On the scalability of joint channel and mismatch estimation for time-interleaved analog-to-digital conversion in communication receivers

Sandeep Ponnuru and Upamanyu Madhow [1]

***Abstract*—The analog-to-digital converter (ADC) represents a fundamental bottleneck in power- and cost-efficient realizations of "mostly digital" communication transceivers at multi-Gigabit speeds. Time-interleaved (TI) ADCs, with slower, power-efficient "sub-ADCs" employed in parallel to obtain a high-rate ADC, represent a potential solution to this bottleneck. The performance floor caused by mismatch in TI-ADCs can be eliminated by estimating and correcting for the mismatch. For TI-ADCs employed in communication receivers, prior work has shown that mismatch and channel parameters can be estimated jointly by using the training available in communication systems. In this paper, we propose simplified algorithms for this purpose, and examine how well they scale as the number of sub-ADCs gets large. We conclude that rapid convergence can be attained if the training sequence scales with the number of sub-ADCs, and that the convergence rate can be significantly enhanced by a suitable choice of periodic training sequence.**

I. INTRODUCTION

Modern communication transceivers exploit the availability of exponentially increasing computational power (Moore's law) by employing "mostly digital" architectures, in which the bulk of the signal processing is performed in the digital domain, followed by digital-to-analog converters at the transmitter and analog-to-digital converters at the receiver. The economies of scale thus provided have propelled the growth of mass market digital communication systems such as cellular networks, wireless local area networks, and broadband DSL and cable modems. In order to scale this approach to provide low-cost solutions for emerging multi-Gigabit communication systems (e.g., 60 GHz wireless, optical interconnects), we must address the analog-to-digital converter (ADC) bottleneck [1]: high-speed ADCs implemented in CMOS technology either have limited resolution or excessive power dissipation [2], [3]. An attractive design approach for obtaining high-rate ADCs with reduced power consumption is to employ a time-interleaved architecture, in which several low-rate ADCs (termed "sub-ADCs" here) are employed in parallel with staggered sampling times. The lower sampling rates for the sub-ADCs means that power-efficient architectures as successive approximation or pipelined can be employed, compared to the flash architectures required for directly implementing a high-speed ADC. However, the limiting factor in the performance of such a time-interleaved ADC (TI-ADC) is the mismatch between the sub-ADCs. When used in a communication receiver, for example, uncompensated mismatch leads to error floors. The error floors can be removed by estimating and compensating for the mismatch parameters, and we have shown in earlier work [5] that it is possible to jointly estimate the mismatch and the channel using the training sequences already available in communication systems. In this paper, we ask how well such mismatch estimation procedures scale, in terms of the length of training required, as we increase the number of sub-ADCs.

Contributions

Our starting point is our prior work [5], where we present an iterative algorithm for joint estimation of the channel coefficients and TI-ADC mismatch parameters (modeled as gain, timing and voltage-offset mismatches). A least squares cost function is alternately minimized over the channel and mismatch parameters, keeping the other set of parameters fixed. In this scheme, channel estimation with fixed mismatch parameters has a closed form solution, while mismatch estimation with fixed channel coefficients requires L one-dimensional searches, where L denotes the number of interleaved ADCs.

In this paper, we simplify the iterative algorithm in [5] by using a linear approximation to model the timing mismatch. This leads to closed form solutions for both the channel and mismatch estimation steps. We then analyze the convergence and robustness to noise of the estimation procedure. We first consider estimation performance with pseudorandom training sequences, as a function of the training sequence length M, the number of sub-ADCs L, and the number of channel coefficients N. When M is a large enough multiple of L (e.g., $M = 4L$), we observe that the estimation error decreases exponentially with iterations, with a decay rate that increases with M (for a fixed L) or decreases with L (for fixed M). However, when the training sequence length $M = 2L$ or smaller, we provide some examples showing that the iterative algorithm can get stuck away from the true parameter values. Finally, we evaluate the Cramer-Rao lower bounds for joint channel and mismatch estimation, and observe that the mean-squared error for our estimates is close to these bounds.

We next consider design of training sequences for speeding up convergence of the estimation algorithm. We propose a training sequence for which it is proved that the channel and mismatch parameters can be well estimated within the first iteration.

A. Related literature

There is a significant literature on estimating gain and timing mismatches for TI-ADCs [4]-[12]. Blind approaches [6]-[9] typically rely on the knowledge of signal statistics,

[1]The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara CA, 93117 USA. E-mail: madhow@ece.ucsb.edu. This research was supported in part by the National Science Foundation under grant CCF-0729222.

while training-based approaches [10]-[11] use sinusoidal or pseudorandom training. However, the number of sub-ADCs considered is usually small ($L < 16$). In contrast to the literature on mismatch estimation for generic TI-ADC applications, we focus on joint mismatch and channel estimation for the use of TI-ADCs in communication applications, as in our prior work [5], or as in [12]. This allows us to leverage training sequences already available in communication systems, and potentially eliminates the need for dedicated hardware for mismatch estimation. While our prior work demonstrates the feasibility of this approach [5], the focus of the present paper is on exploring how well the approach scales as we increase the number of sub-ADCs.

