On the convergence of joint channel and mismatch estimation for time-interleaved data converters

Sandeep Ponnuru and Upamanyu Madhow [1]

***Abstract*—Mostly digital architectures for communication transceivers rely on the use of accurate analog-to-digital converters (ADCs), which becomes a bottleneck in scaling to multi-Gigabit speeds. A promising workaround is to employ slower, power-efficient sub-ADCs in a time-interleaved ADC (TI-ADC) architecture. While mismatch between sub-ADCs can lead to performance floors, recent work shows that this can be mitigated using joint channel and mismatch compensation algorithms. In this paper, we investigate the scalability and convergence of a recently proposed iterative channel and mismatch estimation algorithm, and derive rules of thumb relating the required length of training to the number of sub-ADCs.**

I. Introduction

In the past decade, the adoption of a mostly digital architecture significantly contributed to the wide-spread deployment of wireless LAN and cellular systems. However, when we try to extend the same advantage to future communication systems operating at multi-Gigabit speeds, we are limited by the difficulty of realizing fast analog-to-digital converters (ADCs) that provide sufficient precision (8-10 bits) with a low power consumption ($<$ 100 mW) [1], [2], [3]. One approach to this bottleneck is to employ a time-interleaved architecture (see Fig. 1), where several low-rate "sub-ADCs" are interleaved in parallel to implement a fast ADC [4]. By virtue of their relatively lower speeds, the sub-ADCs can be realized using power-efficient architectures (e.g., pipelined, successive approximation), instead of the power-hungry flash architectures typically required for high-speed ADCs. However, an inherent limitation of the time-interleaved ADC (TI-ADC) architecture is the mismatch between the sub-ADCs [5], [6], which, if left uncompensated, can lead to error floors. Several schemes have been proposed for dedicated mismatch calibration in TI-ADCs [4], [5], [9]-[13]. Furthermore, for communication links, channel and mismatch estimation can be done jointly, leveraging the training already available on the link [7], [8]. In this paper, we investigate the scalability of joint channel and mismatch compensation as the number of sub-ADCs increases, with a view to characterizing the convergence rate of a previously proposed iterative algorithm as a function of the choice of training sequence (most importantly, the training sequence length) and the number of sub-ADCs.

Contributions: We consider an iterative algorithm for joint channel and mismatch estimation proposed in [8], which had been empirically observed to provide exponential convergence. In this paper, we provide an analytical approximation to the convergence rate based on the observation that the channel estimates in successive iterations are well approximated as proceeding along a straight line (in N-dimensional space, where N is the number of channel taps), see Figure 2, which shows that the straight line approximation improves as the training sequence length M increases. The expressions that we obtain agree closely with the empirically observed convergence rates. While the convergence rate does depend on the specific choice of training sequence, it is mostly affected by the length M of the training sequence relative to the number L of sub-ADCs. In particular, they imply that the training sequence length must scale linearly with the number of sub-ADCs in order to keep the number of iterations (which dominates the complexity of our iterative algorithm) the same. For random training sequences, the convergence rate can be approximated by $20\log_{10}(M/L)$ dB/iteration, which is reasonably close to the observed rates for $M/L \leq 8$.

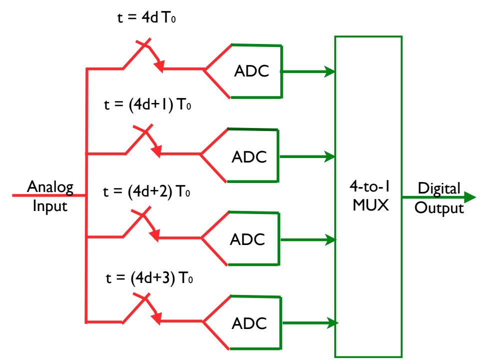

Fig. 1: **A time-interleaved ADC**: four sub-ADCs interleaved to achieve an overall sampling rate of $1/T_0$ (d takes integer values). Red and green indicate the analog and digital components respectively.

Organization: We organize the rest of the paper as follows. First, in section II, we set the stage by giving models of the communication system and the time-interleaved ADC with mismatches, and also briefly describe the iterative algorithm in [8] for jointly estimating the channel and mismatches. Then, in section III, we give formulae obtained for the convergence rate of this iterative algorithm when we used the argument that the channel estimates progress along a straight line. We present simulation results in Section IV to verify the analytical formulae. Later, in section V, we illustrate all the steps of the derivation, and finally conclude the paper in section VI.

