Efficient Calibration of Time-Interleaved ADCs via Separable Nonlinear Least Squares

Jiadong Xu and Thomas Strohmer
Department of Mathematics
University of California, Davis
email: jiadong, strohmer@math.ucdavis.edu

Abstract— **We present a digital background technique for correcting the time and gain mismatches in a time-interleaved analog-to-digital converter (ADC) system. The proposed approach is applicable to any number of time-interleaved ADCs and requires only modest oversampling. While the algorithm is mainly designed for blind calibration, it can as well be operated in non-blind mode. Theoretical analysis and numerical simulations show fast convergence and good estimation performance of the proposed algorithm. For instance, for an 8-ADC system, numerical experiments demonstrate that the resulting signal to noise ratio (SNR) of the output signal after mismatch detection and interpolation, is higher than the SNR of the input signal.**

Index Terms— **Analog-digital converter, Time-interleaved, Calibration, Nonlinear estimation, Signal sampling, Least square method.**

I. Introduction

Due to ever-increasing demand for higher data rate and larger bandwidth, modern digital communication systems depend on analog-to-digital converters (ADCs) operating at faster speed and providing higher resolution. However it is difficult to scale ADCs to satisfy these performance requirements while maintaining low production costs. Time-interleaving multiple ADCs is a well known approach to increase the sampling rate [1], [2] while keeping hardware costs at bay.

Time-interleaved ADCs increase the sampling rate of a system by sending the analog input signal simultaneously to multiple ADCs, which have the same sampling rate but different phases [1], as depicted in Fig. 1. In this way, a system with sampling rate $\frac{1}{T}$ can be realized from r individual ADCs, each operating with a sampling rate $\frac{1}{rT}$. This idea has been proposed for various applications such as ultra wideband communications [3], [4]. However, interleaving of multiple ADCs is sensitive to the time errors and gain mismatches between different interleaved ADCs, which degrades the performance of ADCs significantly if the errors remain uncorrected [5]. A considerable amount of research has been done on calibration to correct the timing errors and gain mismatches. Hardware methods for compensation have been proposed in [6], [7], however the analog components involved make such approaches often difficult to use in practice. Some methods employ training signals methods [8], [9], which is also known as foreground- or nonblind calibration. This approach however may cause problems since it requires to inject the training signal periodically during operation of the ADC. In [10], [11], [12], [13] blind estimation methods are proposed with some appealing features, such as the calibration can be done in the normal operation and the mismatch changes are easy to track.

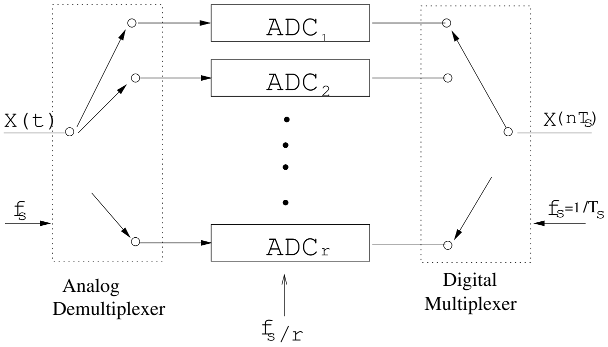

Fig. 1. Block diagram of the time-interleaved ADCs.

Once the mismatch errors are estimated, we have two ways to eliminate the error effects. One way is to adjust the sampling clock for each ADC [14], which may increase the random jitter of each controlled clock. Another way is to do the interpolation to achieve the correct values at the ideal times [15]. The second approach is attractive because it can be done with the required accuracy using digital signal-processing circuits, which are portable and will benefit from evolving scaled CMOS technologies [9]. There are a number of algorithms in the literature to perform this step. A method that is both numerically efficient and requires only short filters to achieve high accuracy has been proposed in [16].

As pointed out in [17], existing estimation methods are *"either imprecise, limited in the number of channels, or have an enormous computational complexity"*. Our goal in this paper is to develop a method that overcomes these limitations. We formulate the timing and gain estimation problem as separable nonlinear least squares problem and propose a Gauss-Newton type iteration method for its solution. The proposed method can be applied in both blind calibration and non-blind calibration mode and requires only a modest amount of oversampling. Unlike some other methods, it is computationally efficient and

This work was partially supported by NSFDMS grant 0511461.

works for any number of interlaced ADCs. Theoretical result and numerical simulations show that the method exhibits fast convergence.

Our method is tested in blind calibration mode for an 8-ADC system with 40% oversampling. An interpolation method is applied at the output and the resulting SNR of output signal is shown to be higher than the SNR of the input signal, which we call error-free ADC system. When applied in non-blind mode, we find that even for 64 or 256 time-interleaved ADCs the algorithm provides an error-free ADC.

The rest of this paper is organized as follows. In section II, the details of the calibration problem for an r-ADC system are described. Our Gauss-Newton method is proposed in section III, followed by the analysis of its convergence in section IV. In section V, some technical details for this method is discussed. In section VI, we discuss the application of our method for non-blind calibration. The interpolation method at the output is discussed in section VII. The simulation results of our calibration methods are shown in section VIII. Section IX concludes this paper.

II. Problem description

We model the converter input signal $x(t)$ as a stationary bandlimited zero-mean Gaussian random process with bandwidth B, whose continuous-time Fourier Transform

$$X(w) = \int_{t=-\infty}^{\infty} x(t) e^{-2\pi i t w} dt \tag{1}$$

is zero when $|w| > B$, where $i = \sqrt{-1}$. Without loss of generality, we consider $B = \frac{1}{2}$ in this paper. It is well-known that $x(t)$ can be expressed as

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \operatorname{sinc}(\pi(t - nT)), \tag{2}$$

provided that $T \leq 1$, cf. [18].

As explained in the introduction, instead of sending x through one single ADC operating at rate $\frac{1}{T}$, we apply r parallel ADCs with the same sampling rate $\frac{1}{rT}$, but with different phases, in order to achieve an overall sampling rate of $\frac{1}{T}$, cf. also Fig. 1. However, due to the existence of time errors and gain mismatches, for each of the k-th ADCs we receive a sequence of uniform samples

$$y_k(n) = (1 + g_k^*) z(nrT + (k - 1 + \delta_k^*)T) \tag{3}$$

for $k = 1, 2, \dots, r \quad n \in \mathbb{Z}$, where $z(t) = x(t) + v(t)$, $v(t)$ is assumed to be Additive White Gaussian Noise (AWGN). Combining all these individual uniform sampling sets into one set results no longer in a sequence of uniformly spaced samples, but in a sequence of *periodic nonuniform samples* of the signal $x(t)$.

In order to obtain a better approximation of the output $\{x(nT)\}_{n\in\mathbb{Z}}$, estimating the time error δ_k^* and gain mismatch g_k^* is necessary. Our goal is to develop estimation methods that are highly accurate, robust, and numerically efficient.

Since we only study the dynamic performance of the r-ADC system, it is not important whether for instance the gain for the k-th channel is $(1 + g_k^*)$ or $\beta(1 + g_k^*)$ as long as β is constant for all ADCs. In other words, we only care about the relative difference of gain mismatches and time errors between different ADCs. Therefore, without lost of generality, we set $\delta_1^* = 0$ and $g_1^* = 0$, $\delta_k^* T$ denotes the timing error between the k-th sampling sequence and the first sampling sequence and g_k^* is the gain mismatch for the k-th ADC.