II. System Model

As shown in Fig. 1, we consider transmission over a dispersive channel. We denote by $h(t)$ the impulse response of the cascade of the transmit filter, the channel and the receive filter. The transmitter sends a training sequence of M symbol-rate samples, extending over a time interval of length MT_0, where T_0 denotes the symbol period. We assume that the impulse response $h(t)$ has support in the interval $[0,(N-1)T_0]$. We can therefore write the received analog signal $r(t)$ for the duration of a single training sequence($t \in [0,(M-1)T_0]$) as follows:

$$r(t) = \sum_{t-(N-1)T_0 \le kT_0 \le t} b[k]h(t-kT_0)+n(t), \qquad k \in \mathbb{Z} \tag{1}$$

where $b[k]$ denotes the symbol transmitted at time kT_0 and $n(t)$ denotes receiver thermal noise. The receiver performs symbol-rate sampling of $r(t)$ in (1) using a TI-ADC (see Fig. 1). The sub-ADCs are indexed by $i = 0, 1, \cdots, L-1$, such that the sub-ADC with index i digitizes samples $r(mT_0)$ such that $m \bmod L = i$. In practice, sampling the I- and Q-channels requires two different TI-ADCs, which potentially can have different sets of mismatch parameters. In order to simplify the exposition, we assume that both I and Q TI-ADCs have the same mismatch parameters. We can now write the received samples as follows:

$$r[m] = (1+g_i)\sum_{k=m-(N-1)}^{m} b[k]h((m-k+\delta_i)T_0)+n[m], \tag{2}$$

where $i = m \bmod L$, $r[m] = r(mT_0)$, $n[m] = n(mT_0)$, and g_i, δ_i represent the (normalized) gain and timing mismatches, respectively, for the i^{th} sub-ADC. From (2), we observe that the mean values of the gain and timing mismatch, denoted by $\overline{g}$ and $\overline{\delta}$, can be absorbed into the channel by replacing $h(t)$ by $(1+\overline{g})h(t+\overline{\delta}T_0)$. We therefore assume, without loss of generality, that the mismatch parameters have zero mean:

$$\sum_{i=0}^{L-1} g_i = \sum_{i=0}^{L-1} \delta_i = 0 \tag{3}$$

We assume that there is no excess bandwidth in the transmission (a good approximation for OFDM, for example), so that the transmit filter is band-limited to $[-\frac{1}{2T_0}, \frac{1}{2T_0}]$. This implies that $h(t)$ is also band limited to the same range and hence, we can use the sampling theorem to write $h(t)$ in terms of (the symbol-rate samples) $h[q]$ as follows:

$$h(t) = \sum_{q=0}^{N-1} h[q] \text{ sinc } (\frac{t}{T_0} - q), \tag{4}$$

where we used the assumption that $h[q] = h(qT_0)$ is zero unless q lies between 0 and $N-1$. We now substitute $h(t)$ from (4) in (2) and collect all samples corresponding to the i^{th} sub-ADC (i.e., $\{r[m]\}$ for $m = i+pL$) into a vector $\boldsymbol{r}_i$, modeled as follows:

$$\boldsymbol{r}_i = \boldsymbol{C}_i(g_i,\delta_i)\boldsymbol{h} + \boldsymbol{n}_i \tag{5}$$

where $\boldsymbol{h}$ and $\boldsymbol{n}_i$ denote the vector of N channel coefficients $\{h[q]\}$ and the vector of noise samples respectively. The matrix $\boldsymbol{C}_i$ is a function of mismatch parameters, with $(p,q)^{\text{th}}$ element given by

$$\begin{aligned}[\boldsymbol{C}_i]_{(p,q)} &= (1+g_i) \\ &\times \sum_{k=i+pL-(N-1)}^{i+pL} b[k] \text{ sinc } (i+pL-k-q+\delta_i)\end{aligned} \tag{6}$$

We now describe a linear approximation to model timing mismatch, where the samples of the sinc function are approximated as follows:

$$\text{sinc } (k+\delta) = \begin{cases} 1 & k=0 \\ \delta \cdot \text{ sinc}'(k) & k \ne 0, k \in \mathbb{Z} \end{cases} \tag{7}$$

where sinc$'$ represents the derivative of the sinc function. As shown in our simulations later, this is a good approximation (in the least squares sense) as long as the timing mismatches (relative to T) are small ($< 10\%$). Using (7) in (6), we can decompose the matrix $\boldsymbol{C}_i$ as

$$\boldsymbol{C}_i = (1+g_i)\boldsymbol{A}_i + \tilde{\delta}_i\boldsymbol{B}_i \tag{8}$$

where $\tilde{\delta}_i = (1+g_i)\delta_i$. The elements of the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are given by

$$\begin{aligned}[\boldsymbol{A}_i]_{(p,q)} &= b(i+pL-q), \\ [\boldsymbol{B}_i]_{(p,q)} &= \sum_{\substack{k = i+pL-(N-1) \\ k \ne i+pL-q}}^{i+pL} b[k] \text{ sinc}'(i+pL-q-k)\end{aligned} \tag{9}$$

