Compensation numérique pour convertisseur large bande hautement parallélisé.

Nicolas Le Dortz

CentraleSupélec

N° d'ordre : 2015-02-TH

SUPELEC

ÉCOLE DOCTORALE STITS

"Sciences et Technologies de l'Information des Télécommunications et des Systèmes"

THÈSE DE DOCTORAT

DOMAINE : STIC

Spécialité : Électronique

Soutenue le 14 janvier 2015

par :

Nicolas LE DORTZ

Compensation numérique pour convertisseur large bande hautement parallélisé

Directeur de thèse : Philippe BÉNABÈS HDR - SUPELEC

Composition du jury :

Examinateur :	Philippe BÉNABÈS	HDR - SUPELEC
Rapporteur :	Dominique DALLET	Professeur - IMS Bordeaux
Examinateur :	David DECLERCQ	Professeur - ENSEA
Examinateur :	Pierre DUHAMEL	Professeur - CNRS-LSS
Rapporteur :	Andreas KAISER	Professeur - IEMN
Examinateur :	Caroline LELANDAIS-PERRAULT	Enseignant chercheur - SUPELEC
Membre invité :	Thierry SIMON	Ingénieur - STMicroelectronics
Membre invité :	Pascal URARD	Ingénieur - STMicroelectronics
Membre invité :	Filipe VINCI DOS SANTOS	Enseignant chercheur - SUPELEC
Président du jury :	Pierre DUHAMEL	Professeur - CNRS-LSS

Abstract

Because of their parallel scalable structure, time-interleaved analog-to-digital converters (TIADC) seem to be the holy grail of analog-to-digital conversion. Theoretically, their sampling speed can be increased, very simply, by duplicating the sub-converters. The real world is different because mismatches between the converters strongly reduce the TIADC performance, especially when trying to push forward the sampling speed, or the resolution of the converter. In this thesis, I show that this limitation can be alleviated by using background digital mismatch calibration.

The first part of the thesis is dedicated to studying the mismatches themselves. The various technological sources of offset, gain, skew and bandwidth mismatches are explained. Following this explanation is the derivation of a frequency model that gives an insight of the way the mismatch errors affect the TIADC output signal. Performance metrics such as the SNDR and the SFDR are expressed as the function of the mismatch levels.

In the second part of the thesis, I introduce new background digital mismatch calibration techniques that are able to reduce the offset, gain, skew and bandwidth mismatch errors. In the proposed calibration methods, the mismatches are estimated by using the statistical properties of the input signal. The calibration algorithms are shown to converge for non-stationary signals, provided that the input signal autocovariance is well-conditioned. Digital FIR filters are used to recover the correct output samples.

In the third part, a 1.6 GS/s TIADC test chip, implementing offset, gain and skew mismatch calibration, demonstrates a reduction of the mismatch spurs down to a level of -70 dBFS, up to an input frequency of 750 MHz. Temperature measurements of the chip emphasize the need for background calibration by showing the mismatch variations with respect to the temperature. The circuit achieves the lowest level of mismatches among TIADCs in the same frequency range, with a reasonable power and area, despite the overhead caused by the calibration.

Contents

List of Figures

Aknowledgements

As I am writing the last words of this dissertation, it is time for me to look back at these 3 years of work during which I crossed the path of many talented people. My PhD work was an incredible experience on a purely technical basis, but I would certainly not have achieved as much without the precious advice of colleagues, professors or simply friends.

First I want to thank Pascal Urard, my manager at ST, who gave me the opportunity to join his team in 2010, for what was then a 4-month internship. He was the first in ST to envision that blind digital mismatch calibration for TIADCs could be successful. From 2010, he was critical in convincing the most skeptical people about the potential of digital mismatch calibration. His hard work left me with free hands to focus on the research, and without his relentless obstination, the results of my PhD would certainly not have reached the same extent. More than a colleague, Pascal has been a confident and a friend with whom I shared more than professional experiences. The wonderful hiking and backcountry skiing trips that we went on together were a necessary escape from my day-to-day work.

