Efficient real-time blind calibration for frequency response mismatches in two-channel TI-ADCs

Guiqing Liu, Yinan Wang\textsuperscript{a),} Xiangyu Liu, Husheng Liu, and Nan Li\textsuperscript{b)}

\textit{College of Electronic Science, National University of Defense Technology, Changsha 410073, People’s Republic of China}
\textsuperscript{a)wangyinan@nudt.edu.cn}
\textsuperscript{b)linan@nudt.edu.cn}

\textbf{Abstract:} This paper proposes an efficient full-parallel real-time blind calibration algorithm for frequency response mismatches in two-channel time-interleaved analog-to-digital converters (TI-ADCs). To make the algorithm compatible for high-speed and real-time scenarios, the algorithm is designed with full-parallel structure, where all the filters adopted in the algorithm are realized by using fast FIR algorithm (FFA). In order to reduce the computational complexity, we present a downsampling signed-FxLMS method to estimate the mismatch parameters, which can substantially save the resource consumption. Furthermore, the proposed calibration algorithm can provide low latency due to the utilization of FFA. Finally, we demonstrate the performance and efficiency of the proposed algorithm through simulations.

\textbf{Keywords:} TI-ADCs, real-time, full-parallel calibration, frequency response mismatches, resource-saving

\textbf{Classification:} Circuits and modules for electronic instrumentation

\textbf{References}


1 Introduction

Since Black and Hodges firstly proposed the concept of time-interleaved analog-to-digital converters (TI-ADCs) in 1980 [1], this high-speed sampling architecture has become a focus in the field of modern mixed signal processing. As shown in Fig. 1, the two sub-ADCs have the same sampling rate but different sampling phases, consequently, they cooperate as one single converter with a sampling rate of two
times higher than each sub-ADC. Due to the differences between the two channels, various mismatch errors are introduced, such as gain, offset, timing, bandwidth mismatches and nonlinearity mismatches [2, 3]. These mismatch errors introduce modulated spectral components and degrade the dynamic performance of the overall sampling system, which greatly limit the applicability of TI-ADCs. With the gain, offset and timing mismatch studied thoroughly [4, 5], much work has been focused on the calibration of frequency response mismatches [6, 7, 8, 9, 10]. In [8], a flexible and scalable frequency response mismatches compensation structure has been proposed. According to the compensation structure in [8], the author proposed an adaptive calibration algorithm for frequency response mismatches in two channel TI-ADCs [9]. The algorithm can effectively calibrate the frequency response mismatches, it is however not designed for real-time application scenarios. Due to the structural limitation of the algorithm, the clock frequency of the digital background calibration algorithm must be the same as the overall sampling frequency of TI-ADCs. Therefore, the digital signal processing devices, that are adopted to implement the calibration algorithms, become the bottleneck to achieve real-time calibration for the high-speed TI-ADCs.

Aiming at the real-time application scenarios, this paper proposes an efficient full-parallel and real-time blind calibration algorithm for frequency response mismatches in two-channel TI-ADCs. Firstly, we realize the calibration algorithm by utilizing the full-parallel structure, where all the filters adopted in the algorithm are designed and implemented with the fast FIR algorithm (FFA). This can effectively mitigate the requirements of clock performance and make the proposed calibration algorithm compatible for high-speed and real-time scenarios. In order to further reduce the computational complexity, a downsampling signed-FxLMS method is presented to estimate the channel mismatch parameters. With the estimated mismatch coefficients, we compensate the frequency response mismatches by using digital differentiators. Finally, we demonstrate the performance of the proposed algorithm by simulations. Compared with the calibration method in [9], the proposed algorithm can save nearly 40% resource consumption and a half latency by achieving the comparative performance as in [9].

It is mentioned that we during the preparation of this paper aware of a related work described in [11]. In [11], M-channel timing mismatches are calibrated by adaptive calibration algorithm. However, this paper calibrates the frequency response mismatch. Compared to timing mismatch, the calibration of the frequency response mismatch is more general for high-speed wide-band applications and
more complex to implement. In addition, this paper utilizes the FFA to effectively mitigate the requirements of clock performance and make the proposed calibration algorithm compatible for high-speed and real-time scenarios.

The outline of this paper is as follows. In section 2, we introduce the mismatch model and the calibration strategy. In section 3, we present the full-parallel and real-time blind calibration algorithm. Section 4 discusses the performance of the algorithm with simulations. Section 5 analyzes and compares the resource consumption. Section 6 concludes the paper.

