Fast Algorithms for Blind Calibration in Time-Interleaved Analog-to-Digital Converters

Thomas Strohmer and Jiadong Xu
Department of Mathematics
University of California, Davis
email: strohmer, jiadong@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 blind calibration is applicable to any number of time-interleaved ADCs and requires only modest oversampling. Simulation results show fast convergence and desirable detection accuracy. After the mismatch errors detection, the resulting signal to noise ratio (SNR) of the output signal is shown to be higher than the SNR of the input signal in a 16-ADC system.**

***Index Terms*— Analog-digital conversion, Calibration, Nonlinear estimation, Signal sampling, Least squares methods.**

I. INTRODUCTION

The ever-increasing demand for higher data rate communication applications requires high-speed and high-resolution ADCs. Such ADCs can be achieved by employing a time-interleaved architecture, which has attracted considerable attention in recent years [1], [2], [3].

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 [4], 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 [2], [5]. 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 [6].

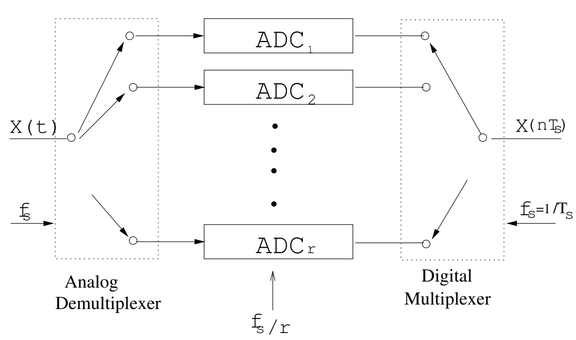

Fig. 1. Block diagram of the time-interleaved ADCs.

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 [7], [8], however the analog components involved make such approaches often difficult to use in practice. Some methods employ training signals methods [9], [10], 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 [1] 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.

As pointed out in [11], 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 given only a modest amount of oversampling and non-blind calibration modes. Unlike some other methods, it is computationally efficient and works for any number of interlaced ADCs. Simulation results show that this method exhibits fast convergence and desirable detection accuracy.

II. PROBLEM DESCRIPTION

We model the converter input signal $x(t)$ as a 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. [12].

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

$$y_k(n) = (1+g_k^*) z(nrT + (k-1+\delta_k^*)T) \quad k = 1, 2, \ldots, r \quad n \in \mathbb{Z}, \tag{3}$$

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 $x(t)$.

This work was partially supported by NSF grant 0511461.

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 $g_1^*=0$, δ_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. [13]. 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 $\frac{rNT}{2}\le\frac{M\tilde{T}}{2}\le\frac{rNT}{2}+\tilde{T}$. 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 [14] 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

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

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 [15]. 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 $i>r$; $i=1,2,\dots,rN$ and $j=1,\dots,M+1$. We reorganize the rN samples $\{y_k(n)\}_{k=1}^{r}$ from time $\frac{-rNT}{2}$ to $\frac{rNT}{2}$, then put them into vector $\boldsymbol{y}$ of length rN. Now problem (8) becomes

$$\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c}\}=\operatorname{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c}\}}\|A\boldsymbol{c}-\boldsymbol{y}\|^2. \tag{9}$$

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

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

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

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

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

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

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
$$\{\boldsymbol{\gamma}^{(m)}\}=\operatorname{argmin}_{\boldsymbol{\gamma}}\|F(\boldsymbol{\gamma}^{(m-1)})+J(\boldsymbol{\gamma}^{(m-1)})(\boldsymbol{\gamma}-\boldsymbol{\gamma}^{(m-1)})-\boldsymbol{y}\|^2, \tag{13}$$
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 $\{\boldsymbol{\gamma}|\|F(\boldsymbol{\gamma})-\boldsymbol{y}\|\le\|F(\boldsymbol{\gamma}^{(0)})-\boldsymbol{y}\|\}$ is bounded and the Jacobian $J(\boldsymbol{\gamma})$ has full rank [16]. Moreover, if the truncation error is small and the SNR is high, this method has superlinear convergence [16].