Then, I want do deeply acknowledge Thierry Simon, my closest colleague at ST, who shares a significant part of the credit for this work. His skills in digital design made it possible to put together a functional testchip that demonstrated the benefits of digital mismatch calibration. Our numerous challenging discussions forced me to always go deeper in understanding the principles underlying mismatch calibration. I think our collaboration was ideal in many ways, particularly because we understood each other really well, and because we had complementary skills. I am also very grateful to the ADC analog design team, particularly Jean-Pierre Blanc, Sarah Verhaeren and Emmanuel Rouat for providing the analog front-end of the TIADC presented in this dissertation. A special thank you goes to Stéphane Le Tual who made me discover ADC design, and was always available to answer my endless questions about analog design. I have been particularly happy to count him as a friend an a colleague. I also want to express my gratitude to Frederic Paillardet for his circuit-debugging skills, to Andreia Cathelin for the valuable guidance, Dimitri Goguet for the lab support, and Pascale Maillet Contoz,

for the administrative support. Finally, among my colleagues, I cannot forget to express my gratitude to Fabien Todeschnini and Florian Barrau, with whom I have shared the CIFRE PhD experience at ST (and numerous coffee breaks).

My work at Supélec was quite different from the one I was doing at ST, and it was an opportunity for me to think more about the theoretical basis for my work. I want to thank Caroline Lelandais-Perrault, my PhD advisor, for her continued support. She made sure that I took the time to think about the theoretical proofs of my work, which to my opinion, is as important as having a circuit realization. Our discussions about the tiniest theoretical details helped make sure that I was not leaving any aspect behind. Then, I thank Philippe Benabes, for accepting me as one of his PhD students, and for his valuable opinions on my work. I am also grateful to him for communicating about our work to the Supélec community. I also acknowledge Stéphane Font, Filipe Vinci, and my other colleagues at Supélec for having made me discover the academic word, and for their permanent positive mood. A last thank you goes to Luc Batalie and Alexandra Siebert for their administrative support.

I have also been lucky to spend one year as a visiting researcher at UC Berkeley. I thank Borivoje Nikolic for integrating me in his research group, for supporting me throughout my stay in California, and for having me meet key people in my field. He also provided me with very valuable feedback on my presentations and papers, which greatly improved my technical communication skills. I have also deeply enjoyed the company of my friends Camilo, Miki, Dajana and Luis, with whom I often enthusiastically worked late at night in the lab.

My last words go the people who often had to stand me after a day of work. I particularly think about Pierre and Quentin, my roommates in Grenoble, and Antoine and Kilian who have accepted to host me numerous times when I was staying in Paris. I also thank my friend Alan (and his wife Fanny). Having gone through the PhD experience himself, he gave me critical advice and shared his insights about the academic world. I would also like to deeply thank my parents and my brother for their lifetime support. My final thoughts are for my dear Amanda (and Pepper) who has been wonderful in cheering me up in the past throughout this adventure.

List of abbreviations

ADC Analog-to-Digital Converter

ASIC Application-specific integrated circuit

CAD Computer Aided Design

CMOS Complementary Metal Oxyde Semiconductor

CTFT Continuous-Time Fourier Transform

DC Direct Current

DFT Discrete Fourier Transform

DNL Differential Non-Linearity

DOCSIS Data Over Cable Service Interface Specification

DTFT Discrete-Time Fourier Transform

DTPSD Discrete-Time Power Spectral Density

DVB-S Digital Video Broadcasting - Satellite

ENOB Effective number of bits

FDSOI Fully Depleted Silicon On Insulator

FFT Fast Fourier Transform

FIR Finite Impulse Response

FOM Figure Of Merit

FPGA Field Programmable Gate Array

HFB Hybrid Filter Bank

INL Integral Non-Linearity

LMS Least Mean Square

LSB Least Significant Bit

LVT Low Voltage Threshold

MOM Metal Oxide Metal (capacitor)

MSB Most Significant Bit

MUX Multiplexer

NL-HFB Non-Linear Hybrid Filter Bank

NMOS N-type Metal Oxyde Semiconductor (transistor)

OFDM Orthogonal Frequency Division Multiplexing

PAPR Peak to Average Power ratio

PMOS P-type Metal Oxyde Semiconductor (transistor)