III. Mismatch detection via separable nonlinear least squares

We start by considering the problem of how to approximate a bandlimited signal well by using known samples. It is well-known that if we reduce the time interval T with which we sample x then the Fourier domain periodization of x is increased to $\frac{1}{T}$. Hence, if $T < 1$, i.e., if we oversample the signal, we can use a filter $\psi(t)$ with fast decay instead of the slowly decaying sinc-function to recover x, cf. [19]. In this case we arrive at the following expression for x

$$x(t) = \sum_{m=-\infty}^{\infty} c_m^* \psi(t - m\tilde{T}), \tag{4}$$

where $\tilde{T} \in [T, 1]$ is the corresponding time spacing which depends on the particular choice for ψ. For instance, when the Fourier transform of $\psi(t)$ is the raised cosine we have

$$\psi(t) = \frac{\tilde{T} \sin(\frac{\pi t}{\tilde{T}})}{\pi t} \frac{\cos(\frac{\pi \alpha t}{\tilde{T}})}{1 - (\frac{2\alpha t}{\tilde{T}})^2}, \tag{5}$$

where α is the roll-off factor.

Now let $y_k(n)$ be the samples we get for $n \in \mathbb{Z}$ and $k = 1, 2, \dots, r$, concerning the time errors and gain mismatches, we arrive at the following nonlinear least squares problem:

$$\{\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}\} = \operatorname{argmin}_{\{\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}\}} \sum_{k=1}^{r} \sum_{n=-\infty}^{\infty} \left| y_k(n) - \Psi_k(n) \right|^2, \tag{6}$$

where $\Psi_k(n) = (1 + g_k) \sum_{m=-\infty}^{\infty} c_m \psi((nr + \delta_k + k - 1)T - m\tilde{T})$ and the vectors $\boldsymbol{\delta}$ and $\boldsymbol{g}$ of length $r-1$ are the estimations for the time errors and gain mismatches, $\boldsymbol{\delta}(k) = \delta_{k+1}$, $\boldsymbol{g}(k) = g_{k+1}$, for $k = 1, \dots r - 1$ since we already assumed that $\delta_1 = 0$ and $g_1 = 0$, and $\boldsymbol{c}(m) = c_m$. In practice it is of course not feasible to first collect all samples of x and then estimate the parameters $\boldsymbol{\delta}$ and $\boldsymbol{g}$, not to mention that it would be also impossible to numerically solve an infinite-dimensional optimization problem. Therefore we need to truncate (6) to a finite dimensional problem.

Assume N, M are even numbers, for given samples $\{y_k(n)\}_{k=1}^{r}$ where $n = \frac{-N}{2}, \dots, \frac{N}{2} - 1$, we approximate the signal $x(t)$ in $t \in [\frac{-rNT}{2}, \frac{rNT}{2}]$ by the truncated series

$$x_e(t) = \sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_m^* \psi(t - m\tilde{T}), \tag{7}$$

where $\frac{rNT}{2} \leq \frac{M\tilde{T}}{2} \leq \frac{rNT}{2} + \tilde{T}$. Note that for $\psi = \operatorname{sinc}$ this series may diverge in presence of noise when $N \to \infty$. Even if it does converge, the rate of convergence will be annoyingly slow. From [20] we know that, unlike (2), for properly chosen $\psi(t)$ in (7) the truncated sum will always converge. Furthermore, the truncation error $\frac{\|x_e - x\|}{\|x\|}$ decays very fast with respect to the number of samples (here, $\|.\|$ denotes the usual Euclidean norm). Thus, assuming we choose a proper ψ we are concerned with the finite-dimensional optimization problem

$$\{\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}\} = \text{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c}\}} \sum_{k=1}^{r} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left| y_k(n) - \tilde{\Psi}_k(n) \right|^2. \quad (8)$$

Where $\tilde{\Psi}_k(n) = (1 + g_k) \sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_m \psi((\delta_k + nr + k - 1)T - m\tilde{T})$. Problem (8) is a *separable nonlinear least squares problem* which was analyzed in [21], [22], [23]. Our approach for solving (8) is a Gauss-Newton type method, which is introduced in the following.

Define the $rN \times (M + 1)$ matrix A by $A(i, j) = (1 + g_i)\psi((i + \delta_i - \frac{rN}{2})T - (j - \frac{M}{2})\tilde{T})$, where $g_1 = 0$, $g_i = g_{i-r}$ when $i > r$; $\delta_1 = 0$, $\delta_i = \delta_{i-r}$ when $i > r$; $i = 1, 2, \ldots, rN$ and $j = 1, , \ldots, M + 1$. We reorganize the rN samples $\{y_k(n)\}_{k=1}^{r}$ by the real sampling time from $\frac{-rNT}{2}$ to $\frac{rNT}{2}$, then put them into vector $\boldsymbol{y}$ of length rN. Now problem (8) becomes

$$\{\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}\} = \text{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c}\}} \|A\boldsymbol{c} - \boldsymbol{y}\|^2. \quad (9)$$

We solve (8) by using an iterative method. We use initial values $\boldsymbol{\delta}^{(0)}(k) = 0$ and $\boldsymbol{g}^{(0)}(k) = 0$, $k = 1, 2, \ldots r - 1$. From (9), we can get the initial value $\boldsymbol{c}^{(0)}$ by solving a linear least squares problem.

$$\{\boldsymbol{c}^{(0)}\} = \text{argmin}_{\{\boldsymbol{c}\}} \|A^{(0)}\boldsymbol{c} - \boldsymbol{y}\|^2. \quad (10)$$

where $A^{(0)}(i, j) = \psi((i - \frac{rN}{2})T - (j - \frac{M}{2})\tilde{T})$.

We collate the solutions $\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}$ of (9) in the vector $\boldsymbol{\gamma} = (\boldsymbol{\delta}^{\boldsymbol{T}}, \boldsymbol{g}^{\boldsymbol{T}}, \boldsymbol{c}^{\boldsymbol{T}})^T$ and introduce the $rN \times 1$-vector-valued function $F(\boldsymbol{\gamma}) := A\boldsymbol{c}$. The linearization of the nonlinear function F at the exact solution $\boldsymbol{\gamma}^{(*)}$ is given by

$$F(\boldsymbol{\gamma}) \approx F(\boldsymbol{\gamma}^{(*)}) + J(\boldsymbol{\gamma}^{(*)})(\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(*)}), \quad (11)$$

with the Jacobian $J(\boldsymbol{\gamma})_{i,j} := \frac{\partial F_i}{\partial \boldsymbol{\gamma}_j}$, $i = 1, 2, \ldots rN$; $j = 1, 2, \ldots M + 2r - 2$. Thus in each iteration step we solve the linear least squares problem

$$\{\boldsymbol{\gamma}^{(m)}\} = \text{argmin}_{\boldsymbol{\gamma}} \|F(\boldsymbol{\gamma}^{(m-1)}) + J(\boldsymbol{\gamma}^{(m-1)})(\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(m-1)}) - \boldsymbol{y}\|^2 \quad (12)$$

with starting value $\boldsymbol{\gamma}^{(0)} = ((\boldsymbol{\delta}^{(0)})^{\boldsymbol{T}}, (\boldsymbol{g}^{(0)})^{\boldsymbol{T}}, (\boldsymbol{c}^{(0)})^{\boldsymbol{T}})^T$.

Algorithm 3.1: Given the optimization problem (9), and the corresponding vector function $F(\boldsymbol{\gamma})$ and matrix $J(\boldsymbol{\gamma})$, starting at $\boldsymbol{\gamma}^{(0)} = ((\boldsymbol{\delta}^{(0)})^{\boldsymbol{T}}, (\boldsymbol{g}^{(0)})^{\boldsymbol{T}}, (\boldsymbol{c}^{(0)})^{\boldsymbol{T}})^T$, we solve this problem by the following algorithm:

1) At the m-th step, we solve linear least square problem (12). to find $\boldsymbol{\gamma}^{(m)}$.
2) Let $m := m + 1$.
3) Stop if m is greater than n_1, otherwise go to the next step.

It is well known that the Gauss-Newton method has guaranteed convergence, provided that $\{\boldsymbol{\gamma} \| \|F(\boldsymbol{\gamma}) - \boldsymbol{y}\| \leq \|F(\boldsymbol{\gamma}^{(0)}) - \boldsymbol{y}\|\}$ is bounded and the Jacobian $J(\boldsymbol{\gamma})$ has full rank [24]. Moreover, if the truncation error is small and the SNR is high, this method has superlinear convergence [24].