2 Mismatches model and calibration strategy

2.1 Frequency response mismatches model

As depicted in Fig. 1, the discrete-time frequency response of the two-channels TI-ADCs is \(H_0(e^{j\omega})\) and \(H_1(e^{j\omega})\), respectively. The ideal output of a two channel TI-ADCs is denoted as \(x(n)\). The discrete-time Fourier transform (DTFT) of \(x(n)\) is \(X(e^{j\omega})\). Thus, the DTFT of the TI-ADC’s output \(y(n)\) can be given as [8, 9]

\[
Y(e^{j\omega}) = \tilde{H}_0(e^{j\omega})X(e^{j\omega}) + \tilde{H}_1(e^{j(\omega-\pi)})X(e^{j(\omega-\pi)})
\] (1)

where

\[
\tilde{H}_0(e^{j\omega}) = \frac{1}{2}(H_0(e^{j\omega}) + H_1(e^{j\omega}))
\]

\[
\tilde{H}_1(e^{j\omega}) = \frac{1}{2}(H_0(e^{j\omega}) - H_1(e^{j\omega}))
\] (2)

If the two channels have no mismatch, i.e., \(H_0(e^{j\omega}) = H_1(e^{j\omega})\). The desired output can be given as \(\tilde{X}(e^{j\omega}) = \tilde{H}_0(e^{j\omega})X(e^{j\omega})\). In order to separate the desired output from the error signal \(e(n)\), we can rewrite (1) as

\[
Y(e^{j\omega}) = \tilde{H}_0(e^{j\omega})X(e^{j\omega}) + \tilde{Q}(e^{j(\omega-\pi)})X(e^{j(\omega-\pi)})\tilde{H}_0(e^{j(\omega-\pi)})
\] (3)

where

\[
\tilde{Q}(e^{j\omega}) = \frac{\tilde{H}_1(e^{j\omega})}{\tilde{H}_0(e^{j\omega})}
\] (4)

From the above frequency-domain expressions, we can obtain the equivalent model with frequency response mismatches as shown in Fig. 2. Here, \(x(t)\) is the analog input signal. \(\tilde{x}(n)\) represents the desired TI-ADCs output without frequency response mismatches. \(y(n)\) denotes the TI-ADCs output with frequency response mismatches.

![Fig. 2. Equivalent model of a two-channel TI-ADCs with frequency response mismatches.](image-url)
2.2 Adaptive blind calibration strategy

The basic idea of the calibration is to obtain an estimation of the mismatch, which is denoted as $\tilde{e}(n)$, and then eliminate it from $y(n)$ to approximate $\tilde{x}(n)$. The structure in Fig. 3 can significantly improve the output signal $y(n)$ by using the normalized filter $\hat{Q}(e^{j\omega})$ defined in (4) [8]. According to [10], it’s reasonable to model the $\hat{Q}(e^{j\omega})$ with a $P$th-order polynomial in $j\omega$ as

$$\hat{Q}(e^{j\omega}) = \sum_{p=0}^{P} \tilde{c}_p D_p(e^{j\omega})$$  \hspace{1cm} (5)

where $\tilde{c}_p$ is the $p$th coefficient of polynomial series and

$$D_p(e^{j\omega}) = (j\omega)^p \quad -\pi < \omega < \pi$$ \hspace{1cm} (6)

is the discrete-time representation of a bandlimited $p$th order continuous-time differentiator. In general, it’s accurate enough to make $P$ equal to 2 or 3 [9, 10].

As long as we obtain the estimation $\tilde{c}_p$ of the polynomial series $\hat{c}_p$, frequency response mismatches can be compensated according to the calibration structure in Fig. 3, by replacing $\hat{Q}(e^{j\omega})$ with its estimation $\hat{Q}(e^{j\omega})$. In order to estimate $\tilde{c}_p$, we assume a special band as shown in Fig. 4, in which the input signal is absent but mismatch signal appears. Consequently, this method needs a slight oversampling of typically 10%~20%, which anyhow is commonly adopted in practical applications to accommodate filter transition bands [13]. A high-pass filter $f(n)$ is adopted to extract the mismatch signal $e(n)$ located in the input-free band from $y(n)$. Finally, we minimize the filtered error energy $\varepsilon(n)$ by finding the estimates $\tilde{c}_p$ of the coefficients $\hat{c}_p$. When $\varepsilon(n)$ converges to its minimum value, the calibrated output $\tilde{x}(n)$ approximates the desired output $\tilde{x}(n)$.