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

$$\tilde{\boldsymbol{\delta}}^*=\frac{1}{K}\sum_{l=1}^{K}\tilde{\boldsymbol{\delta}}^{\boldsymbol{l}}\quad\text{and}\quad\tilde{\boldsymbol{g}}^*=\frac{1}{K}\sum_{l=1}^{K}\tilde{\boldsymbol{g}}^{\boldsymbol{l}}. \tag{14}$$

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 (14), 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:

$$\operatorname{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1},\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K}\sum_{k=1}^{r}\sum_{n=\frac{-N}{2}}^{\frac{N}{2}-1} \left|y_{lkn} - \tilde{\Psi}_{lk}(n)\right|^2 \qquad (15)$$

where y_{lkn} is the n-th sampling value of the k-th ADC in the l-th data block, $\tilde{\Psi}_{lk}(n) = (1+g_k)\sum_{m=\frac{-M}{2}}^{\frac{M}{2}} c_{lm}\psi_l((\delta_k+nr+k-1)T-m\tilde{T})$, 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 (15) becomes:

$$\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1},\ldots,\boldsymbol{c_K}\} = \operatorname{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1}\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K} \|A_l\boldsymbol{c_l}-\boldsymbol{y}_l\|^2. \quad (16)$$

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 (16), we use the same linearization as in (11) for different data blocks, and make $\tilde{\boldsymbol{\gamma}}^l = (\tilde{\boldsymbol{\delta}}^{l^T},\tilde{\boldsymbol{g}}^{l^T},\tilde{\boldsymbol{c}}^{l^T})^T$ 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

$$\operatorname{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g},\boldsymbol{c_1},\ldots,\boldsymbol{c_K}\}} \sum_{l=1}^{K} \|F_l(\tilde{\boldsymbol{\gamma}}^l) + J_l(\tilde{\boldsymbol{\gamma}}^l)(\boldsymbol{\gamma}_l-\tilde{\boldsymbol{\gamma}}^l)-\boldsymbol{y}_l\|^2, \quad (17)$$

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

$$J_l(\boldsymbol{\gamma}) = [\Delta_l, G_l, A_l]. \qquad (18)$$

In problem (17), if $\boldsymbol{\delta}$ and $\boldsymbol{g}$ are given, the $\boldsymbol{c_l}$ can be determined by solving the linear least squares problem $\{\boldsymbol{c_l}\} = \operatorname{argmin}_{\boldsymbol{c_l}} \|F_l(\tilde{\boldsymbol{\gamma}}^l) + J_l(\tilde{\boldsymbol{\gamma}}^l)(\boldsymbol{\gamma}_l-\tilde{\boldsymbol{\gamma}}^l)-\boldsymbol{y}_l\|^2$. Since we are only interested in the solution of $\boldsymbol{\delta}$ and $\boldsymbol{g}$ in (17), we formulate our problem as:

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

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

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

where $f_l = F_l(\tilde{\boldsymbol{\gamma}}^l) + P_{A_l}(\boldsymbol{y}_l + F_l(\tilde{\boldsymbol{\gamma}}^l)) - \boldsymbol{y}_l + H_l((\tilde{\boldsymbol{\delta}}^l)^T,(\tilde{\boldsymbol{g}}^l)^T)^T$. Thus problem (17) becomes

$$\{\boldsymbol{\delta},\boldsymbol{g}\} = \operatorname{argmin}_{\{\boldsymbol{\delta},\boldsymbol{g}\}} \sum_{l=1}^{K} \|H_l \begin{pmatrix}\boldsymbol{\delta} \\ \boldsymbol{g}\end{pmatrix} - f_l\|^2. \qquad (19)$$

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 (19) is given by $(H)^{-1}f$.

As we said before the simulation results by using (19) show better performance than by using (14). One intuitive explanation is that in (19), the essential solution depends on the matrix H, which reflects the condition of the resulting matrix H_l in all data blocks, while in (14), the final results depend on the condition of the resulting matrix individually.