PSD Power Spectral Density

QAM Quadrature-Amplitude Modulation

RC Resitance Capacitor

RMS Root Mean Square

S&H Sample and Hold

SAR Successive Approximation Register

SFDR Spurious-Free Dynamic Range

SHA Sample and Hold Amplifier

SNDR Signal-to-Noise and Distorsion Ratio

SNR Signal-to-Noise Ratio

T&H Track and Hold

THD Total Harmonic Distorsion

TIADC Time-Interleaved Analog-to-Digital Converter

WSCS Wide-Sense Cyclostationary

WSS Wide-Sense Stationary

Chapter 1

Introduction

1.1 The challenges of high speed analog-to-digital conversion

In modern communication systems, the information between a transmitter and a receiver is carried on continuous-time *analog* signals. This can be done either through physical media, like copper wires (e.g. TV cable, phone line) or optic fiber (e.g local networks, Internet), but the signal can also be transmitted in the vacuum via electromagnetic waves (e.g. WiFi, satellite). However, the information carried by the analog signal is usually processed by *digital* circuits. Digital circuits have certain advantages as compared to their analog counterparts:

— Complex signal processing is easier performed in the digital domain
— Digital compression algorithms and error correcting codes enable to encode signals that carry more data on a given bandwidth
— Digital designs are easier to adapt to different CMOS technologies
— Digital designs benefit from the semi-automated CAD tools for synthesis, placing and routing.
— Digital circuits are more robust to noise
— The simulation of digital circuits is faster than the simulation of analog circuits

The conversion from the analog domain to the digital domain is performed by Analog-to-Digital Converters (ADC), which are often a bottleneck in modern communication systems. The purpose of an ADC is to sample and quantize a continuous-time analog signal in order to obtain discrete-time quantized values.

For these reasons, the trend is to reduce the analog circuitry by sampling the received signal at an earlier stage at the receiver end.

The extreme case – where the market is going nowadays – is RF (Radio-Frequency)

direct sampling, which consists in sampling the continuous-time signal without converting it upstream to IF (Intermediate Frequency) with analog mixers. The block diagrams shown in Figure 1.1 illustrate the difference between IF sampling and direct RF sampling.

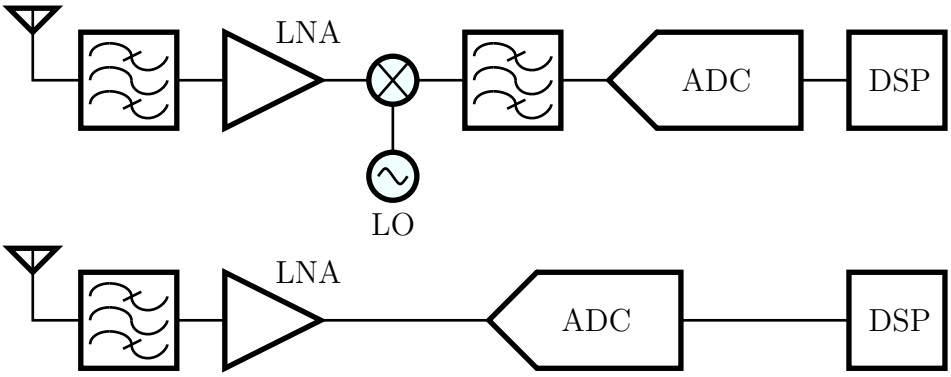


Figure 1.1 – Traditional IF sampling vs. direct RF sampling

However, moving the ADC ahead in the reception chain increases the performance requirements on the ADC. One can cite cable TV (CATV) as an example to illustrate the requirements of RF-direct sampling.

In cable TV transmissions, the signal is carried on a coaxial cable, and occupies a frequency band that goes from 54 MHz to 1002 MHz (EIA Specifications in North America). In the most common standard in use today (DOCSIS 3.0 [1]), the channels are transmitted on 6 MHz frequency wide channels distributed across the full frequency band. Those channels use QAM 256 constellations as a modulation scheme. Future improvements of this standard (DOCSIS 3.1 [2]) include increasing the size of the constellations (up to QAM 2048) as well as using OFDMA signals instead of the 6 MHz channels currently in use. Plus, certain service providers want to use the cable as a mean of transmitting other types of multimedia data (gaming, video streaming, HDTV) under the MoCA standard (Multimedia over Coax Alliance). This standard uses a band that can go from 500 MHz to 1650 MHz. To summarize, the signal that is transmitted on the cable can span a frequency band going from 54 MHz to 1650 MHz, and is encoded using complex modulation techniques such as QAM2048 and OFDMA.