Fig. 3. Calibration strategy of frequency response mismatches.

Fig. 4. Output frequency spectrum with a input-free band dominated by mismatch error signal.
3 Low-complexity and full-parallel adaptive blind calibration algorithm

In order to effectively eliminate the mismatch error signal, the algorithm in [9] requires that the clock frequency of the digital calibration devices must be the same as the overall sampling frequency of TI-ADCs. Limited by the maximum clock frequency, the digital signal processing devices is difficult to achieve real-time calibration as the increase of TI-ADCs sampling frequency. Thus, this paper proposes an efficient full-parallel real-time blind calibration algorithm. Taking advantage of the fast FIR algorithm, the FIR filters, as marked in Fig. 5, are converted into parallel form. Furthermore, to save resource consumption and parallelize the LMS structure, we present a downsampling signed-FxLMS method which saves a third resources of a normal FFA filter.

3.1 Full-parallel calibration structure

Aiming at parallelizing the algorithm as shown in Fig. 5, we firstly parallelize all the FIR filters, including the differentiator $d_1(n)$ and the high-pass filter $f(n)$. The polyphase representation of a 2-parallel high-pass filter $f(n)$ can be given as

$$Y_0(z^2) + Y_1(z^2)z^{-1} = (X_0(z^2) + X_1(z^2)z^{-1})(H_0(z^2) + H_1(z^2)z^{-1})$$  \hspace{1cm} (7)

where $H$ is the transfer function of the high-pass filter $f(n)$. $X$ and $Y$ represent the Z-transform of the input signal $x(n)$ and the output signal $y(n)$, respectively. The subscript 0 and subscript 1 represent the Z-transform of the even coefficients and the odd coefficients, respectively. For example, $X_0$ is the Z-transform of $x(2k)$, whereas $X_1$ denotes the Z-transform of $x(2k+1)$. $Y_0$ is the Z-transform of $y(2k)$, whereas $Y_1$ denotes the Z-transform of $y(2k+1)$.

To separate the even output $y(2k)$ and the odd output $y(2k+1)$, we can rewrite Eq. (7) as [12]

$$Y_0 = H_0X_0 + Z^{-2}H_1X_1$$  
$$Y_1 = H_0X_1 + H_1X_0 = (H_0 + H_1)(X_0 + X_1) - H_0X_0 - H_1X_1$$  \hspace{1cm} (8)

According to Eq. (8), the two-parallel high-pass filter $f(n)$ can be realized according to the structure given in Fig. 6. Correspondingly, the modulated differentiator $(-1)^nd_1(n)$ is also designed with the 2-parallel structure.
According to the 2-parallel filter structure, the structure of the proposed adaptive blind calibration algorithm is depicted in Fig. 7. The sampled output \( y(n) \) of the two-channel TI-ADCs is first divided into two groups as \( y_e(n) = y(2k) \) and \( y_o(n) = y(2k + 1) \). As illustrated in Fig. 7, \( y_{0e}(n) \) and \( y_{0o}(n) \) are the output of 0th-order differentiator (i.e., the delayed output of \( y_e(n) \) and \( y_o(n) \) to consider the casual implementation of the calibration algorithm). \( y_{1e}(n) \) and \( y_{1o}(n) \) are the output of the 1st-order differentiator, whereas \( y_{2e}(n) \) and \( y_{2o}(n) \) are the output of the 2nd-order differentiator. The even outputs \( y_{0e}(n) \), \( y_{1e}(n) \), \( y_{2e}(n) \) are multiplied by the corresponding coefficients to generate the estimated error signal \( \hat{e}_e(n) \). Similarly, the odd output \( y_{0o}(n) \), \( y_{1o}(n) \), \( y_{2o}(n) \) are adopted with the estimated coefficients to generate \( \hat{e}_o(n) \). Then we use the two groups of mismatch signal \( y_e(n) \) and \( y_o(n) \) to subtract the estimated error signal \( \hat{e}_e(n) \) and \( \hat{e}_o(n) \) to obtain the estimation \( \hat{x}_e(n) \) and \( \hat{x}_o(n) \) of \( x(2k) \) and \( x(2k + 1) \), respectively.