Our aim is to jointly estimate the unknown channel coefficients $\boldsymbol{h}$ and the gain and timing mismatch parameters corresponding to all sub-ADCs. Using (5) and (8), we can write the joint estimation problem as,

$$\begin{aligned}&(\hat{\boldsymbol{h}}, \{\hat{g}_0, \cdots, \hat{g}_{L-1}\}, \{\hat{\delta}_0, \cdots, \hat{\delta}_{L-1}\}) \\ &= \arg\min \sum_{i=0}^{L-1} ||\boldsymbol{r}_i - (1+g_i)\boldsymbol{A}_i\boldsymbol{h} - \tilde{\delta}_i\boldsymbol{B}_i\boldsymbol{h}||^2\end{aligned} \tag{10}$$

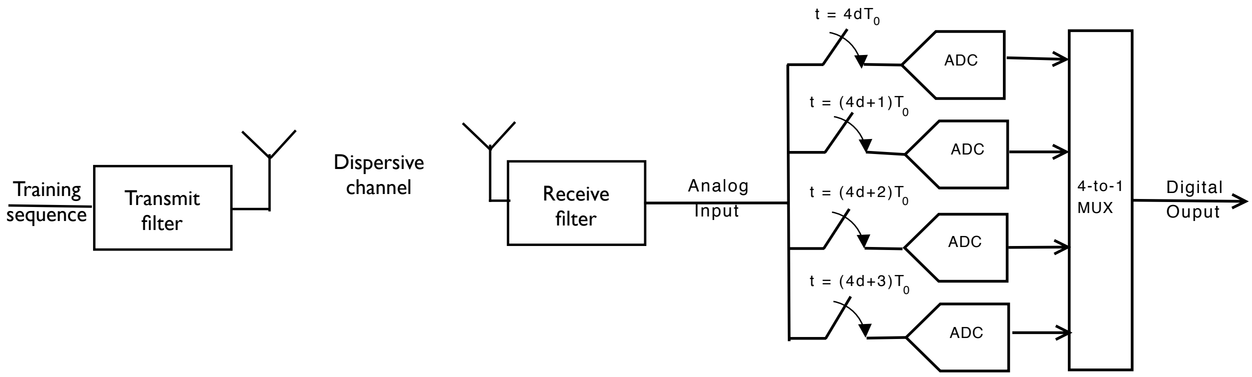

Fig. 1: Base-band model for transmission over dispersive channel using a time-interleaved ADC with 4 sub-ADCs at the receiver (d =integer, T_0 = sampling period).

where we assume that the noise samples are i.i.d. While this is not strictly true if the samples are not equi-spaced, it is a good approximation as long as the relative mismatches are small ($< 10\%$); see [5] for more discussion.

III. Alternating minimization algorithm

Our numerical experiments show that the objective function in (10) is non-convex, because the Hessian matrix [13] has positive as well as negative eigenvalues. Thus, the powerful tools of convex optimization cannot be applied (except as an approximate solution to the problem). Direct search is also not a valid option, due to the increase in the dimensionality of the problem (10) when either L or N become large. However, we observe from (10) that the estimation problem is quadratic in the channel parameters when the mismatch parameters are fixed, and vice versa, which means that there is a closed form solution for each step of alternating minimization. This simplification is possible due to the linear approximation in (7). It is worth noting that the mismatch estimation problem (with channel fixed) is quadratic in $\tilde{\delta}_i$, rather than in the actual timing mismatches δ_i. Thus, we first estimate g_i and $\tilde{\delta}_i$ in closed form, and then estimate δ_i. We can now specify each minimization step explicitly as follows:

- **Channel estimation given mismatches**: Given mismatch estimates $\{\hat{g}_0, \cdots, \hat{g}_{L-1}\}$ and $\{\hat{\delta}_0, \cdots, \hat{\delta}_{L-1}\}$, the channel estimate is given using (5) by
$$\hat{\boldsymbol{h}} = \boldsymbol{C}^{\dagger}\boldsymbol{r} \tag{11}$$
where the matrix $\boldsymbol{C}$ and $\boldsymbol{r}$ are formed by vertically concatenating, for $i = 0, 1, \cdots, L-1$, the matrices $\boldsymbol{C}_i$ and the vectors $\boldsymbol{r}_i$, respectively. The matrix $\boldsymbol{C}^{\dagger}$ is the pseudoinverse of $\boldsymbol{C}$.
- **Mismatch estimation given channel**: Since the mismatch parameters are real-valued, it is convenient to expand the complex-valued received vector (5) as a real-valued vector. Using (8) in (5), we can write
$$\begin{pmatrix} \Re[\boldsymbol{r}_i] \\ \Im[\boldsymbol{r}_i] \end{pmatrix} = \begin{pmatrix} \Re[\boldsymbol{u}] & \Re[\boldsymbol{v}] \\ \Im[\boldsymbol{u}] & \Im[\boldsymbol{v}] \end{pmatrix} \begin{pmatrix} g_i \\ \tilde{\delta}_i \end{pmatrix} + \begin{pmatrix} \Re[\boldsymbol{n}_i] \\ \Im[\boldsymbol{n}_i] \end{pmatrix} \tag{12}$$
where $\boldsymbol{u} = \boldsymbol{A}_i\boldsymbol{h}$, $\boldsymbol{v} = \boldsymbol{B}_i\boldsymbol{h}$ and $\Re[\cdot]$, $\Im[\cdot]$ denote the real and imaginary parts of a complex number respectively. Given the channel estimate $\hat{\boldsymbol{h}}$, the estimates of g_i and $\tilde{\delta}_i$ are obtained from (12) as follows:
$$\begin{pmatrix} \hat{g}_i \\ \hat{\tilde{\delta}}_i \end{pmatrix} = \begin{pmatrix} \Re[\hat{\boldsymbol{u}}] & \Re[\hat{\boldsymbol{v}}] \\ \Im[\hat{\boldsymbol{u}}] & \Im[\hat{\boldsymbol{v}}] \end{pmatrix}^{\dagger} \begin{pmatrix} \Re[\boldsymbol{r}_i] \\ \Im[\boldsymbol{r}_i] \end{pmatrix} \tag{13}$$
where $\hat{\boldsymbol{u}} = \boldsymbol{A}_i\hat{\boldsymbol{h}}$, $\hat{\boldsymbol{v}} = \boldsymbol{B}_i\hat{\boldsymbol{h}}$. Using (13), we can find the estimate for the timing mismatch parameter as $\hat{\delta}_i = \hat{\tilde{\delta}}_i/(1 + \hat{g}_i)$.

We now discuss the convergence and estimation error for the joint estimation algorithm (11)-(13).

IV. Joint estimation with pseudorandom training

In this section, we use a pseudorandom training sequence to estimate the channel and the TI-ADC mismatches. In our numerical results, we consider an m-sequence with generator polynomial $z^8 + z^6 + z^5 + z^4 + 1$. We append a zero to this length 255 m-sequence to obtain a training sequence of length 256 (an integer number of bytes). When we desire to use a smaller training sequence length, we simply truncate the sequence to retain the first M bits. The transmitted training sequence is comprised of BPSK symbols $b[n] = (-1)^{t[n]}$, where $t[n] \in \{0, 1\}$ are the elements of the m-sequence. The values of gain and timing mismatches are generated uniformly and independently in the range $[-0.1, 0.1]$. We generate channel coefficients with independent zero mean Gaussian real and imaginary parts.

We study the convergence of the algorithm using the square of the ℓ^2-error norm between the estimate and the truth as the metric. Thus, the channel estimation error is given by $\sum_{q=0}^{N-1} |h[q] - \hat{h}[q]|^2$. For mismatch estimation, we average the metrics over all sub-ADCs to obtain

$\frac{1}{L}\sum_{i=0}^{L-1}|g_i - \hat{g}_i|^2$ and $\frac{1}{L}\sum_{i=0}^{L-1}|\delta_i - \hat{\delta}_i|^2$ as the estimation error for the gain and timing mismatches, respectively.

A. Progress of iterations

We first consider the progress of the algorithm in the absence of noise. Fig. 2 depicts the decrease in estimation error as the iterations progress for $L = 32$ sub-ADCs, with channel length $N = 20$ and training length $M = 256$. We say that convergence is achieved when the estimation error falls below -100 dB. Fig. 2 shows that the algorithm converges in as few as 7 iterations. Since the graphs in Fig. 2 (with error expressed on a log scale) can be closely approximated by straight lines, we infer that the estimation errors decrease exponentially with iterations, with rate given by the slopes of the corresponding linear fits. We observe similar trends for the mismatch and channel estimation errors, and hence restrict our attention to the latter hereafter.

B. Rate of Convergence:

We now consider values of M given by powers of 2 ranging between 32 and 256. For each M, we choose the number of sub-ADCs L as powers of 2 between 2 and $M/2$. We define the *convergence rate* as the decrease in the channel estimate error (in dB) from the 9$^{\text{th}}$ to the 10$^{\text{th}}$ iteration. From Fig. 3, we observe that the convergence rate (averaged over 1000 random instances of channel and mismatches) is inversely proportional to L (with M fixed) and proportional to M (with L fixed). Thus, when convergence is desired with fewer iterations, the training sequence length must scale with the number of sub-ADCs.