Performing direct RF sampling of this kind of signal requires the ADC to be capable of sampling at a frequency F_s at least equal to twice the signal bandwidth – i.e. at least 3.2 GS/s for the cable TV application. In terms of resolution, the ADC is required to quantize the signal with enough precision to make the demodulation possible. An effective resolution of 10 bits is a minimum requirement for the cable TV application.

1.2 Analog-to-Digital converter architectures

Over the years, numerous ADC architectures have been proposed, such as flash ADCs, pipeline ADCs, Successive Approximation Register (SAR) ADCs and Time Interleaved ADCs (TIADC). They differ by the sampling frequency and the effective resolution that they achieve. Figure 1.2 shows a classification of the ADCs published between 1997 and 2014 at the International Solid State Circuit Conference (ISSCC) and the VLSI Symposium. Flash ADCs are known to be fast ADCs with a low resolution whereas SAR

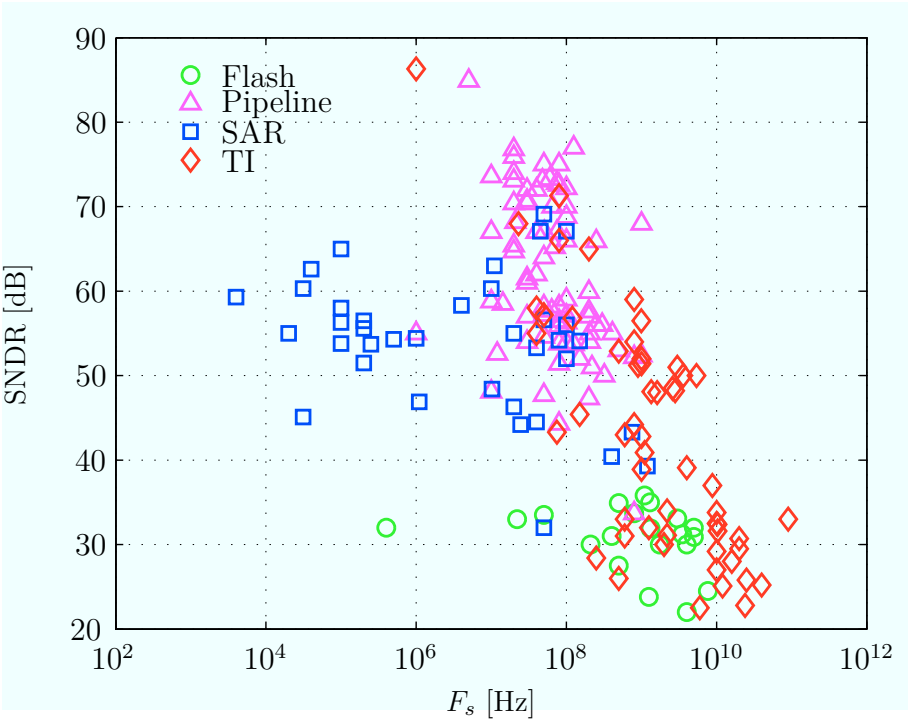


Figure 1.2 – Different ADC architectures classified with respect to their sampling frequency and their SNDR (from data published at the ISSCC and the VLSI symposium between 1997 and 2014 [3])

ADCs enable to achieve a higher SNDR at a lower sampling frequency. Pipeline ADCs have long been the dominating architecture for achieving a high sampling speed ($>$ 100 MHz) with a good SNDR but they are now surpassed by TIADCs.