To achieve an even better estimation of $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*\}$, we may solve problem (8) multiple times by using K consecutively disjoint sampling blocks of size rN. One natural way is to average all these solutions, which gives the following estimates

$$\tilde{\boldsymbol{\delta}}^* = \frac{1}{K} \sum_{l=1}^{K} \tilde{\boldsymbol{\delta}}^{\boldsymbol{l}} \quad \text{and} \quad \tilde{\boldsymbol{g}}^* = \frac{1}{K} \sum_{l=1}^{K} \tilde{\boldsymbol{g}}^{\boldsymbol{l}}. \quad (13)$$

Here $\tilde{\boldsymbol{g}}^{\boldsymbol{l}}$ and $\tilde{\boldsymbol{\delta}}^{\boldsymbol{l}}$ are the solution of (8) for the l-th data block.

Instead of using (13), we use another method which exhibits even better performance. We simultaneously consider K data blocks as before, but now we set up the following optimization problem:

$$\text{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1},\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K} \sum_{k=1}^{r} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left| y_{lkn} - (1 + g_k)\tilde{\Psi}_{lk}(n) \right|^2 \quad (14)$$

where y_{lkn} is the n-th sampling value of the k-th ADC in the l-th data block, and

$$\tilde{\Psi}_{lk}(n) = (1 + g_k) \sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_{lm}\psi_l((\delta_k + nr + k - 1)T - m\tilde{T}),$$

usually if t_{l1} and t_{lrN} are the ideal time positions of first and the last points in the l-th data block, we define $\psi_l(t) = \psi(t - \frac{t_{l1} + t_{lrN}}{2})$, vector $\boldsymbol{c}_l$ with components $\boldsymbol{c}_l(m) = c_{lm}$ for $l = 1, \ldots, K$ and $m = \frac{-M}{2}, \ldots, \frac{M}{2}$.

Let the vector $\boldsymbol{y}_l$ contain all the samples in the l-th data block. Define matrix A_l by $A_l(i, j) = (1 + g_i)\psi_l((\delta_i + i - \frac{rN}{2})T - (j - \frac{M}{2})\tilde{T})$, where $g_1 = 0$ and $g_i = g_{i-r}$ when $i > r$, $\delta_1 = 0$ and $\delta_i = \delta_{i-r}$ when $i > r$ for $i = 1, 2 \ldots, rN$ and $j = 1, 2, \ldots, M + 1$, problem (14) becomes:

$$\{\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c_1}, \ldots, \boldsymbol{c_K}\} = \text{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1}\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K} \|A_l\boldsymbol{c_l} - \boldsymbol{y}_l\|^2. \quad (15)$$

Before we proceed, we introduce some notation. In the l-th data block, let $F_l(\boldsymbol{\gamma}) = A_l\boldsymbol{c_l}$, and define the $rN \times (r - 1)$ matrices Δ_l and G_l via $\Delta_l(i, j) = \frac{\partial F_{li}}{\partial \delta_j}$ and $G_l(i, j) = \frac{\partial F_{li}}{\partial g_j}$ for $j = 1, \ldots, r - 1$. Then the $rN \times (M + 1)$ matrix A_l has components $A_l(i, j) = \frac{\partial F_{li}}{\partial \boldsymbol{c}_{l\,j}}$ for $j = 1, \ldots, M + 1$.

In order to solve the nonlinear least squares problem (15), we use the same linearization as in (11) for different data blocks, and make $\tilde{\boldsymbol{\gamma}}^l = (\tilde{\boldsymbol{\delta}}^{\boldsymbol{l}^T}, \tilde{\boldsymbol{g}}^{\boldsymbol{l}^T}, \tilde{\boldsymbol{c}}^{\boldsymbol{l}^T})^T$ the initial value, which is the solution from the iteration method (12) in l-th data block. By using the approximation in (11), we arrive at the following problem

$$\text{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1},\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K} \|F_l(\tilde{\boldsymbol{\gamma}}^l) + J_l(\tilde{\boldsymbol{\gamma}}^l)(\boldsymbol{\gamma}_l - \tilde{\boldsymbol{\gamma}}^l) - \boldsymbol{y}_l\|^2, \quad (16)$$

where $J_l(\boldsymbol{\gamma})_{i,j} := \frac{\partial F_{li}}{\partial \boldsymbol{\gamma}_j}$ and $\boldsymbol{\gamma}_l = (\boldsymbol{\delta}^T, \boldsymbol{g}^T, \boldsymbol{c_l}^T)^T$. Actually from the definitions above we have

$$J_l(\boldsymbol{\gamma}) = [A_l, \Delta_l, G_l]. \quad (17)$$

In problem (16), if $\boldsymbol{\delta}$ and $\boldsymbol{g}$ are given, the $\boldsymbol{c_l}$ can be determined by solving the linear least squares problem $\{\boldsymbol{c_l}\} = \text{argmin}_{\boldsymbol{c_l}} \|F_l(\tilde{\gamma}^l) + J_l(\tilde{\gamma}^l)(\gamma_l - \tilde{\gamma}^l) - \boldsymbol{y}_l\|^2$. Since we are only interested in the solution of $\boldsymbol{\delta}$ and $\boldsymbol{g}$ in (16), we formulate our problem as:

$$\begin{aligned}\{\boldsymbol{\delta}, \boldsymbol{g}\} &= \text{argmin}_{\{\boldsymbol{\delta}, \boldsymbol{g}\}} \min_{\{\boldsymbol{c_1}, \ldots, \boldsymbol{c_K}\}} \sum_{l=1}^{K} \|F_l(\tilde{\gamma}^l) + A_l(\tilde{\gamma}^l)(\boldsymbol{c_l} \\ &\quad -\tilde{\boldsymbol{c}}^l) + \Delta_l(\boldsymbol{\delta} - \tilde{\boldsymbol{\delta}}^l) + G_l(\boldsymbol{g} - \tilde{\boldsymbol{g}}^l) - \boldsymbol{y}_l\|^2\end{aligned}$$

Let $H_l = [\Delta_l, G_l] - P_{A_l}([\Delta_l, G_l])$, where P_{A_l} is the orthogonal projection to space A_l. We have

$$\begin{aligned}\min\ &{}_{\{\boldsymbol{c_1}, \ldots, \boldsymbol{c_K}\}} \sum_{l=1}^{K} \|F_l(\tilde{\gamma}^l) + A_l(\tilde{\gamma}^l)(\boldsymbol{c_l} - \tilde{\boldsymbol{c}}^l) \\ &+\Delta_l(\boldsymbol{\delta} - \tilde{\boldsymbol{\delta}}^l) + G_l(\boldsymbol{g} - \tilde{\boldsymbol{g}}^l) - \boldsymbol{y}_l\|^2 \\ = &\sum_{l=1}^{K} \min_{\{\boldsymbol{c_l}\}} \|F_l(\tilde{\gamma}^l) \\ &+A_l(\tilde{\gamma}^l)(\boldsymbol{c_l} - \tilde{\boldsymbol{c}}^l) + \Delta_l(\boldsymbol{\delta} - \tilde{\boldsymbol{\delta}}^l) + G_l(\boldsymbol{g} - \tilde{\boldsymbol{g}}^l) - \boldsymbol{y}_l\|^2 \\ = &\sum_{l=1}^{K} \|H_l \begin{pmatrix} \boldsymbol{\delta} - \tilde{\boldsymbol{\delta}}^l \\ \boldsymbol{g} - \tilde{\boldsymbol{g}}^l \end{pmatrix} + \boldsymbol{y}_l - F_l(\tilde{\gamma}^l) - P_{A_l}(\boldsymbol{y}_l \\ &-F_l(\tilde{\gamma}^l))\|^2 \\ = &\sum_{l=1}^{K} \|H_l \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{g} \end{pmatrix} - f_l\|^2\end{aligned}$$

where $f_l = F_l(\tilde{\gamma}^l) + P_{A_l}(\boldsymbol{y}_l + F_l(\tilde{\gamma}^l)) - \boldsymbol{y}_l + H_l((\tilde{\boldsymbol{\delta}}^l)^T, (\tilde{\boldsymbol{g}}^l)^T)^T$. Thus problem (16) becomes

$$\{\boldsymbol{\delta}, \boldsymbol{g}\} = \text{argmin}_{\{\boldsymbol{\delta}, \boldsymbol{g}\}} \sum_{l=1}^{K} \|H_l \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{g} \end{pmatrix} - f_l\|^2. \qquad (18)$$

Let $H = \sum_{l=1}^{K} H_l^T H_l$ and $f = \sum_{l=1}^{K} H_l^T f_l$, the solution of problem (18) is given by $(H)^{-1} f$.