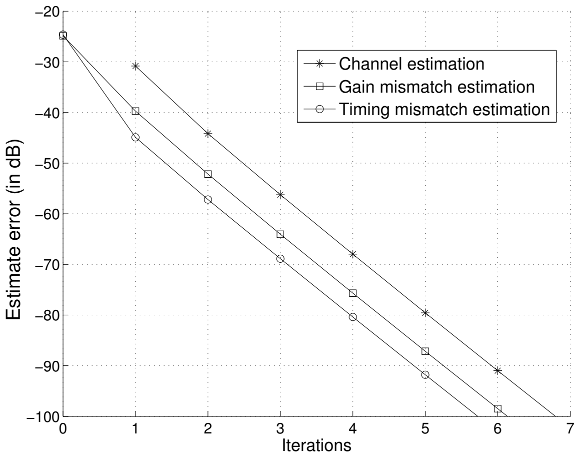

Fig. 2: Progress with iterations for $M = 256$, $N = 20$ and $L = 32$.

For fixed training length M, it is of interest to determine the largest number of sub-ADCs, L, for which the algorithm converges. When $M/L = 1$, the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in (13) are scalar multiples of each other. This is because the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ have only $M/L = 1$ row. This implies that there is no unique solution for the mismatch parameters $(g_i, \tilde{\delta}_i)$. Hence, we at least need $M/L \geq 2$. In Fig. 4, we demonstrate for $M = 32$ and $L = 16$ that the iterative algorithm gets stuck (albeit in fewer than

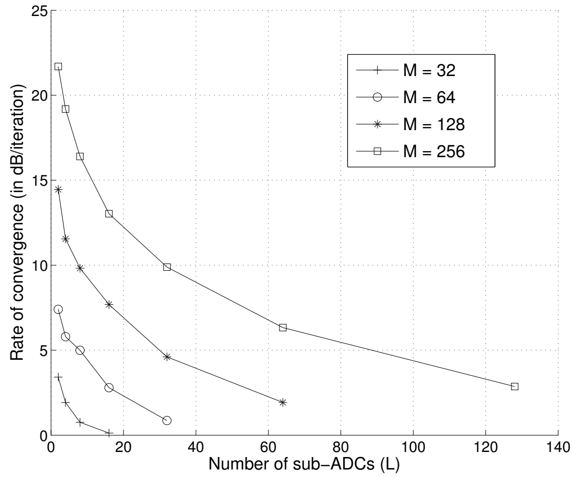

Fig. 3: Variation of mean convergence rate with L for different values of M ($N = 20$).

2 out of 10^4 instances) away from the true channel and mismatch parameters. We also observe from Fig. 4 that the graph of the least-squares cost (i.e., the right hand side of (10)) settles near -39 dB, or about 10^{-4}, suggesting that the solution is not a global minimum (otherwise the least squares cost would be 0). This cost floor is not observed for $M \geq 4L$ when we simulate 10^5 instances of channel and mismatch parameters, for values of M given by powers of 2 between 32 and 256.

C. Convergence with noise:

In Fig. 5, we consider thermal noise in (10) with a non-zero power $\sigma^2 = 1/\text{SNR}$. The mean-squared error (MSE) refers to the estimate error averaged over 50 algorithm runs with a randomly generated instance of thermal noise in each run. First, we observe that the iterative algorithm converges even in the presence of noise. We observe from Fig. 5 that the convergence rate, before the estimate error settles, ranges between $6 - 8$ dB/iteration, which is close to the no-noise, mean-convergence rate for $M = 256$ and $L = 64$ in Fig. 3. Thus, we observe that the convergence rate, before the algorithm settles, is fairly independent of the noise power.

D. Comparison with CRLB

We now evaluate the Cramer-Rao lower bound (CRLB) for the joint estimation problem, which serves as a lower bound on the level of MSE at which the algorithm settles in Fig. 5. Assuming i.i.d gaussian noise samples in (10), the likelihood function is a scaled (by $1/\sigma^2$) version of the least-squares cost function given in (10). We first evaluate the Fisher information matrix, $\boldsymbol{F}$, which can be inverted to obtain the CRLB:

$$\boldsymbol{F} = \frac{1}{\sigma^4}\mathbb{E}\left[\left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\left(\frac{\partial l(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^t\right] \tag{14}$$

where $\boldsymbol{x} = \{\Re[h_0], \cdots, \Re[h_{N-1}], \Im[h_0], \cdots, \Im[h_{N-1}], g_0, \cdots, g_{N-1}, \delta_0, \cdots, \delta_{N-1}\}$ denotes the vector of channel and mismatch parameters. We evaluate the derivatives required in (14) numerically at $\boldsymbol{x}$. Further, the expectation

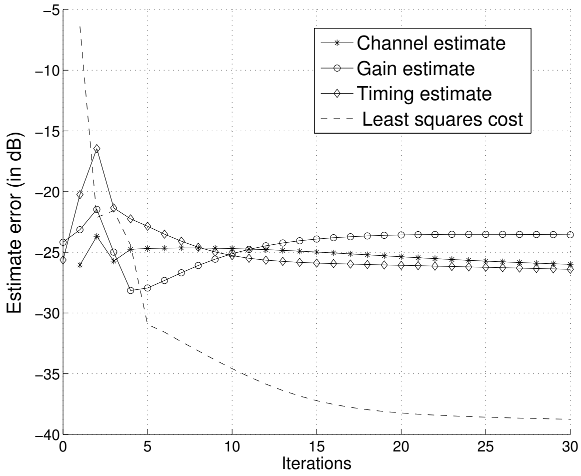

Fig. 4: Progress with iterations for the joint estimation algorithm with $(M, N, L) = (32, 16, 4)$.