Invented in 1980 [4], the TIADC is the only architecture that reaches a sampling frequency of the order of 1 GS/s with a SNDR over 50 dB (around 8 effective bits). Distinguishing TIADCs from the other ADC families is not completely accurate because TIADCs are comprised of several sub-ADCs belonging to one of the other ADC families (for example SAR ADCs). Time interleaving consists in having several "low frequency" sub-ADCs sample the signal one after an other (as illustrated in Figure 1.3), in order to increase the overall sampling frequency.

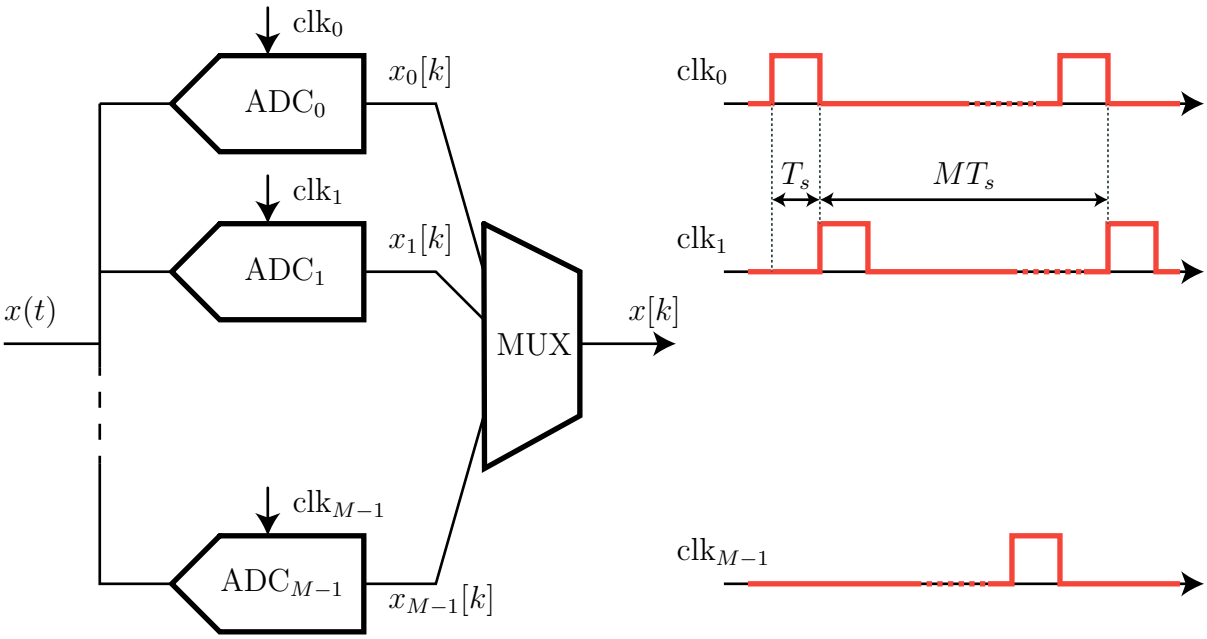


Figure 1.3 – Principle of a time-interleaved ADC

Time-interleaving also improves the energy efficiency of the analog-to-digital conversion because each of the sub-ADCs can be designed to operate in its most energy efficient frequency zone. The scatter plot in Figure 1.4 classifies the different ADC architectures based on their Figure of Merit (FOM) and their sampling frequency. The FOM is an indicator that measures the energy that an ADC uses per conversion step. It takes into account the power consumption, the effective resolution, and the sampling frequency of the ADC. It is defined as follows:

$$\text{FOM} = \frac{P}{2^{\text{ENOB}} F_s} \tag{1.1}$$

where P is the power consumption of the ADC, $\text{ENOB} = (\text{SNDR} - 1.76)/6.02$ is the effective number of bits of the ADC and F_s is the sampling frequency. The idea behind this metric is that doubling the sampling frequency of an ADC is as hard as improving its effective resolution or reducing its power consumption by a factor of 2.