As we said before the simulation results by using (18) show better performance than by using (13). One intuitive explanation is that in (18), the essential solution depends on the matrix H, which reflects the condition of the resulting matrix H_l in all data blocks, while in (13), the final results depend on the condition of the resulting matrix individually.

By solving (18) we arrive at our final estimation for the time errors and gain mismatches.

Algorithm 3.2: Given an r-ADC system, if in each l-th data block the resulting matrix H_l has full rank, we have the following algorithm for estimating the timing errors and gain mismatches:

Let H be the $(2r-2) \times (2r-2)$ zero-matrix and f be a zero vector of length $2r-2$.

1) For the l-th data block, we first solve (9) by using Algorithm 3.1, then we compute the matrix H_l and the vector f_l in (18).
2) Let $H := H + H_l^T H_l$ and $f := f + H_l^T f_l$.
3) Finally we obtain the estimates for the timing and gain mismatch by solving

$$\begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{g} \end{pmatrix} = (H)^{-1} f. \qquad (19)$$

Details of implementation are provided in Sections VII and VIII. Here we just notice that we typically set $n_1 = 4$ in Algorithm 3.1 and the total number of data blocks K about 20.

IV. Convergence analysis

One might hope that the original problem (7) has a unique solution and no local minimums except for one global minimum. Unfortunately (7) is a nonlinear non convex problem with potentially multiple local minima. Thus we can only except that there is a local minimum nearby the true solution and our algorithm can converge to it by properly choosing the initial value $\boldsymbol{\gamma}^{\mathbf{0}}$. Theoretical analysis and simulation results give an affirmative answer to this expectation.

From (7) we know that in the interval $[\frac{-rNT}{2}, \frac{rNT}{2}]$ the truncation error $x_c = x - x_e$ is

$$x_c = \sum_{|m| > \frac{M}{2}} c_m^* \psi(t - m\tilde{T}). \qquad (20)$$

For a properly chosen filter $\psi(t)$ the truncation error $\frac{\|x_c\|}{\|x_e\|}$ goes to 0 rapidly as N becomes large [16]. Since we can control the number of samples, we can make the truncation error very small by using large N. When the truncation error is small and the SNR is high, it is reasonable to expect that problem(8) has a local minimum which is close to $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*\}$. The following theorem confirms this.

Define the function $\Phi(\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c}) := \sum_{k=1}^{r} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left| y_k(n) - (1 + g_k) \sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_m \psi((\delta_k + nr + k - 1)T - m\tilde{T}) \right|^2$. At any of its local minimum $(\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c})$, it is obvious that

$$\begin{aligned}\boldsymbol{c} &= \text{argmin}_{\{\boldsymbol{c}\}} \sum_{k=1}^{r} \sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \Big| y_k(n) - \\ &\quad (1 + g_k) \sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_m \psi((\delta_k + nr + k - 1)T - m\tilde{T}) \Big|^2\end{aligned}$$

Theorem 4.1: In each data block, if the matrix $\mathbf{J}$ has full rank, the function $\Phi(\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c})$ has a local minimum given by $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\}$+ $(J^T J)^{-1} J^T (\boldsymbol{v} + x_c)$+$\mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2} + \frac{\|x_c\|^2}{\|x_e\|^2})$. A proof of this theorem and the next theorem will be given in the Appendix.

Remark: If J does not have full rank, we do not have a local minimum around γ^*. If the condition number of J is too large, the second order term may not be negligible, thus, it is not guaranteed that we have a local minimum around γ^*.

For nonlinear problems we may have more than one local minimum. If the initial value is close to $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*\}$ we have

Proposition 4.2 ([24]): If the initial value of (8), $\boldsymbol{\gamma}^{(\mathbf{0})}$ is close to the local minimum

$$\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\} + (J^T J)^{-1} J^T (\boldsymbol{v} + x_c) + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2} + \frac{\|x_c\|^2}{\|x_e\|^2}) \qquad (21)$$

then the Gauss-Newton method (11) will converge to (21) with superlinear convergence rate.

By using multiple data blocks simultaneously, we achieve a better approximation, actually we have

Theorem 4.3: If in the l-th data block, the iteration method converges to the local minimum

$$\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\} + (J_l^T J_l)^{-1} J_l^T (\boldsymbol{v} + x_c) + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2} + \frac{\|x_c\|^2}{\|x_e\|^2}), \tag{22}$$

the solution of (13) is

$$\{\boldsymbol{\delta}^*, \boldsymbol{g}^*\} + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2} + \frac{\|x_c\|^2}{\|x_e\|^2}) \tag{23}$$

with probability 1 when $K \to \infty$.

Remark: From the theoretical analysis we can see, if the matrix J_l is ill conditioned , this method converges slowly.

Theorem 4.1 tells us that if we use one data block (no matter how many samples it has), our estimation has an error at the order of $\mathcal{O}(\|\boldsymbol{v}\|)$. By considering multiple data blocks simultaneously as in (13), we can make the estimation error of the order $\mathcal{O}(\|\boldsymbol{v}\|^2)$. This is the reason why we do not set up the optimization problem by considering all the K data blocks together as one big data block. Of course, separating all the available samples into K data block will introduce more truncation error, but by choosing proper filters, we can make the truncation error negligible.

V. Filter selection strategy

In this section, we discuss some practical strategies for how to choose a set of proper filters $\{\psi(t - m\tilde{T})\}_m$ in equation (4). Recall that the functions $\psi(t - m\tilde{T})$ determine the entries of the matrix appearing in the optimization problem.

We have seen in section III that by taking a very large number of samples in each data block, we can reduce the truncation error. But in practice the number of samples that can be stored or processed is limited, therefore it is desirable if a small number of samples would suffice. Since the filter $\psi(t)$ decays fast in equation (20), only a few terms of $\psi(t - m\tilde{T})$, for which $|m|$ is close to $\frac{M}{2}$, contribute most to the truncation error. This means we should use the following expression to approximate $x(t)$ in the interval $[\frac{-rNT}{2}, \frac{rNT}{2}]$

$$x_e(t) = \sum_{m=-\frac{M}{2}-r_1}^{\frac{M}{2}+r_1} c_m \psi(t - m\tilde{T}), \tag{24}$$

where r_1 is a small number. Since adding more basis functions leads to an ill-conditioned matrix J and thus enhances noise effects, in practice we usually choose $r_1 \leq 4$.

It is a well-known fact that oversampling in equation (4) leads to a reduction of the noise energy by a factor of $\frac{T}{\tilde{T}}$. On the other hand, while larger $\tilde{T}$ means better noise reduction, a larger $\tilde{T}$ also implies that the decay of the filter $\psi(t)$ becomes slower, which in turn will introduce more truncation error. To balance this trade-off, for given SNR and sampling rate T, we choose $\tilde{T}$ by solving the following optimization problem:

$$\{\tilde{T}\} = \text{argmin}_{T \leq \tilde{T} \leq 1} \mathcal{E}\|x(t) - x_e(t)\|^2), \tag{25}$$

where $\mathcal{E}$ denotes expectation, x_e is given by (24). For $\psi(t)$ we can choose the raised cosine filter given in (5). This problem is independent of the input signal and our original problem (9), and can be solved beforehand.