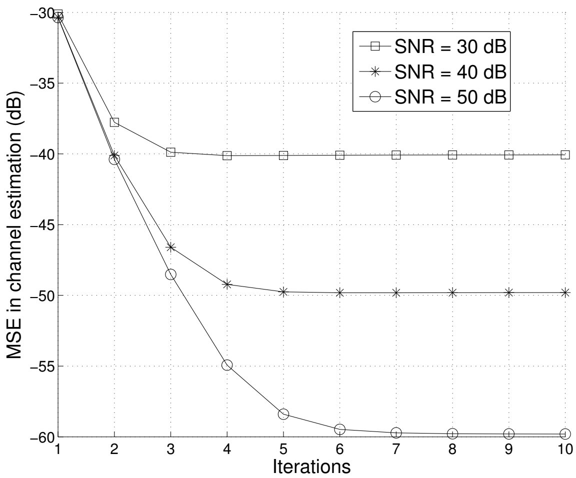

Fig. 5: Performance of the iterative algorithm in the presence of thermal noise with $(M, N, L) = (256, 64, 20)$.

in (14) is approximated by an empirical average over 10^4 instances of noise samples.

From Fig. 6, we observe that the channel MSE is close to the corresponding CRLB for all SNR $(= 1/\sigma^2)$ values considered. For example, the CRLB for channel estimation for SNR levels 30 dB and 40 dB are -40 dB and -50 dB, respectively, which are very close to the values the MSE settles in Fig. 5. However, the mismatch MSEs are approximately 3dB higher than the corresponding CRLB.

V. Training Sequence Design for Fast Convergence

In this section, we design training sequences so that the mismatch and channel parameters can be estimated with fewer iterations than with standard pseudorandom training sequences. The chosen training sequence satisfies the following three conditions:

(a) total length $M = NL$,

(b) periodic with period N,

(c) does not equal any i-circular shift of itself for $0 \leq i \leq N-1$.

In addition to these conditions, we also stipulate that N and L have no common factors. We refer to the subsequence $\{b_0, \cdots, b_{N-1}\}$ as the *training frame*. Thus, the actual training sequence (by conditions (a) and (b)) is a repetition of L training frames. In order to understand the higher convergence rate for the proposed sequence, we first simplify the structure of the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ in (9).

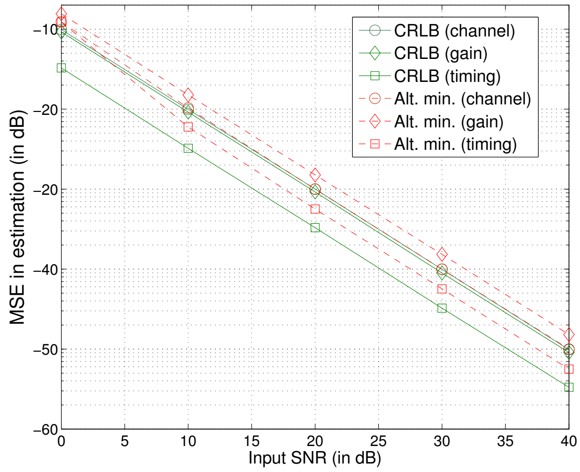

Fig. 6: Comparison of the algorithm's MSE with Cramer-Rao lower bounds for $(M, N, L) = (256, 64, 20)$.

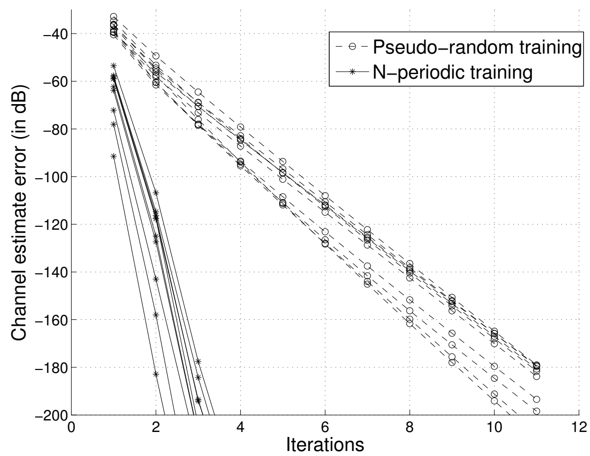

Fig. 7: Comparison of the progress of the iterative algorithm with the psuedo-random and the proposed training sequences. $(M, N, L) = (1216, 19, 64)$.