We utilize the parallel outputs of the differentiators and the estimated signals \( \hat{x}_e(n) \) and \( \hat{x}_o(n) \) as the input of the high-pass filters \( f(n) \). The filtered outputs are used to derive the LMS algorithm to obtain the mismatch coefficients of \( \hat{c}_0(n) \), \( \hat{c}_1(n) \) and \( \hat{c}_2(n) \). Moreover, it is worth noting that one can implement the 4-parallel or 8-parallel calibration structure based on 4-parallel or 8-parallel FIR filter [12] to further reduce the requirements of the clock performance.

### 3.2 Downsampling signed-FxLMS estimation method

As indicated above, we use FxLMS algorithm to converge the mismatch parameters. The coefficient updating expression is given as

\[
\hat{c}(n) = \hat{c}(n-1) + \mu \cdot \varepsilon(n) \cdot y_d'(n) 
\]

The differentiator’s outputs are filtered by the high-pass filter \( f(n) \) to generate \( y_d'(n) \). The estimated signal \( \hat{x}_e(n) \) and \( \hat{x}_o(n) \) are filtered by the high-pass filter \( f(n) \).
to generate $e(n)$. When updating the coefficients using Eq. (9), the coefficient $\hat{c}(n)$ may not converge in the case of fixed-point operations due to the two successive multiplication [14]. Therefore, to make the coefficients convergent, we adopt the sign of $y_f^c(n)$ and $e(n)$ to realize the iteration function as

$$\hat{c}(n) = \hat{c}(n-1) + \mu \cdot \text{sign}(e(n)) \cdot \text{sign}(y_f^c(n)) \quad (10)$$

For signed-FxLMS algorithm, the iteration step of $\hat{c}(n)$ is a fixed value. In order to reduce the computational complexity, we propose a downsampling signed-FxLMS estimation structure for the mismatch coefficients, which is illustrated in Fig. 8. Here, $\downarrow M$ denotes $M$ times extraction. The high-pass filter spectrally separates the desired signal $\tilde{x}(n)$ from the error signal $e(n)$ by attenuating the $\tilde{x}(n)$ out of the input-free band, therefore, one can use the extracted error signal to drive the signed-FxLMS algorithm, which does not affect the convergence of the mismatch coefficients. The decimation ratio $M$ is determined by the adopted parallel number of the FFA. For this paper, we use 2-parallel structure, $M$ is thus equal to 2. Benefited from downsampling process, it is not necessary to calculate the odd output ($y(2k+1)$ shown in Fig. 6) of the high-pass filter $f_0(n)$, $f_1(n)$, $f_2(n)$ and $f(n)$. That is to say, the filter $H_0 + H_1$ of 2-parallel structure can be saved. The improved 2-parallel high-pass filter $f(n)$ is shown in Fig. 9, which saves a third resources of a normal FFA filter. We adopt the XOR logic operation of the sign bit of the $e(n)$ with the sign bit of the downsampled $y_f^c(n)$. Then, the XOR’s result is used to drive a lookup table (LUT) with outputs of $\mu$ or $-\mu$. The output of the LUT is connected to an accumulator to estimate the mismatch coefficients.

As shown in Fig. 8, in general, we have $f(n) = f_0(n) = f_1(n) = f_2(n)$. In order to further reduce the algorithm complexity, one can replace the $f_0(n)$ with $k$ delay units, where $k$ is the group delay of $f(n)$. Furthermore, it is feasible to replace

---

**Fig. 8.** Downsampling signed-FxLMS algorithm for mismatch coefficient estimation.

**Fig. 9.** Improved 2-parallel high-pass filter $f(n)$. 

---
the \( f_1(n) \) and \( f_2(n) \) with \( f_d(n) \) and \( s \) delay units, where \( s \) is the difference of the group delay between the filter \( f(n) \) and \( f_d(n) \). The filter \( f_d(n) \) is designed as a high-pass filter with a lower order than the \( f(n) \) that remains unchanged.

4 Simulation results

In order to verify the proposed algorithm, we construct the behavior model of the algorithm in MATLAB. We verify the performance of the calibration algorithm with various input signals. In the simulations below, we introduce different kinds of channel mismatches, including gain, timing and bandwidth mismatch. Thus, the two channel frequency response \( H_n(j\Omega) \), \( n = 0, 1 \), are modeled as

\[
H_n(j\Omega) = \frac{g_n e^{-j\Omega T d_n}}{1 + j\Omega/(1 + \Delta_n)} = \frac{g_n e^{-j\Omega T s}}{1 + j\Omega T C_{10}}(1 + \Delta_n)
\]

(11)

where \( f_s = 1/T_s \) denotes the sampling frequency, \( f_c \) and \( \Omega_c \) denote the 3-dB cutoff frequency and angular frequency, respectively. \( g_n \) and \( d_n \) are the relative gain mismatch and timing mismatch, respectively. \( \Delta_n \) represents the deviations from \( \Omega_c \).