[1]The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara CA, 93117 USA. E-mail: ponnuru.sandeep@gmail.com, madhow@ece.ucsb.edu

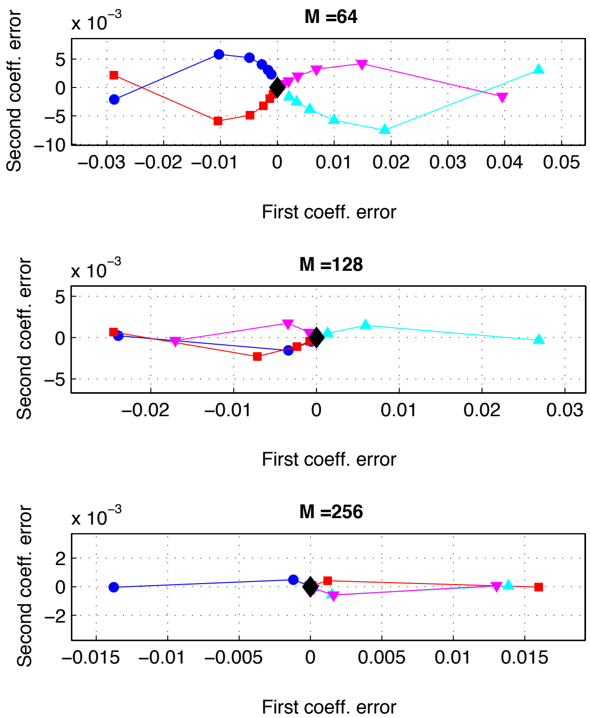

Fig. 2: **Approximately linear trajectories of the channel estimate with iterations of joint estimation algorithm**: Trajectories for several randomly generated channels (length $N = 20$ taps) are plotted in the two-dimensional plane containing the true channel (indicated by black diamond), the initial channel estimate, and the channel estimate after the first iteration. M denotes the training sequence length and $L = 8$ ADCs are interleaved. Other simulation parameters are given in Section IV.

II. Joint Channel and Mismatch Estimation

The received signal is given by

$$r(t) = \sum_{k=0}^{M-1} b[k]h(t-kT) + n(t), \quad (1)$$

where $\{b[k]\}$ is the training sequence of length M, $1/T$ is the symbol rate, and the "channel impulse response" $h(t)$ includes the effect of transmit, channel and receive filters. We assume Nyquist sampling using the TI-ADC, so that the sampling interval for each sub-ADC is LT, where L is the number of sub-ADCs. For simplicity, we consider only the in-phase baseband signal (the analysis extends in a straightforward manner to include the quadrature part).

Gain and timing mismatches are the dominant sources of error in TI-ADC architectures. Gain mismatch is a linear effect, while timing mismatch is nonlinear. However, as shown in our prior work [8], timing mismatch can be well approximated as linear in deriving the iterative algorithm for joint channel and mismatch estimation investigated here (and described shortly). However, we restrict attention in our analysis to gain mismatch in this paper, in order to obtain detailed insight into the progress of the algorithm. In this setting, the received samples are given by [7]:

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

where $i = m \bmod L = i$ (corresponding to the samples collected by sub-ADC i, $i = 0, 1, ..., L-1$), g_i denotes the gain mismatch for the ith sub-ADC (with respect to unity gain), and $\{h[m]\}$ denotes the channel coefficients expressed at symbol rate, assumed to be non-zero only when $m \in \{0, \cdots, N-1\}$.