Lemma 1. *Structure of $\boldsymbol{A}_i$: The collection of the rows of $\boldsymbol{A}_i$ is the collection of all possible circular shifts of the training frame $\{b_0, \cdots, b_{N-1}\}$.*

Proof: We first prove by contradiction that no two rows of $\boldsymbol{A}_i$ are equal. From the definition in (9), we observe that p^{th} row of $\boldsymbol{A}_i$ is obtained by circularly shifting the first row by pL. Suppose the p_1^{th} and the p_2^{th} rows of $\boldsymbol{A}_i$ are equal. Using property (c) of the training sequence, this can only happen when $(p_1 - p_2)L \bmod N = 0$. Since N and L have no common factors, this implies $(p_1 - p_2) \bmod N =$

0. This leads to a contradiction, since $0 \leq p_1, p_2 \leq N-1$ or $-(N-1) \leq p_1, p_2 \leq (N-1)$. ■

The lemma is true for any i. Thus, the rows of $\boldsymbol{A}_i$ represent a permutation of the rows of A_0. Equivalently, $\boldsymbol{A}_i = \boldsymbol{P}_i \boldsymbol{A}_0$, where $\boldsymbol{P}_i$ represents the matrix obtained by performing the same row permutations on an Identity matrix.

Lemma 2. *Structure of $\boldsymbol{B}_i$: The matrix $\boldsymbol{B}_i$ can be expressed as $\boldsymbol{P}_i \boldsymbol{B}_0$, where $\boldsymbol{P}_i$ is the same row-permutation matrix as in $\boldsymbol{A}_i = \boldsymbol{P}_i \boldsymbol{A}_0$.*

Proof: From (9), we can rewrite the elements of $\boldsymbol{B}_i$ as,

$$\boldsymbol{B}_i(p,q) \quad = \sum_{q'=0, q' \neq q}^{N-1} b(i + pL - q') \text{ sinc}'(q' - q) \tag{15}$$

Using (15) and the definition of $\boldsymbol{A}_i$ from (9), it can be observed that the columns of $\boldsymbol{B}_i$ are linear combinations of the columns of $\boldsymbol{A}_i$, and hence we can write $\boldsymbol{B}_i = \boldsymbol{A}_i \boldsymbol{Q}$. It is noteworthy that $\boldsymbol{Q}$ does not depend on the choice of i. Since $\boldsymbol{A}_i = \boldsymbol{P}_i \boldsymbol{A}_0$, it implies that $\boldsymbol{B}_i = \boldsymbol{P}_i \boldsymbol{A}_0 \boldsymbol{Q}$. Thus, we infer that $\boldsymbol{B}_i = \boldsymbol{P}_i \boldsymbol{B}_0$. ■

Lemma 3. *Fast convergence: For the proposed training sequence, the channel estimate can be directly obtained from the received samples without any iterations (to a first-order in mismatch terms).*

Proof: We use Lemmas 1 and 2 in (5) to obtain

$$\boldsymbol{r}_i = \boldsymbol{P}_i[(1 + g_i)\boldsymbol{A}_0 + \tilde{\delta}_i \boldsymbol{B}_0]\boldsymbol{h} \tag{16}$$

Since row permutation matrices are invertible, we can premultiply (16) by $\boldsymbol{P}_i^{-1}$. Further, we sum over i to obtain

$$\sum_{i=0}^{L-1} \boldsymbol{P}_i^{-1} \boldsymbol{r}_i = \boldsymbol{A}_0 \boldsymbol{h} + g_i \delta_i \boldsymbol{B}_0 \boldsymbol{h} \tag{17}$$

where we use the normalization of $\{g_i\}$ and $\{\delta_i\}$ as given in (3). From (17), we note that the second term on the right-hand side includes a product of mismatch terms, which represents a second-order term in mismatch magnitude. Assuming mismatches to be small, we can approximate the estimate of $\boldsymbol{h}$ from (17) as

$$\boldsymbol{h} \approx \boldsymbol{A}_0^{-1} \sum_{i=0}^{L-1} \boldsymbol{P}_i^{-1} \boldsymbol{r}_i \tag{18}$$

We note that when we initialize the values of mismatches to zero in (17), the least-squares estimate for $\boldsymbol{h}$ is as given in (18). Thus, when we use the iterative algorithm of section III, we can estimate the channel and mismatch parameters, accurate to first order in the relative mismatch parameters, within the first iteration. ■

In Fig. 7, we plot the channel estimation error for pseudorandom training and the proposed periodic training. We choose $N = 19$ and $L = 64$ (no common factors) for a training length of $M = NL = 1216$. For the proposed training, we first randomly generate a vector with entries as ± 1 of length 19 and take its FFT, which is repeated 64 times to obtain the overall training sequence. This sequence has the advantage that any matrix $\boldsymbol{A}_i$ (see (9)) has all eigenvalues equal in magnitude. Thus, we see from (18) that when we invert $\boldsymbol{A}_i$, the resulting noise enhancement would be similar for the estimates for all of the channel coefficients. For pseudorandom training, we use the same m-sequence (from section IV) of length 255, repeated five times, but with the last 59 bits deleted to obtain a total length of 1216. We observe that the proposed sequence indeed demonstrates much better convergence rates, over several random instances of channel and mismatch parameters, compared to pseudorandom training.