Example 5.1: For $T = \frac{1}{2.6}$ and randomly given bandlimited signal $x(t)$, we have computed the interpolation errors by using expression (24) with $r_1 = 4$. For different $\tilde{T}$, numerical results are shown in fig. 2.

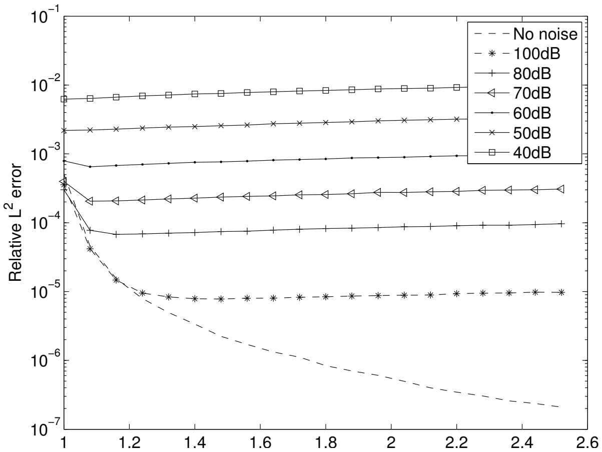

Fig. 2. Interpolation errors by using with different $\frac{1}{\tilde{T}}$

In Fig. 2, the units on x-axis are in terms of $\frac{1}{\tilde{T}}$. From the simulation results we see that in general neither $\tilde{T} = 1$ nor $\tilde{T} = T$ is the best choice. For certain SNR, there is an optimal solution for $\tilde{T}$. In order to have a smaller approximation error, when SNR is low, we should choose larger $\tilde{T}$; When SNR is large, we should choose smaller $\tilde{T}$.

VI. Non-blind calibration

Non-blind calibration refers to determining the timing error and gain mismatch based on sending a training signal [8], [9]. A known signal $s(t)$, usually sinusoidal input, is sent through these ADCs. Then, given the known samples $\{y_k(n)\}$, where $y_k(n) = (1 + g_k^*)(x(nrT + (k - 1 + \delta_k^*)T + v_k(n))$ for $k = 1, 2, \ldots, r$ and $n \in \mathbb{Z}$. We have the following estimation problem

$$\begin{aligned} \{\boldsymbol{\delta}, \boldsymbol{g}\} \quad = \quad & \text{argmin}_{\{\boldsymbol{\delta}, \boldsymbol{g}\}} \sum_{k=1}^{r} \sum_{n \in \mathbb{Z}} |y_k(n) - \\ & (1 + g_k) s(nrT + (k - 1 + \delta_k)T)|^2. \end{aligned}$$

In its finite mode, let $\gamma = (\boldsymbol{\delta^T}, \boldsymbol{g^T})^T$ and the vector F have components $F(i) = (1 + g_i)s((i - 1 - \frac{rN}{2})T + \delta_n T)$ for $i = 1, 2, \ldots rN$. Where $g_1 = 0$, $\delta_1 = 0$ and $g_{i+r} = g_i$,$\delta_{i+r} = \delta_i$. The linearization of the nonlinear function F at γ^* is given by

$$F(\gamma) \approx F(\gamma^{(*)}) + J(\gamma^{(*)})(\gamma - \gamma^{(*)}), \tag{26}$$

with $rN \times (2r - 2)$, matrix $J(\gamma)_{i,j} := \frac{\partial F_i}{\partial \gamma_j}$, $i = 1, 2, \ldots rN$ and $j = 1, 2, \ldots, 2r - 2$. In each iteration we solve the linear least squares problem

$$\{\gamma^{(m)}\} = \text{argmin}_\gamma \|F(\gamma^{(m-1)}) + J(\gamma^{(m-1)})(\gamma - \gamma^{(m-1)}) - \boldsymbol{y}\|. \tag{27}$$

This is a similar iteration method as (12), however, there is a huge difference between the J matrix in these two schemes. If we use the similar notation as in (17), the matrix $J = H$, is always a well-conditioned matrix. This makes the solution of non-blind calibration much easier than blind calibration. In addition, the computational complexity is reduced, since the size of J becomes only $rN \times (2r-2)$ compared to $rN \times (M + 2r - 2)$ for blind calibration. The iteration methods introduced in (13) and (18) are both applicable and numerical simulations show superfast convergence.

VII. Signal reconstruction by interpolation and fast computation

A. Signal Reconstruction

After we have estimated the time errors and gain mismatches, we use raised cosine filters in the interpolation step to evaluate the signal value at the desired uniform sampling points from the periodic nonuniform samples. This can be efficiently accomplished with the algorithm proposed in [16]. The optimized value for $\tilde{T}$, computed in Section V is also used in the interpolation method. Considering the boundary effect, we first take a large interval to do the interpolation, and then only utilize the solution in the middle of the interval.

But there is one more problem which needs to be addressed. How does the approximation error introduced during the interpolation affect the final result? We present the following experiment to illustrate the interpolation method and explore the relationship between the estimation error and the final output.

Example 7.1: The input signal are randomly given in a 4-ADC system with sampling rate 1.8. The normalized time errors are $[0, 0.05, -0.025, 0.03]$ and the gains are $[1, 1.03, 0.98, 1.01]$. For different SNR level, we use different $\tilde{T}$. We generate the estimation error as a Gaussian random variable. Fig. 3 shows the numerical results.

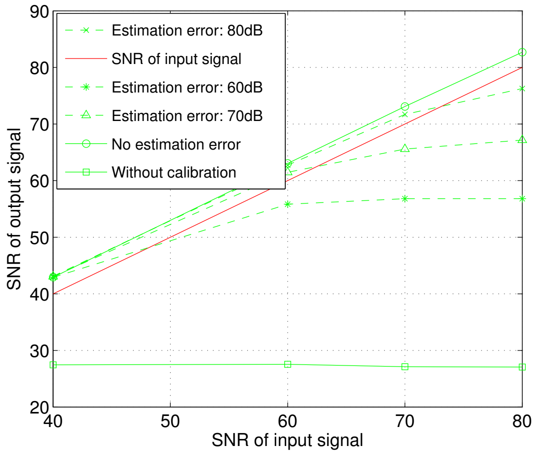

Fig. 3. The effects of different estimation errors to the SNR of output signal

From the results in last experiment, if the estimation error is zero, due to the oversampling, the SNR of the output is higher than the SNR of the input signal. In general, if the approximation error is in the order of $\frac{1}{10}$ of the noise to signal ratio (NSR) of the output, the effect of estimation error is negligible, which also means for given SNR, when our estimation algorithms has reached a certain accuracy, we can stop. If the estimation error is higher than the NSR, it significantly affects the SNR at the output signal. In the next section, we will show the efficiency of our detector through different experiments.

B. Fast Computation

Fast FFT-based multiplication and the conjugate gradient (CG) method [25] are applied to both mismatches estimation and signal reconstruction. For an r-ADC system, the N samples in each data block have a periodic nonuniform structure. By using this structure, we can apply fast multiplication of the $rN \times (M+1)$ matrix A by any vector y of length $M+1$, where $A(i,j) = (1+g_i)\psi((i+\delta_i-\frac{rN}{2})T-(j-\frac{M}{2})\tilde{T})$ for $i = 1, 2, \ldots, rN$ and $j = 1, 2, \ldots, M+1$.