We now express (2) in vector notation:

$$\boldsymbol{r}_i = (1+g_i)\boldsymbol{A}_i\boldsymbol{h} + \boldsymbol{n}, \quad (3)$$

where $\boldsymbol{r}_i$, $\boldsymbol{h}$ and $\boldsymbol{n}_i$ represent the vectors of received samples, channel and noise. The matrix $\boldsymbol{A}_i$ depending on the training sequence can be readily defined using (2). Given the vectors of received samples in (3), the goal is to estimate the channel vector $\boldsymbol{h}$, and the L mismatch parameters, $\{g_i\}$. Neglecting correlations (if any) among the noise samples $\boldsymbol{n}$, the maximum likelihood (ML) estimates satisfy:

$$(\hat{\boldsymbol{h}}, \{\hat{g}_0, \cdots, \hat{g}_{L-1}\}) = \arg\min \sum_{i=0}^{L-1} ||\boldsymbol{r}_i - (1+g_i)\boldsymbol{A}_i\boldsymbol{h}||^2 \quad (4)$$

We now review the alternating minimization strategy proposed in [8] to solve the joint estimation problem in (4). Given the mismatch estimates $g_i^{(n)}$ at the n^{th} iteration, the ML channel estimate for (4) is the solution to

$$\left(\sum_{i=0}^{L-1}(1+g_i^{(n)})\boldsymbol{A}_i^t\boldsymbol{A}_i\right)\boldsymbol{h}^{(n)} = \sum_{i=0}^{L-1}\boldsymbol{A}_i^t\boldsymbol{r}_i, \quad (5)$$

where $n = 0, 1, 2, \cdots$. On the other hand, given the channel estimates, the ML estimate for the mismatch parameters $g_i^{(n)}$ progress as follows:

$$1 + g_i^{(n)} = \frac{(\boldsymbol{r}_i, \boldsymbol{A}_i\boldsymbol{h}^{(n-1)})}{||\boldsymbol{A}_i\boldsymbol{h}^{(n-1)}||^2}, \quad (6)$$

The algorithm is initialized by setting the initial mismatch estimates to zero: $g_i^{(0)} = 0$. We note that there is a scale ambiguity in (4): if $(\boldsymbol{h}, \{g_i\})$ are solutions, then so are $(x\boldsymbol{h}, \{g_i/x\})$. We therefore scale the gain estimates $1 + g_i^{(n)}$ by a constant to set the mean mismatch to zero.

III. Convergence Rate

In (5), the complexity of each iteration depends dominantly on inverting an $N \times N$ matrix, hence the overall complexity of the algorithm scales with the number of iterations. Thus, a scalable design for a TI-ADC architecture must maintain the convergence rate as the number of sub-ADCs, L, increases. We therefore focus on characterizing the convergence rate, defined as

$$\alpha_n = \frac{||\boldsymbol{h} - \boldsymbol{h}_n||}{||\boldsymbol{h} - \boldsymbol{h}_{n+1}||}, \quad (7)$$

(expressed on a logarithmic scale as $20\log_{10}(\alpha_n)$ dB/iteration).

After ignoring noise, our analysis relies on the interesting behavior depicted in Fig. 2, where the channel estimates $\boldsymbol{h}^{(n)}$ progress along a straight line joining the initial channel estimate $\boldsymbol{h}^{(0)}$ and the true channel vector $\boldsymbol{h}$. Our results (derived in Section V) indicate that the convergence rate, α, depends on the norms and the inner products of the vectors $\boldsymbol{u}_i = \boldsymbol{A}_i\boldsymbol{h}$ and $\boldsymbol{v}_i = \boldsymbol{A}_i\boldsymbol{h}^0$. Further, given the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, the asymptotic convergence rate (as iteration number n gets large) is independent of n and weakly depends on the mismatches. In the limit of small mismatches (tending to zero), the asymptotic convergence rate is given by,

$$\alpha = \frac{\sum_{i=0}^{L-1}\sigma_i}{\sum_{i=0}^{L-1}\rho_i(\kappa_i - \overline{\kappa})} \tag{8}$$

where the terms ρ_i, σ_i and κ_i are defined as follows:

$$\rho_i = (\boldsymbol{A}_i\boldsymbol{h}, \boldsymbol{A}_i(\boldsymbol{h}-\boldsymbol{h}^0)), \quad \sigma_i = ||\boldsymbol{A}_i(\boldsymbol{h}-\boldsymbol{h}^0)||^2, \quad \kappa_i = \frac{\rho_i}{||\boldsymbol{A}_i\boldsymbol{h}||^2}, \tag{9}$$

and $\overline{\kappa}$ is a weighted average of κ_i:

$$\overline{\kappa} = \frac{1}{L}\sum_{i=0}^{L-1}(1+g_i)\kappa_i \tag{10}$$

The formula in (8) enables us to compute convergence rates as a function of the training sequence and the initial channel estimate $\boldsymbol{h}^{(0)}$. However, it can be simplified to provide the following compact rule of thumb by exploiting the random-like properties of long pseudo-random sequences:

$$\alpha_r = \frac{M}{L} \tag{11}$$

The preceding formula indicates that we need to increase the training sequence length M linearly with the sub-ADC number L in order to maintain a given convergence rate.