Whereas time-interleaved pipeline ADCs [5, 6, 7, 8, 9] were very popular a few years ago to achieve a sampling frequency of the order of 1 GS/s, they seem to be slowly replaced by time-interleaved SAR ADCs [10, 11, 12, 13]. Time-interleaved SAR ADCs enable to achieve sampling frequencies above 10 GS/s [14, 15], and they hold the record of the highest published sampling speed with 90 GS/s (for 5.5 bits of effective resolution) [16]. The performance of pipeline ADCs tends to worsen with technology scaling because the gain of the MOSFET transistors decreases at each technology node. On the other hand, SAR ADCs do not require dynamic circuits (see Chapter 4), and gain from technology

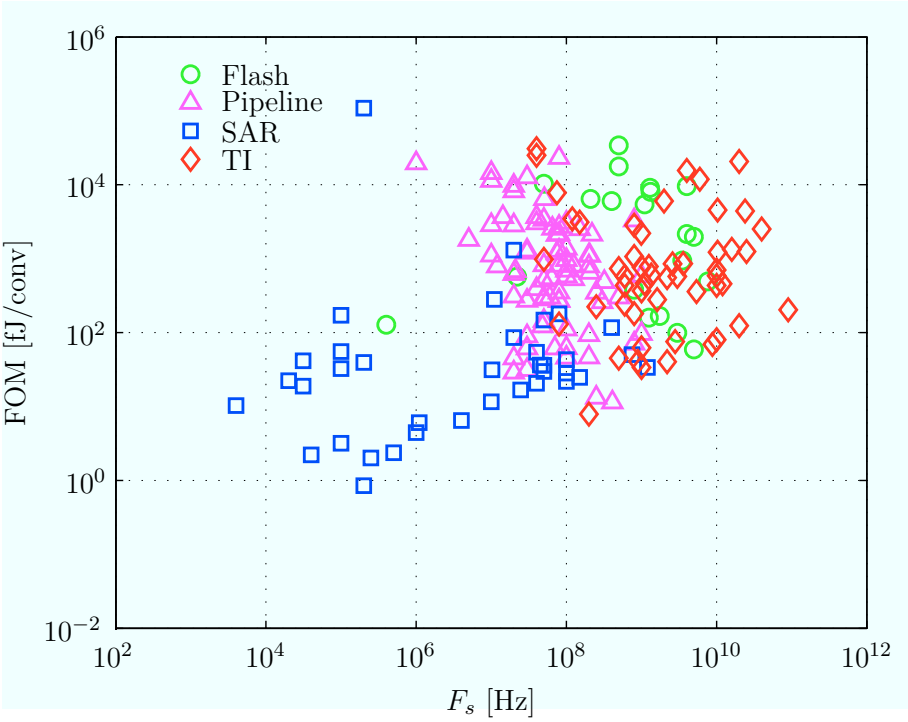


Figure 1.4 – Different ADC architectures classified with respect to their sampling frequency and their Walden FOM (from data published at the ISSCC and the VLSI symposium between 1997 and 2014 [3])

scaling thanks to their seemingly digital nature. Very recently, hybrid flash/SAR time-interleaved ADCs have been published [17]. The idea is to convert the Most Significant Bits (MSB) of the signal at a high speed using a single low resolution flash ADC (≈ 4 bits) and resolve the remaining Least Significant Bits (LSB) using time-interleaved SAR ADCs.

1.3 Mismatches in TIADCs

Theoretically, increasing the sampling frequency of a TIADC is simply done by increasing the number of sub-ADCs. However, at some point, the performance becomes limited by the mismatches between the sub-converters. Indeed, the sub-ADCs have slightly different characteristics due to random physical variations happening during the manufacturing process (see Chapter 2). These mismatches degrade the TIADC digital output signal, and the noise power at the output of the TIADC becomes higher than the one of a single sub-ADC.

The mismatches can be classified in different categories depending on their source. In real circuits, the mismatches that most commonly limit the TIADC performance are:

— the offset mismatches that occur when the sub-ADCs have different offsets

— the gain mismatches that occur when the sub-ADCs have different gains
— the timing skew mismatches that occur when the sub-ADCs sample the signal with different timing offsets
— the bandwidth mismatches that occur when the sub-ADCs have different input bandwidths

The sources and the effects of the mismatches are more detailed in Chapter 2. Figure 1.5 shows the output spectrum of a TIADC having offset, gain, skew and bandwidth mismatches. The offset mismatch noise takes the form of spurious tones at fixed frequencies,