Consider a matrix $B(g,\delta)(i,j) = (1+g)\psi((i+\delta-\frac{rN}{2})T-(j-\frac{M}{2})\tilde{T})$, for any given g and δ. Since we can choose any $\tilde{T} \in [T, 1]$, here we try to select $\tilde{T}$ such that $\frac{T}{\tilde{T}-T} \in \mathbb{Q}$. Thus we can always construct another Toeplitz matrix D such that matrix B is only a submatrix of D, the fast multiplication of B with vector y can be implemented by $D\bar{y}$, where $\bar{y}$ is a zero padding version of vector y.

For example, when the oversampling rate is 1.4, which mean $T = \frac{1}{1.4}$, we may choose $\tilde{T} = 1.3T$. Thus if $\bar{T} = 0.1T$ and the Toeplitz matrix D has components $D(i,j) = (1+g)\psi(i\bar{T} + (\delta - \frac{rN}{2})T - j\bar{T} + \frac{M}{2}\tilde{T})$, the matrix B is a submatrix of D and $B(i,j) = D(10i, 13j)$. For any vector y, let $\bar{y}$ have components $\bar{y}(13i) = y(i)$ and its other components be 0 for $i = 1, 2, \ldots, 13rN$. Obviously multiplication of By can be implemented by $D\bar{y}$. $D\bar{y}$ could be carried out by using about $40rN\log(10rN)$ computations. Thus A can be splitted into r sub-matrices, and the k-th sub matrix is a sub-matrix of a Toeplitz matrix $D(g_k, \delta_k)$. Hence Ay can be computed in $\mathcal{O}(rN\log(rN))$ operations.

Considering the fast computation issue, when we solve the problem (25), we may be interested in some solution $\tilde{T}$ such that the total computation is affordable. We should find the proper $\tilde{T}$ whenever we know the sampling rate $\frac{1}{T}$.

For the mismatch detection method in blind calibration mode, suppose we use K blocks totally, the computational costs are $\mathcal{O}(KrNlog(rN))$. When applied in non-blind mode, our method needs only $rN(2r-2) + \mathcal{O}(M(2r-2)^2)$ computations since the size of matrix J becomes $rN \times (2r-2)$. After we have the estimations, at the output, we need another $\mathcal{O}(rNlog(rN))$ computations for each data block with size N to find the correct sampling values at the ideal sampling times.

VIII. Simulations and analysis

A. Results for blind calibration

Several experiments have been done to evaluate the performance of our algorithms. From Theorem 4.1, 4.3 and Proposition 4.2 we know that the convergence of our approach is mostly effected by the condition number of the matrix J, the noise and truncation error, and the nonlinear distortion of the original problem.

In general, higher oversampling rate results in a thinner matrix which will have a smaller condition number. We did some experiments for an 8-ADC system to test how oversampling affects the convergence of our method. Since the length of $\boldsymbol{g}$ and $\boldsymbol{\delta}$ are $r-1$, in this section, we measure the average estimation error for timing errors by $\frac{\|\boldsymbol{\delta}-\boldsymbol{\delta}^*\|}{\sqrt{r-1}}$ and the estimation error for gain mismatches by $\frac{\|\boldsymbol{g}-\boldsymbol{g}^*\|}{\sqrt{r-1}}$.

Example 8.1: The input signal is bandlimited WGN with bandwidth $B = 1/2$. We choose different oversampling rates, 1.2, 1.6, and 2.4. The time errors and gain mismatches are uniformly distributed in $[-0.1T, 0.1T]$ and $[-0.1, 0.1]$ respectively. The SNR of the input signal is 70dB. We apply Algorithm 3.2 with the following parameters. The number of samples in each data block is 320 and we use a total of 30 data blocks in each experiment. Each experiment is repeated 50 times and the corresponding average errors are shown in fig. 4.

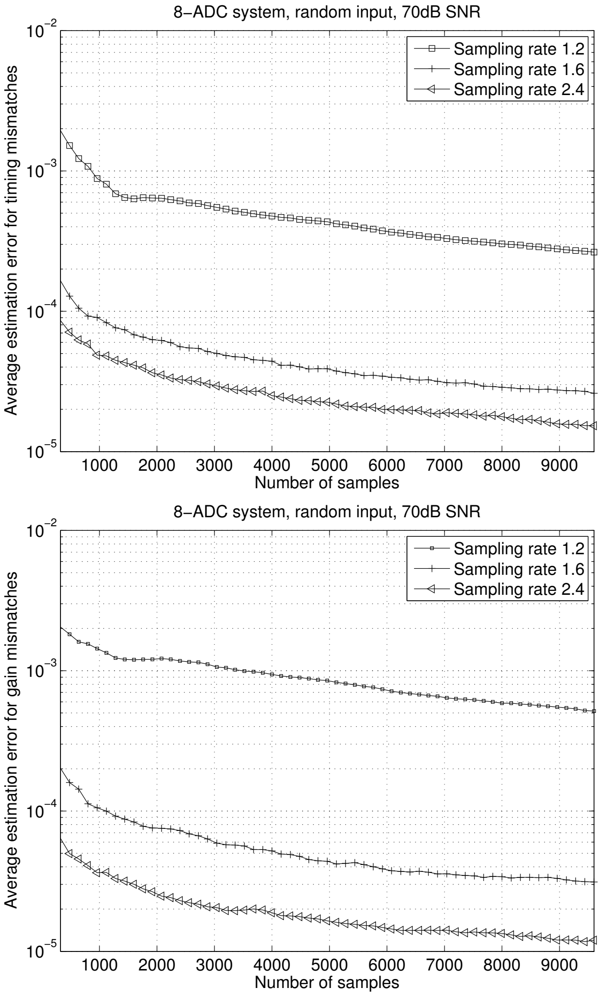

Fig. 4. Estimation errors for different oversampling rate in a 8-ADC system

For higher oversampling rate, Algorithm 3.2 gives smaller estimation error. In general, for the same oversampling rate, the estimation errors for time and gain mismatches are of the same order.

From Example 7.1 and the above example we conclude that higher oversampling rate can give smaller estimation error as well as lead to better noise reduction in the interpolation step.

In order to test the effects for different SNRs, we did the following experiment.

Example 8.2: We consider an 4-ADC system, the input signal consists of multiple sinusiods with frequencies $0.3, 0.52, 0.6$ and 0.94. The sampling rate is 1.4. The time errors and gain mismatches are uniformly distributed in $[-0.1T, 0.1T]$ and $[-0.1, 0.1]$ respectively. The number of samples in each data block is 160 and we use a total of 20 data blocks in each experiment. Each experiment is repeated 50 times and the average time mismatches estimation errors are shown in fig. 5 As expected, the estimation error decreases with increasing SNR. For each SNR, our method can reach an estimation with an error less than NSR. The numerical results for gain mismatches are similar.

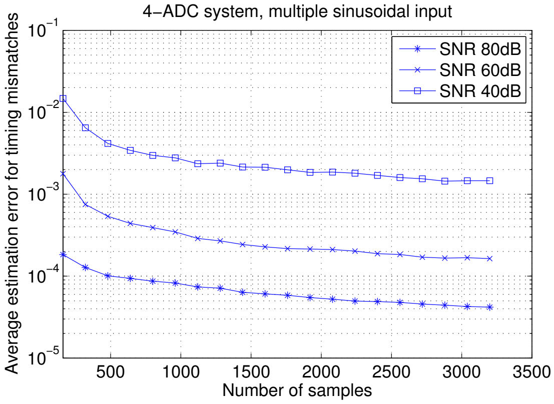

Fig. 5. Estimation errors for a 4-ADC system with blind calibration

We have shown that our approach can detect the mismatch errors very well. In the next example, we combine our detection approach and the interpolation method introduced in section VII to evaluate the overall performance of our method at the output.

Example 8.3: We consider an 8-ADC and a 16-ADC system. The input signal is bandlimited WGN with bandwidth $B = 1/2$, sampling rates are 1.4 and 1.8 respectively. The time errors and gain mismatches are uniformly distributed in $[-0.1T, 0.1T]$ and $[-0.1, 0.1]$ respectively. The number of total samples for mismatch errors detection is 9600. We first estimate the mismatch errors, after that we apply the interpolation method of Subsection VII-A to get the output signal. The SNR is computed for the output signal. We did the same experiment 100 times and the average SNRs are shown in fig. 6.