The cutoff frequency of high-pass filter is set as \( 0.82\pi \). The order of the differentiator is 16. The orders of the high-pass filters \( f_d(n) \) and \( f(n) \) are 16 and 40, respectively.

**Example 1**: we consider a multitone input signal consisting of 41 sinusoids with constant amplitudes, uniformly spaced frequencies, and random phases. Moreover, the signal is quantized to 16 bits. The cutoff frequency of the channel frequency response is set equal to the sampling frequency, i.e., \( f_s = f_c \). As the setting in [9], the values of \( g_n \) are [0.99 1.02], the values of \( d_n \) are [0.01 -0.02] and of \( \Delta_n \) are [-0.004 0.004]. The power spectrum of the uncalibrated output is shown in Fig. 10.1a. Before calibration, the SNDR is 31.4 dB and the largest spur is \(-28.5\) dBc. By using the proposed calibration method, the SNDR is improved to 77.2 dB, and the SFDR increases to 70.2 dBc. The power spectrum of calibrated signal \( \hat{x}(n) \) is shown in Fig. 10.1b. As shown in the Fig. 11, the convergence curves precisely tend to the preset values.\(^1\)

**Example 2**: To demonstrate the versatility of the calibration method with different types of input signal, we consider a signal containing 15, 9 and 15 sinusoids, respectively. Moreover, the signal was quantized to 14 bits. The cutoff frequency is taken equal to the sampling frequency i.e., \( f_s = f_c \). In this example, the mismatch values are set as \( g_n = [1.00 \ 1.02] \), \( d_n = [0.02 \ -0.02] \), \( \Delta_n = [-0.005 \ 0.005] \). The power spectrum of the uncalibrated output \( y(n) \) is shown in Fig. 10.2a. Before calibration, the SNDR is 36.29 dB and the largest spur is \(-26.7\) dBc. By using the proposed calibration method, the SNDR is improved to 68.77 dB, and the SFDR increases to 63.6 dBc. The power spectrum of the calibrated signal \( \hat{x}(n) \) is shown in Fig. 10.2b.

\(^1\)The preset values of \( \hat{c}_0, \hat{c}_1, \hat{c}_2 \) are -1.49e-02, 1.53e-02, -2.182e-05 respectively. Since \( \hat{c}_2 \) is much smaller than \( \hat{c}_0 \) and \( \hat{c}_1 \), its influence on the frequency spectrum is weak and can be covered by the noise floor. Therefore, the estimation results for \( \hat{c}_2 \) may not be as accurate as for \( \hat{c}_0 \) and \( \hat{c}_1 \).
5 Resource consumption and performance analysis

This section discusses the resource consumption of the proposed method. First, we use even orders to implement the digital differentiator, since the performance of the differentiator with odd orders are worse than even orders [19]. We assume the order of the differentiator is \( N_d \), then the 2-parallel differentiator consumes \( (N_d/2) \times 3 + 2 \) multipliers and \( (N_d/2) \times 3 + 3 \) adders. Due to the adoption of the downsampling signed-FxLMS, the high-pass filter \( f(n) \) with order of \( N_h \) consumes \( N_h + 1 \) multipliers and \( N_h \) adders. Furthermore, the proposed algorithm consumes 6 multipliers and 9 adders except for the FIR filters. It is worth indicating that the high-pass filter used for the differentiator’s output, i.e., \( y_{0e}(n), y_{1e}(n), \) and \( y_{2e}(n) \), can be designed with a lower order (denoted as \( N_{dh} \)) than the high-pass filter for the estimation \( \hat{x}_e(n) \). The total resources consumption is given in Table I, where \( N_d \) is the order of differentiator \( d_1(n) \), \( N_h \) is the order of high-pass filter \( f(n) \), and \( N_{dh} \) is the order of high-pass filter \( f_d(n) \).