Next, we present simulation results comparing our analytical results (8) and (11) with empirically observed convergence rates, followed by a sketch of the derivations of these analytical results.

IV. Simulation Results

We consider a maximal length pseudo-random sequence $p[n]$ with generator polynomial $z^8 + z^6 + z^5 + z^4 + 1$ as the training sequence. This sequence repeats every 255 symbols. For training length $M < 255$, we truncate the sequence to M symbols. For transmission, we use antipodal signaling so that the transmitted signal is $(-1)^{p[n]}$ shaped by the transmit filter.

Fig. 3 plots the convergence rate as a function of the number of sub-ADCs, L. As the length of training sequence, M, increases, the convergence rates improve for a fixed L. The analytical estimates obtained using the straight-line progression approximation agree closely with the observed convergence rates (averaged over 50 randomly generated channel and mismatch sets). The analytical results deviate from simulations at large L (i.e., small M/L), which is consistent with Fig. 2, which shows that approximating the algorithm's progression as a straight line is less accurate for smaller M/L. Next, fixing the training sequence length $M = 256$, we see from Fig. 4 that the convergence rate agrees closely with the formula in (11) for relatively large L, when $M/L \leq 8$.

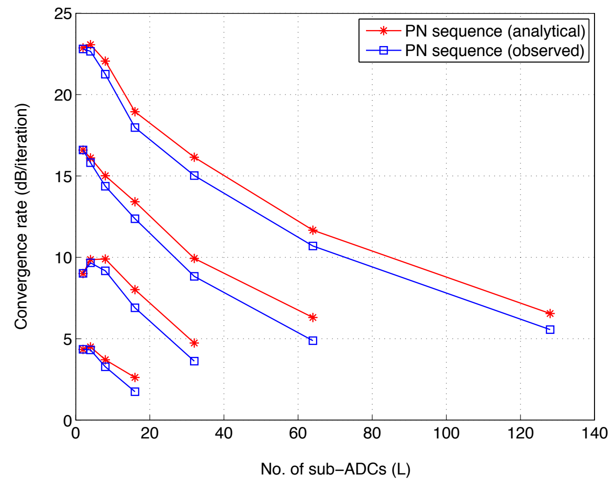

Fig. 3: **Mean convergence rate of alternating minimization**: Training sequence length M increases as we move away from origin as 32, 64, 128 and 256.

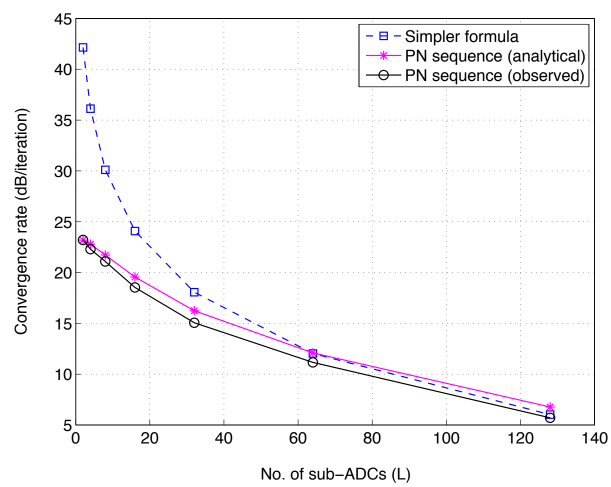

Fig. 4: **Simple formula for convergence rate** : Mean convergence rate is well approximated as $20\log_{10}(M/L)$ dB/iteration when $M/L \leq 8$ for a pseudo-random training sequence of length $M = 256$.

V. Sketch of Derivations

In this section, we sketch the derivations of the convergence rate formulae (8) and (11) in Section III. We introduce the more compact notation $\tilde{g}_i = 1 + g_i$, and refer to $\tilde{g}_i$ as the gain of the i^{th} sub-ADC.