For the 8-ADC and the 16-ADC system, the SNR of the output signal is larger or only a little bit smaller than the SNR of the input signal when the input SNR is between 50dB and 80dB.

B. Results for non-blind calibration

For non-blind calibration the oversampling is not necessary for detecting the mismatch errors. As we analyzed in section

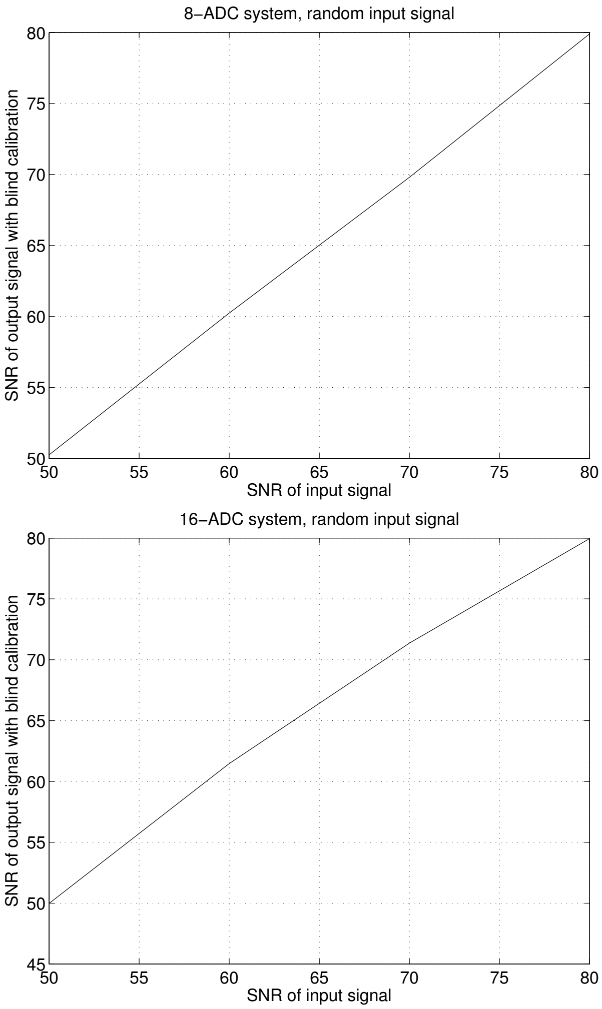

Fig. 6. SNR of output signal with blind calibration in 8-ADC and 16-ADC systems

VI, the problem is much more stable and robust than for blind calibration.

Example 8.4: We consider a 64-ADC system, the training signal is $\sin(0.8\pi t)$ with sampling rate 1. The time errors and gain mismatches are uniformly distributed in $[-0.1T, 0.1T]$ and $[-0.1, 0.1]$ respectively. The number of samples in each data block is 160 and we use a total of 30 data blocks in each experiment. We did the same experiment 50 times and the average estimation errors are shown in fig. 7

From this plot we see that our method achieves very high accuracy for a large range of SNR values, even with less than 1000 samples and without oversampling. Similar results can be achieved for an 128-ADC system or even for an 256-ADC system.

Example 8.5: We consider a 128-ADC system, the training signal is $\sin(0.8\pi t)$ with sampling rate 1. The time errors and gain mismatches are uniformly distributed in $[-0.1T, 0.1T]$ and $[-0.1, 0.1]$ respectively. The number of total samples for mismatch errors detection is 2400. We first estimate the time and gain mismatches then apply this estimation to a 128ADC system with random input signal and sampling frequency 1.1.

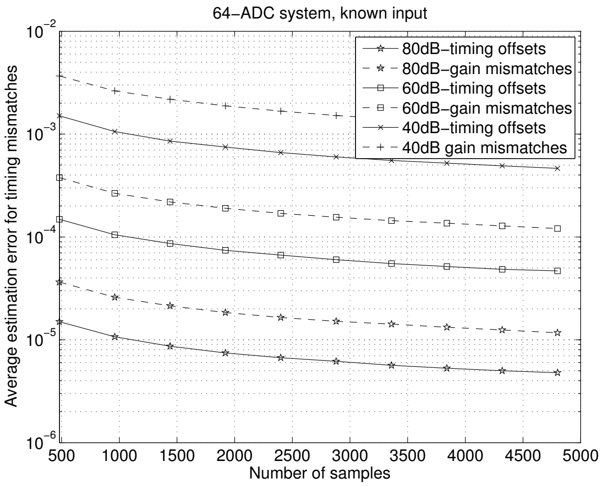

Fig. 7. Average estimation errors in 64ADC system with non-blind calibration

The SNR of output signal is computed for different SNR level. The same experiment is done 50 times and the average SNR is shown in fig. 8.

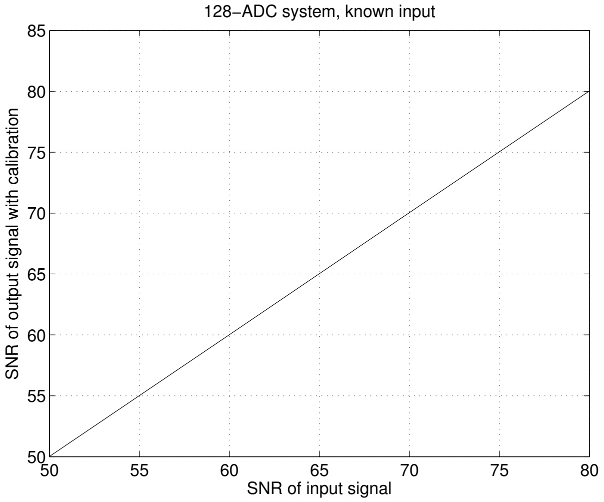

Fig. 8. SNR of output signal with non-blind calibration for 128ADC system

As we can see, the SNR of the output signal is almost the same as the SNR of the input signal with oversampling rate of only 10% for a 128-ADC system!

Compared with blind calibration, non-blind calibration has to inject the training signal periodically, which is its main disadvantage. On the other hand, by using our method, we can make the multiple-ADC system work for a large number of ADCs without oversampling, which highly reduces the sampling frequency of each ADC; We can have a very good estimation with only about 1000 samples, which is a very small number compared to other methods in the literature. Its high accuracy makes it applicable in 14-bit or even higher bit systems.

IX. Conclusion

We have modeled the mismatch errors detection problem for time-interleaved ADCs as a nonlinear separable least squares problem, and proposed a Gauss-Newton type method to find the optimal solution. Theoretical analysis and simulation results show that our approach converges very fast. Our approach can be applied to both blind and non-blind calibration. For blind calibration, our method can achieve an almost error free ADC system with relatively modest oversampling rate in 8-ADC and 16-ADC systems. For non-blind calibration, our method can achieve an error free ADC system even for an 128-ADC system with only 10% oversampling. The computational complexity and memory requirements of the algorithm are low, which makes this method an attractive candidate for various practical applications.

Appendix

Theorem 4.1: In each data block, if the matrix $\mathbf{J}$ has full rank, the function $\Phi(\boldsymbol{\delta}, \boldsymbol{g}, \boldsymbol{c})$ has a local minimum given by $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\}+ (J^TJ)^{-1}J^T(\boldsymbol{v}+x_c)+\mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$.