<table>
<thead>
<tr>
<th>Items</th>
<th>Multiplier</th>
<th>Adder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differentiator ( d_1(n) )</td>
<td>( (N_d/2) \times 3 + 2 )</td>
<td>( (N_d/2) \times 3 + 3 )</td>
</tr>
<tr>
<td>High-pass filter ( f(n) )</td>
<td>( N_h + 1 )</td>
<td>( N_h )</td>
</tr>
<tr>
<td>High-pass filter ( f_d(n) )</td>
<td>( N_{dh} + 1 )</td>
<td>( N_{dh} )</td>
</tr>
<tr>
<td>others</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>
According to the simulation results in [15], it is sufficient to employ the high-pass filters and the FIR differentiators with orders of 40 and 16, respectively, to reach the optimal calibration performance (SFDR of around −75 dBc). Through numerous simulations, it is found that the minimal filter of the high-pass filter \( f_d(n) \) is 16. According to Table I and the proposed calibration structure in Fig. 7, the two differentiators consume 52 multipliers and 54 adders. The two high-pass filters \( f_d(n) \) consume 34 multipliers and 32 adders. We employ one high-pass filter \( f(n) \), which consumes 41 multipliers and 40 adders. Consequently, the proposed calibration algorithm totally consumes 133 (26 * 2 + 17 * 2 + 41 + 6) multipliers and 135 (27 * 2 + 16 * 2 + 40 + 9) adders. Then, we consider the calibration method in [9]. The method totally costs 111 (9 * 2 + 21 * 4 + 9) multipliers and 198 (16 * 2 + 40 * 4 + 6) adders, where 9 multipliers and 6 adders are used except for the FIR filters.\(^2\) The high-pass filter of this paper and [9] can be replaced with \( K \) delay units, however, the convergence speed and accuracy of the coefficients will be reduced.

As we indicated above in Section 3, the operation clock frequency for the proposed calibration algorithm is only half of the correction method in [9] due to the adoption of full-parallel structure. Therefore, for a fair comparison, we consider the multiplication rate, which is denoted as the number of multiplications per output sample in the decimator [16, 17]. Let the multiplication rates of [9] is 111, then the multiplication rates of the proposed algorithm is only \( \frac{133}{2} = 66.5 \).

Furthermore, in both algorithms, the latency is mainly determined by the group delay \( L \) of the differentiator. In a 2-parallel FIR filter, the group delay of each branch is \( L/2 \). Therefore, the proposed algorithm can save half latency compared to [9]. The comparisons of the resources and the latency are shown in Table II.

<table>
<thead>
<tr>
<th>Items</th>
<th>[9]</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication Rate</td>
<td>111</td>
<td>66.5</td>
</tr>
<tr>
<td>Adder</td>
<td>198</td>
<td>135</td>
</tr>
<tr>
<td>Latency</td>
<td>( L )</td>
<td>( L/2 )</td>
</tr>
</tbody>
</table>

According to [18], the dynamic power consumption of a clock synchronized system is

\[
P = \alpha \cdot V^2 \cdot C \cdot f
\]

where \( \alpha \) is the average number of 0-to-1 transitions per clock cycle, \( V \) is the supply voltage, \( C \) is the capacitance, \( f \) is the operation clock frequency.

Assuming that \( \alpha, V \) is invariable, the adder has a capacitance of \( C_A \) and the multiplier has a capacitance of \( C_M \). The total capacitance of algorithm [9] is \( 111C_M + 198C_A \). The total capacitance of the proposed algorithm is \( 133C_M + 135C_A \). When the data throughput is the same, the operation clock frequency denoted as \( f_o \) in [9] is twice of the clock frequency in this paper. Consequently,

\(^2\)The symmetric coefficients of linear-phase FIR filter has been considered in the analysis.
the power consumption of the algorithm in [9] is $a \cdot V^2 \cdot f_\omega \cdot (111C_M + 198C_A)$, whereas, the power consumption of the proposed algorithm is $a \cdot V^2 \cdot f_\omega \cdot (66.5C_M + 67.5C_A)$, which can save 40% of the multiplication power and 65.9% of the addition power than [9].

6 Conclusion

In this paper, we have proposed an efficient full-parallel real-time blind calibration algorithm for frequency response mismatches in two-channel TI-ADCs, where the clock performance is multiplied and the power consumption is reduced in the case of the same data throughput. Secondly, in order to save resources, we present a downsampling signed-FxLMS method to estimate the mismatch parameter. Simulation results confirm the performance of the proposed algorithm. Through the complexity comparison with [9], it is noted that the proposed algorithm can save near 40% of resource consumption and a half latency under the comparative calibrated performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61701509).