A. Straight line progression

The straight line progression motivated by Fig. 2 is mathematically expressed as:

$$\boldsymbol{h}^{(n)} = \beta_n \boldsymbol{h}_0 + (1-\beta_n)\boldsymbol{h} \tag{12}$$

where n denotes the iteration number.

The gain estimates can be written using (6) as follows (recall that we ignore noise in our analysis):

$$\hat{\tilde{g}}_i = \tilde{g}_i f_i(\beta_{n-1}) \tag{13}$$

where $f_i(\beta_{n-1})$ is given by

$$f_i(\beta_{n-1}) = \frac{(\boldsymbol{A}_i\boldsymbol{h}, \boldsymbol{A}_i\boldsymbol{h}_{n-1})}{(\boldsymbol{A}_i\boldsymbol{h}_{n-1}, \boldsymbol{A}_i\boldsymbol{h}_{n-1})} \tag{14}$$

Under the straight line approximation, the channel estimate at the n^{th} iteration, given the gain estimates in (13), depends only on a single scalar parameter β_n. We can now use (12) to recast the channel estimation problem in terms of β_n:

$$\beta_n = \arg\min \sum_{i=0}^{L-1} ||\boldsymbol{r}_i - \hat{\tilde{g}}_i \boldsymbol{A}_i(\beta_n\boldsymbol{h}_0 + (1-\beta_n)\boldsymbol{h})||^2 \tag{15}$$

We now evaluate (15) using $\boldsymbol{r}_i$ from (3) (neglecting noise) and $\hat{\tilde{g}}_i$ from (13) to obtain

$$\beta_n = \arg\min \sum_{i=0}^{L-1} ||\boldsymbol{u}_i - \beta_n \boldsymbol{v}_i||^2 \tag{16}$$

where the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are given by

$$\boldsymbol{u}_i = \tilde{g}_i(1 - f_i(\beta_{n-1}))\boldsymbol{A}_i\boldsymbol{h} \tag{17}$$
$$\boldsymbol{v}_i = \tilde{g}_i f_i(\beta_{n-1})\boldsymbol{A}_i(\boldsymbol{h}_0 - \boldsymbol{h}) \tag{18}$$

The solution to (15) can be obtained in terms of the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ as follows:

$$\beta_n = \frac{\sum_{i=0}^{L-1} \Re(\boldsymbol{v}_i^H \boldsymbol{u}_i)}{\sum_{i=0}^{L-1} ||\boldsymbol{v}_i||^2} \tag{19}$$

Using the definition of convergence rate in (7) and of β_n in (12), we obtain the convergence rate α as

$$\alpha = \frac{\beta_{n-1}}{\beta_n} \tag{20}$$

We now use the expression for β_n from (19) in (20) to obtain an expanded expression for α :

$$\alpha = \beta_{n-1}\frac{\sum_{i=0}^{L-1} ||\boldsymbol{v}_i||^2}{\sum_{i=0}^{L-1} \Re(\boldsymbol{v}_i^H \boldsymbol{u}_i)} \tag{21}$$

Using the expressions for vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ from (17), we can rewrite α as

$$\alpha = \beta_{n-1}\frac{\sum_{i=0}^{L-1} \sigma_i \tilde{g}_i^2 f_i^2(\beta_{n-1})}{\sum_{i=0}^{L-1} \rho_i \tilde{g}_i^2 f_i(\beta_{n-1})[1 - f_i(\beta_{n-1})]} \tag{22}$$

where the scalars σ_i and ρ_i are defined as in (9).