Proof: If the input signal is (4), and the true timing errors and gain mismatches are $\boldsymbol{\delta}^*, \boldsymbol{g}^*$, then $\boldsymbol{y}_n = (1+g_n)^*(x(rnT+(k-1+\delta_n^*)T)+v(n))$. When we apply Taylor expansion and use the initial value $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\}$ as in (11), we have $F(\gamma) = F(\gamma^*) + J(\gamma^*)(\gamma-\gamma^*) + \mathcal{O}(\|\gamma-\gamma^*\|^2)$. Here F,γ,Q and J are as defined in section III and $\gamma^* = \{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\}$. Now $\Phi(\gamma) = \|J(\gamma^*)(\gamma-\gamma^*) - (x_c+\boldsymbol{v}) + \mathcal{O}(\|\gamma-\gamma^*\|^2)\|$, Φ has a local minimum at $\gamma = \gamma^* + (J^TJ)^{-1}J^T(x_c+\boldsymbol{v}) + \mathcal{O}(\|\gamma - \gamma^*\|^2)$ and $\mathcal{O}(\|\gamma-\gamma^*\|^2) = \mathcal{O}(\|(J^TJ)^{-1}J^T(x_c+\boldsymbol{v})\|^2 = \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$. ■

Theorem 4.3: If in the l-th data block, the iteration method converges to the local minimum

$$\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\} + (J_l^TJ_l)^{-1}J_l^T(\boldsymbol{v}+x_c) + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2}), \tag{28}$$

the solution of (13) is

$$\{\boldsymbol{\delta}^*, \boldsymbol{g}^*\} + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2}) \tag{29}$$

with probability 1 when $K \to \infty$.

Proof: If the solution of the l-th data block is $\{\boldsymbol{\delta}^*, \boldsymbol{g}^*, \boldsymbol{c}^*\}+ (J_l^TJ_l)^{-1}J_l^T(\boldsymbol{v}+x_c)+\mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$, then $\tilde{\boldsymbol{\delta}}^{\boldsymbol{l}} = \boldsymbol{\delta}^* + \boldsymbol{\alpha}_{\boldsymbol{l}} + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$, where $r-1$ by 1 vector $\boldsymbol{\alpha}_{\boldsymbol{j}} = (J_l^TJ_l)^{-1}J_l^T(\boldsymbol{v}+x_c)(1:r-1,1)$. Since the signal is randomly given, the truncation x_c is also a random signal. Thus $E(\boldsymbol{\alpha}_{\boldsymbol{l}}) = \mathbf{0}$, consider another sequence $\bar{\boldsymbol{\delta}}^{\boldsymbol{l}} = \boldsymbol{\delta}^* + \boldsymbol{\alpha}_{\boldsymbol{l}}$, by the strong large number theory, $P(\frac{\sum_{l=1}^K \bar{\boldsymbol{\delta}}^{\boldsymbol{l}}}{K} - \boldsymbol{\delta}^* = 0) = 1$ when $K \to \infty$. Since $\tilde{\boldsymbol{\delta}}^{\boldsymbol{l}} = \bar{\boldsymbol{\delta}}^{\boldsymbol{l}} + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$, we conclude in equation (13), $\tilde{\boldsymbol{\delta}^*} \to \boldsymbol{\delta}^* + \mathcal{O}(\frac{\|\boldsymbol{v}\|^2}{\|x_e\|^2}+\frac{\|x_c\|^2}{\|x_e\|^2})$ with probability 1 when $K \to \infty$. A similar proof holds for $\tilde{\boldsymbol{g}^*}$ in equation (13), which gives the result. ■

References

[1] Black W.C. and Hodges D.A., "Time interleaved converter arrays," *IEEE Journal of Solid-State Circuits*, vol. 15, pp. 1022–1029, Dec. 1980.

[2] Petraglia A. and Mitra S.K., "Analysis of mismatch effects among A/D converters in a time-interleaved waveform digitizer," *IEEE Transactions on Instrumentation and Measurement*, vol. 40, pp. 831–835, Oct. 1991.

[3] J. Tang, Z. Xu, and B.M. Sadler, "Digital receiver for TR-UWB systems with inter-pulse interference," *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop*, pp. 420–424, 5-8 June 2005.

[4] Liuqing Yang and G.B. Giannakis, "Ultra-wideband communications: an idea whose time has come," *Signal Processing Magazine, IEEE*, vol. 21, no. 6, pp. 26–54, Nov. 2004.

[5] Vogel C., "Comprehensive error analysis of combined channel mismatch effects in time-interleaved ADCs," *Instrumentation and Measurement Technology Conference, Proceedings of the 20th IEEE*, vol. 1, pp. 733–738, May 2003.

[6] Conroy C.S.G., Cline D.W., and P.R. Gray, "An 8-b 85-ms/s parallel pipeline A/D converter in 1-νm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 447–454, April 1993.

[7] Dyer K.C., Daihong Fu, Lewis S.H., and Hurst P.J., "An analog background calibration technique for time-interleaved analog-to-digital converters," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 1912–1919, Dec 1998.

[8] Huawen Jin and Lee E.K.F., "A digital-background calibration technique for minimizing timing-error effects in time-interleaved ADCs," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 47, pp. 603–613, July 2000.

[9] Jamal S.M., Fu D., Singh M.P., P. Hurst, and Lewis S.H., "Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 51, pp. 130–139, Jan 2004.

[10] Elbornsson J., Gustafsson F., and J.-E. Eklund, "Blind adaptive equalization of mismatch errors in a time-interleaved A/D converter system," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, pp. 151–158, Jan 2004.

[11] Divi V. and Wornell G., "Signal recovery in time-interleaved analog-to-digital converters," *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 2, pp. 17–21, 2004.

[12] Steven Huang and Bernard C. Levy, "Adaptive blind calibration of timing offset and gain mismatch for two-channel time-interleaved adcs," *IEEE Transactions on Circuits and Systems I*, vol. 53, pp. 1278–1288, June 2006.

[13] Steven Huang and Bernard C. Levy, "Blind calibration of timing offsets for four-channel time-interleaved A/D converters," *Preprint*, 2005.

[14] Jenq Y.-C., "Digital spectra of nonuniformly sampled signals: a robust sampling time offset estimation algorithm for ultra high-speed waveform digitizers using interleaving," *IEEE Transactions on Instrumentation and Measurement*, vol. 39, pp. 71–75, Feb. 1990.

[15] Won Namgoong, "Finite-length synthesis filters for non-uniformly time-interleaved analog-to-digital converter," *IEEE Transactions on Signal Processing*, vol. 4, pp. 815–818, May 2002.

[16] T. Strohmer and J. Tanner, "Fast reconstruction methods for bandlimited functions from periodic nonuniform sampling," *SIAM J. Numer. Anal.*, vol. 44, no. 3, pp. 1073–1094, 2006.

[17] C. Vogel and H. Johansson, "Time-interleaved analog-to-digital converters: Status and future directions," in *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, 2006, pp. 3386–3389.

[18] A.I. Zayed, *Advances in Shannon's Sampling Theory*, CRC Press, Boca Raton, 1993.

[19] J.D. Gibson, *Principles of analog and digital communications*, Prentice Hall, 1993, 2nd ed.

[20] Thomas Strohmer and J. Tanner, "Implementations of shannon's sampling theorem, a time-frequency approach," *Sampling Theory in Signal and Image Processing*, vol. 4, no. 1, pp. 1–17, 2005.

[21] Gene Golub and Victor Pereyra, "Separable nonlinear least square: the variable projection method and its appliations," *Institute of Physics Publishing*, pp. R1–R26, Feb 14,2003.

[22] G.H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM J. Numer. Anal.*, vol. 10, pp. 413–432, 1973.

[23] Golub G H, Pereyra V, and Scolnik H D, "Least squares estimation for a class of nonlinear models," *Technometrics*, vol. 15, pp. 209–218, 1973.

[24] K. Madsen, H.B. Nielsen, and O. Tingleff, *Methods for Non-linear Least Squares Problems*, Informatics and Mathematical Modelin Technical University of Denmark; 2 edition, April, 2004.

[25] R. Chan and M. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Review*, vol. 38, no. 3, pp. 427–482, 1996.