VI. Conclusion

We have demonstrated in this paper that effective mismatch and channel estimation is possible with the pseudorandom training typical of communication systems. However, the required training length does scale with the number of sub-ADCs (four times the number of sub-ADCs is a good rule of thumb), so that offline calibration with longer training sequences may be more suitable for a large number of ADCs. Optimized periodic training sequences can be employed to considerably speed up convergence relative to pseudorandom training; however, these are typically quite long for channels of moderate length, and may be more suitable for offline calibration.

To get a sense of the numbers, consider the example of an indoor multiGigabit wireless link operating over the 60 GHz unlicensed band. Assuming 1 GHz of transmission bandwidth, a typical delay spread of 20 ns corresponds to $N = 20$ channel coefficients. When we use a time-interleaved ADC with $L = 32$ sub-ADCs, each ADC samples at a rate of 31.25 MHz, which is small enough to be implemented using power efficient pipelined or SAR architectures [2]. Using a pseudorandom based training sequence, with length $M = 256$, three iterations suffice to obtain the estimates accurate to an error less than -50 dB. The complexity of estimation is tractable, owing to closed-form solutions at each step of the iteration. Alternatively, when we use the proposed training sequence with length $M = 672$, one iteration suffices to obtain the same estimation error (we estimate $N = 21$ channel coefficients in order to ensure that N and L have no common factors).

References

[1] C. H. Doan, S. Emami, D. A. Sobel, A. M. Niknejad and R. W. Brodersen, "Design Considerations for 60 GHz CMOS Radios," *IEEE Communications Magazine* , Dec. 2004.

[2] B. Le, T.W. Rondeau, J. H. Reed and C. W. Bostian, "Analog-to-digital converters," *IEEE Signal Processing Magazine* , vol.22, Nov. 2005.

[3] "ADC083000: 8-Bit, 3 GSPS, High Performance, Low Power A/D Converter," *National semiconductors*, www.national.com

[4] B. P. Ginsburg and A. P. Chandrakasan, "Highly Interleaved 5-bit, 250-MSample/s, 1.2-mW ADC With Redundant Channels in 65-nm CMOS," *IEEE J. Solid State Circuits*, vol. 43, Dec. 2008.

[5] P. Sandeep, M. Seo, U. Madhow. M. Rodwell, "Joint mismatch and channel compensation for high-speed OFDM receivers with time-interleaved ADCs," *To appear in IEEE Tran. Commun.*, pre-print available at www.ece.ucsb.edu/wcsl

[6] V. Divi and G. W. Wornell, "Blind Calibration of Timing Skew in Time-Interleaved Analog-to-Digital Converters," *IEEE Jour. selec. topics in Sig. Proc.* , vol. 3, Jun. 2009.

[7] T. Strohmer and J. Xu, "Fast Algorithms for Blind Calibration in Time-Interleaved Analog-to-Digital Converters," *IEEE ICASSP* , Apr. 2007.

[8] A. Haftbaradaran and K. W. Martin, "A Background Sample-Time Error Calibration Technique Using Random Data for Wide-Band High-Resolution Time-Interleaved ADCs," *IEEE Trans. Circuits and Systems-II*, vol. 55, Mar. 2008.

[9] S. Huang and B. C. Levy, "Blind Calibration of Timing Offsets for Four-Channel Time-Interleaved ADCs," *IEEE Tran. Circuits and Systems*, vol. 54, pp 863-876, April 2007.

[10] D. Camarero, K. B. Kalaia, J. F. Naviner, and P. Loumeau, "Mixed-Signal Clock-Skew Calibration Technique for Time-Interleaved ADCs," *IEEE Trans. Circuits and Systems - I*, vol. 55, Dec. 2008.

[11] M. Seo, M. J. W. Rodwell and U. Madhow, "Comprehensive digital correction of mismatch errors for a 400-msamples/s 80-dB SFDR time-interleaved analog-to-digital converter," *IEEE Trans. Microwave Theory and Techniques*, vol. 53, pp. 1072-1082, March 2005.

[12] Y. Oh and B. Murmann, "System Embedded ADC Calibration for OFDM Receivers," *IEEE Trans. Circuits and Systems-I*, vol.53, Aug. 2006.

[13] Binmore and Davies, "Calculus Concepts and Methods," *Cambridge University Press*, p.190, 2007.