B. Asymptotics of the convergence rate

While the iterative algorithm only takes a few steps to converge in a well-designed system, it is useful to compute the asymptotic convergence rate (i.e., the limit of α for large n) in order to provide simplified rules of thumb. From the expression for convergence rate in (22), we see that the iteration number is buried in β_{n-1}. If the algorithm converges, $\beta_{n-1} \to 0$, hence we evaluate the limit of α as $\beta_{n-1} \to 0$. We first consider the following limits:

$$\lim_{\beta_{n-1}\to 0} f_i(\beta_{n-1}), \qquad \lim_{\beta_{n-1}\to 0} \frac{1 - f_i(\beta_{n-1})}{\beta_{n-1}} \tag{23}$$

where $f_i(\beta_{n-1})$ is defined in (14) in terms of $\boldsymbol{h}_{n-1}$. We can rewrite $f_i(\beta_{n-1})$ as a function of β_{n-1} by using the expression for $\boldsymbol{h}_{n-1}$ from (12):

$$f_i(\beta_{n-1}) = \frac{\beta c_i + (1-\beta)e}{\beta^2 e_0 + (1-\beta)^2 e + 2\beta(1-\beta)c_i} \tag{24}$$

where we have dropped the subscript of β for simplicity, and where the scalars e_i, e_i^0, c_i are defined as:

$$e_i = ||\boldsymbol{A}_i\boldsymbol{h}||^2, \qquad e_i^0 = ||\boldsymbol{A}_i\boldsymbol{h}_0||^2, \qquad c_i = (\boldsymbol{A}_i\boldsymbol{h}, \boldsymbol{A}_i\boldsymbol{h}_0) \tag{25}$$

We can now employ expansions in terms of β to evaluate the desired limits as follows (details are omitted due to lack of space):

$$\lim_{\beta\to 0} f_i(\beta) = 1$$
$$\lim_{\beta\to 0} \frac{1 - f_i(\beta)}{\beta} = \frac{c_i - e_i}{e_i} \triangleq \kappa_i \tag{26}$$

We now obtain the asymptotic convergence rate by evaluating the limit of (22) as β_{n-1} tends to zero:

$$\alpha = \frac{\sum_{i=0}^{L-1} \sigma_i \tilde{g}_i^2}{\sum_{i=0}^{L-1} \rho_i \tilde{g}_i^2 \kappa_i} \tag{27}$$

C. Accounting for gain scaling

We now modify the asymptotic convergence rate estimate to account for the scaling step in our algorithm, where the gain mismatches are scaled such that the mean mismatch is zero. We use (13) to obtain the modified gain mismatch estimates as,

$$\text{modified } \hat{\tilde{g}}_i = \frac{\tilde{g}_i f_i(\beta)}{\frac{1}{L}\sum_{i=0}^{L-1} \tilde{g}_i f_i(\beta)} \tag{28}$$

Comparing (28) with (13), we observe that they are identical except that in the former, $f_i(\beta)$ is scaled by an additional factor, G, given by,

$$G = \frac{1}{L}\sum_{i=0}^{L-1} \tilde{g}_i f_i(\beta) \tag{29}$$

We can simplify (29) by using $f_i(\beta)$ from (24) and the fact that the true mismatches have zero-mean:

$$G = 1 - \beta\overline{\kappa} + o(\beta), \text{ where } \overline{\kappa} = \frac{1}{L}\sum_{i=0}^{L-1}\kappa_i\hat{\tilde{g}}_i \quad (30)$$

We now re-evaluate the limits in (26) taking into account that $f_i(\beta)$ are scaled by G:

$$\lim_{\beta\to 0} f_i(\beta) = 1, \quad \lim_{\beta\to 0}\frac{1 - f_i(\beta)}{\beta} = \kappa_i - \overline{\kappa} \quad (31)$$

Hence, the asymptotic convergence rate gets modified to:

$$\alpha = \frac{\sum_{i=0}^{L-1}\sigma_i\tilde{g}_i^2}{\sum_{i=0}^{L-1}\rho_i\tilde{g}_i^2(\kappa_i - \overline{\kappa})} \quad (32)$$

D. First order approximation w.r.t mismatches

In practice, the mismatches are small, and the gains $\{\tilde{g}_i\}$ are in a small range around unity. From (32), we understand that these cause second order variations in the convergence rate. Ignoring these, we can simplify the expression for the convergence rate as follows:

$$\alpha = \frac{\sum_{i=0}^{L-1}\sigma_i}{\sum_{i=0}^{L-1}\rho_i(\kappa_i - \overline{\kappa})} \quad (33)$$