By solving (19) 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 (19).
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

$$\begin{pmatrix}\boldsymbol{\delta} \\ \boldsymbol{g}\end{pmatrix} = (H)^{-1}f. \qquad (20)$$

In our simulations we found that a good choice for n_1 is $n_1 = 4$ in algorithm 3.1. And we typically set the total number of data blocks K about 20.

The details for the convergence analysis of Algorithm 3.2 is given in [17]. Fast Fourier Transform (FFT)-based multiplication and the conjugate gradient (CG) method [18] are applied to both algorithm 3.1 and algorithm 3.2. The details are introduced in [17].

IV. Simulations and Analysis

Several experiments have been done in 4-ADC and 16-ADC systems to demonstrate the efficiency of our algorithm.

Example 4.1: We consider a 4-ADC system, the input signal consists of multiple sinusoids 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. 2.

As expected, the estimation error decreases with increasing SNR.

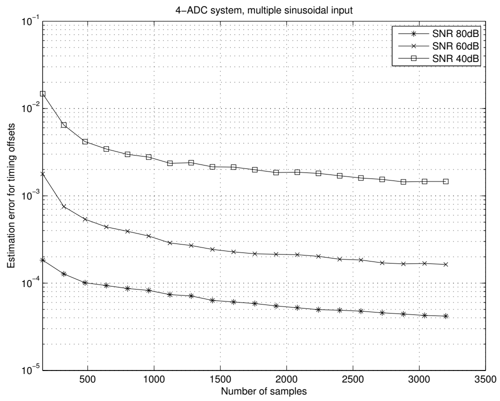

Fig. 2. Estimation errors for a 4-ADC system with blind calibration

For each SNR, our method can reach an estimation with an error which is much less than NSR. As we can see in fig. 2, the estimation errors converge very fast. In general, we can achieve a very small estimation error with only a few thousand samples. The numerical results for the detection error of gain mismatches are similar.

Since our method can achieve a very small estimation error as shown in the last example, the SNR of output signal is expected to be very close or higher than the SNR of input signal provided some oversampling.

Example 4.2: We consider a 16-ADC system. The input signal is bandlimited WGN with bandwidth $B = 1/2$, sampling rate is 1.8. 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 240 and we use a total of 40 data blocks in each experiment. We first estimate the mismatch errors, after that we apply the interpolation method introduced in [14] 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. 3.

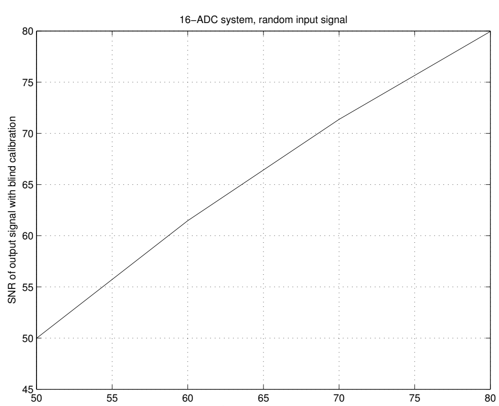

Fig. 3. SNR of output signal with blind calibration in 16-ADC systems

For the 16-ADC system, the SNR of the output signal is larger than the SNR of the input signal when the input SNR is between 50dB and 80dB with the given timing errors up to 10% of the sampling space and gain mismatches up to 0.1.

V. 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 solution. Simulation results show that our approach converges very fast and the resulting SNR of output signal is shown to be higher than the SNR of input signal in a 16-ADC system.

References

[1] 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.

[2] 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.

[3] M.C.W. McNeill, J.and Coln and B.J. Larivee, ""split adc" architecture for deterministic digital background calibration of a 16-bit 1-ms/s adc," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 2437–2445, Dec. 2005.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

[17] Jiadong Xu and Thomas Strohmer, "Efficient calibration of time-interleaved adcs via separable nonlinear least squares," *Preprint*, 2006.

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