow scanning electron microscope users to acquire higher resolution images without sacrificing MSE. This paper shows that significant improvements over the best linear postprocessing are possible; thus, further work may impact these and other applications.

REFERENCES

[1] I. King, "Understanding high-speed signals, clocks, and data capture," *Nat. Semiconductor Signal Path Designer*, vol. 103, pp. 1–7, Apr. 2005.
[2] T. H. Lee and A. Hajimiri, "Oscillator phase noise: A tutorial," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 326–336, Mar. 2000.
[3] K. Uyttenhove and M. S. J. Steyaert, "Speed-power-accuracy tradeoff in high-speed CMOS ADCs," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 4, pp. 280–287, Apr. 2002.
[4] B. Brannon, "Aperture uncertainty and ADC system performance," Analog Devices, Tech. Rep. AN-501, Sep. 2000.
[5] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.
[6] Y. Lee and R. Mittra, "Electromagnetic interference mitigation by using a spread-spectrum approach," *IEEE Trans. Electromagn. Compat.*, vol. 44, no. 2, pp. 380–385, May 2002.

[7] A. V. Balakrishnan, "On the problem of time jitter in sampling," *IRE Trans. Inf. Theory*, vol. 8, no. 3, pp. 226–236, Apr. 1962.

[8] W. M. Brown, "Sampling with random jitter," *J. Soc. Indust. Appl. Math.*, vol. 11, no. 2, pp. 460–473, Jun. 1963.

[9] B. Liu and T. P. Stanley, "Error bounds for jittered sampling," *IEEE Trans. Autom. Control*, vol. 10, no. 4, pp. 449–454, Oct. 1965.

[10] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, "Efficient numerical methods in non-uniform sampling theory," *Numer. Math.*, vol. 69, no. 4, pp. 423–440, Feb. 1995.

[11] P. Marziliano and M. Vetterli, "Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3462–3471, Dec. 2000.

[12] T. E. Tuncer, "Block-based methods for the reconstruction of finite-length signals from nonuniform samples," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 530–541, Feb. 2007.

[13] V. Divi and G. Wornell, "Signal recovery in time-interleaved analog-to-digital converters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, May 2004, vol. 2, pp. 593–596.

[14] D. S. Weller, "Mitigating timing noise in ADCs through digital post-processing," SM Thesis, Electrical Eng. Comput. Sci. Dept., Massachusetts Inst. of Technology, Cambridge, MA, Jun. 2008.

[15] P. K. Kythe and M. R. Schäferkotter, *Handbook of Computational Methods for Integration*. Boca Raton, FL: CRC, 2005.

[16] G. H. Golub and J. H. Welsch, "Calculation of Gauss quadrature rules," *Math. Comp.*, vol. 23, no. 106, pp. 221–230, Apr. 1969.

[17] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*. Orlando, FL: Academic, 1984.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[19] C. Herzet and L. Vandendorpe, "Prediction of the EM-algorithm speed of convergence with Cramer-Rao bounds," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Apr. 2007, vol. 3, pp. 805–808.

[20] X.-L. Meng and D. B. Rubin, "Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm," *J. Amer. Statist. Assoc.*, vol. 86, no. 416, pp. 899–909, Dec. 1991.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[22] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Prentice-Hall Signal Processing Series. Upper Saddle River, NJ: Prentice-Hall, 1993, vol. 1.

[23] J. Kusuma and V. K. Goyal, "Delay estimation in the presence of timing noise," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 9, pp. 848–852, Sep. 2008.

[24] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Netw.*, vol. 11, no. 2, pp. 271–282, Mar. 1998.

[25] 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*, vol. 40, no. 9, pp. 1821–1828, Sep. 2005.

Daniel S. Weller (S'05) received the B.S. degree in electrical and computer engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2006 and the S.M. degree in electrical engineering and computer science (EECS) from the Massachusetts Institute of Technology (MIT), Cambridge, in 2008. He is working towards the Ph.D. degree in EECS with the Signal Transformation and Information Representation Group, Research Laboratory of Electronics, MIT.

He was an undergraduate research assistant with the General Motors Collaborative Research Laboratory at CMU and has completed summer internships at Texas Instruments, Apple, and ATK. His current research interests include compressed sensing, magnetic resonance imaging, and applications of estimation theory in sampling and signal processing.

Mr. Weller is a member of Tau Beta Pi, Eta Kappa Nu, and SIAM. He is the recipient of a National Defense Science and Engineering Graduate (NDSEG) fellowship and a graduate research fellowship from the National Science Foundation (NSF).

Vivek K Goyal (S'92–M'98–SM'03) received the B.S. degree in mathematics and the B.S.E. degree in electrical engineering (both with highest distinction) from the University of Iowa, Iowa City. He received the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1995 and 1998, respectively.

He was a Member of Technical Staff in the Mathematics of Communications Research Department of Bell Laboratories, Lucent Technologies, from 1998 to 2001 and a Senior Research Engineer for Digital Fountain, Inc. from 2001 to 2003. He is currently Esther and Harold E. Edgerton Associate Professor of electrical engineering at the Massachusetts Institute of Technology, Cambridge. His research interests include source coding theory, sampling, quantization, and information gathering and dispersal in networks.

Dr. Goyal is a member of Phi Beta Kappa, Tau Beta Pi, Sigma Xi, Eta Kappa Nu, and SIAM. In 1998, he received the Eliahu Jury Award of the University of California, Berkeley, awarded to a graduate student or recent alumnus for outstanding achievement in systems, communications, control, or signal processing. He was also awarded the 2002 IEEE Signal Processing Society Magazine Award and an NSF CAREER Award. He served on the IEEE Signal Processing Society's Image and Multiple Dimensional Signal Processing Technical Committee. He is a Technical Program Committee Co-Chair of IEEE ICIP 2016 and a permanent Conference Co-Chair of the SPIE Wavelets conference series.