E. Long pseudo-random training sequence and channel

We now use the random-like properties of long pseudo-random sequences to simplify (8) to (11). For large N and $M/L \ll N$, the matrix $\boldsymbol{A}_i^t\boldsymbol{A}_i$ has M/L non-zero eigenvalues all of which lie close to N. Then, we can rewrite ρ_i in (9) as,

$$\rho_i = N\sum_{i=1}^{M/L}(\boldsymbol{h}^t\boldsymbol{a}_i)(\boldsymbol{a}_i^t(\boldsymbol{h} - \boldsymbol{h}_0)) \quad (34)$$

where $\{\boldsymbol{a}_i\}$ represent orthonormal eigenvectors of $\boldsymbol{A}_i^t\boldsymbol{A}_i$. The random variables $\boldsymbol{h}^t\boldsymbol{a}_i$ (and similarly, $\boldsymbol{a}_i^t(\boldsymbol{h} - \boldsymbol{h}_0)$) are zero-mean and independent across i. Also, since a randomly chosen channel vector, say $\boldsymbol{h}$, looks random w.r.t an orthonormal basis, say $\{\boldsymbol{a}_i\}$, we have $\mathbb{E}[(\boldsymbol{h}^t\boldsymbol{a}_i)^2] = ||\boldsymbol{h}||^2/N$. This implies,

$$\mathbb{E}[\rho_i^2] = \frac{M}{L}\cdot||\boldsymbol{h}||^2\cdot||\boldsymbol{h} - \boldsymbol{h}_0||^2 \quad (35)$$

We now estimate the terms κ_i and σ_i in the expression (33) for the convergence rate, using the approximation $||\boldsymbol{A}_i\boldsymbol{x}||^2 \approx \frac{M}{L}||\boldsymbol{x}||^2$. This yields

$$\kappa_i \approx \frac{\rho_i}{\frac{M}{L}||\boldsymbol{h}||^2}, \quad \sigma_i \approx \frac{M}{L}||\boldsymbol{h} - \boldsymbol{h}_0||^2 \quad (36)$$

For small mismatches, we can approximate $\overline{\kappa}$ by the mean of κ_i, which implies the convergence rate to be,

$$\alpha \approx \frac{M||\boldsymbol{h} - \boldsymbol{h}_0||^2}{\sum_{k=0}^{L-1}\rho_k^2 - \frac{\left(\sum_{k=0}^{L-1}\rho_i\right)^2}{L}}\cdot\frac{M}{L}||\boldsymbol{h}||^2 \quad (37)$$

Using (35) and neglecting the negative term in the denominator, we obtain the result in (11).

VI. Conclusions

We have studied an iterative algorithm (with exponential convergence) for jointly estimating the channel and time-interleaved ADC mismatches in a communication receiver. Based on the interesting observation that the progression of channel estimates is well approximated as a straight line joining the initial estimate and the true channel, we were able to derive closed form expressions for the convergence rate. A key insight is that the training sequence length must scale linearly with the number of sub-ADCs in order to maintain the convergence rate. As a concrete example, consider a sampling rate of 2 Gsamples/s obtained using a TI-ADC with 32 sub-ADCs, each sampling at 62.5 Msamples/s (easily realized using a power-efficient architecture). For a nominal channel delay spread of 10 ns (e.g., for an indoor 60 GHz channel), we have $N = 20$ channel coefficients. From the results in this paper, a pseudo-random training sequence of length $M = 128$ results in a convergence rate of approximately $20\log(M/L) = 13.3$ dB/iteration, implying that 3 iterations of the algorithm could lower the mean-squared error in the channel estimates by 40 dB, which should be adequate to support constellations as large as 16QAM without significant error floors.

References

[1] J. Singh, S. Ponnuru and U. Madhow, "Multi-Gigabit communication: the ADC bottleneck," *Presented at IEEE International Conference on Ultra-Wideband*, Vancouver, Canada, Feb 2009.

[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] 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.

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

[7] P. Sandeep, M. Seo, U. Madhow. M. Rodwell, "Joint mismatch and channel compensation for high-speed OFDM receivers with time-interleaved ADCs," *IEEE Tran. Commun.*, vol. 58, Aug. 2010.

[8] S. Ponnuru and U. Madhow, "On the scalability of joint channel and mismatch estimation for time-interleaved analog-to-digital conversion in communication receivers," *Presented at Allerton Conference on Communication, Control and Computing*, Moticello, Illinois, Sep. 2010.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.