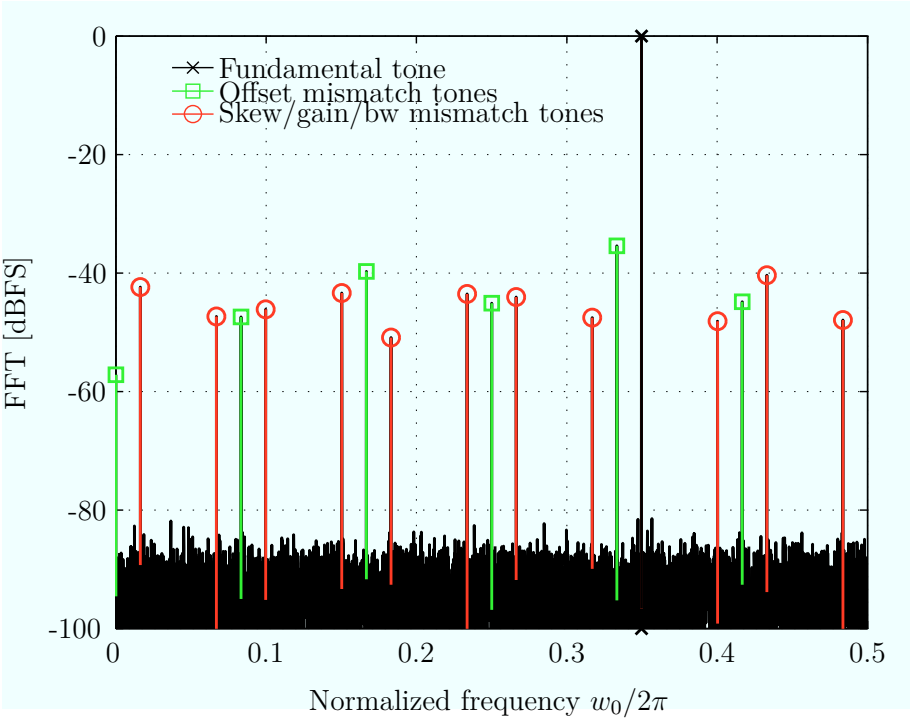


Figure 1.5 – TIADC output spectrum with mismatches

whereas the skew, gain and bandwidth mismatch noises consist of attenuated aliases of the input signal.

Since the invention of the first TIADC [4], reducing the mismatches has been a challenge. This challenge is still existing today and a lot of research is being done to develop calibration techniques to reduce the effects of the mismatches.

1.4 Mismatch calibration prior art

There are numerous types of mismatch calibration techniques, and the aim of this section is to give an overview of the different paths that have been explored in the state of the art.

1.4.1 Foreground mismatch calibration

Foreground calibration techniques [18, 19] require an offline phase where the converter is in "calibration mode". During the offline phase, a known signal, for example a sine wave with a known amplitude and a known frequency, is connected at the input of the TIADC. In the circuit presented in [16], the timing offsets are adjusted in the analog domain with a fixed-frequency sine input. In [20], the offset mismatches are calibrated in the foreground by trimming the comparators of the sub-ADCs. Foreground mismatch calibration is however not suitable for applications where the converter is always 'on', for example in communications receivers. Indeed, temperature variations and circuit aging may require the calibration to be done frequently, or even continuously. These techniques can however find their application in high-end measurement systems where the equipment can either self-calibrate or be sent to calibration. This dissertation focuses on TIADCs in the context of communication systems For that reason, foreground calibration techniques will not be further discussed.

1.4.2 Background mismatch calibration

The alternative to foreground calibration is background calibration, where the mismatch calibration is performed during the normal operation of the converter, in the background.

Some background calibration techniques require the input signal to be slightly modified in the analog domain. Those techniques are referred as *non blind* calibration techniques. For example, offset and gain mismatch calibrations can be performed with the help of a random signal that is added to the analog input signal [21] or, multiplied with it [22]. The techniques presented in [23] and [24] perform bandwidth mismatch calibration by adding a known sine wave to the TIADC input signal.

On the other hand, *blind* calibration techniques do not require any modification of the input signal, which reduces the risk of adding other sources of noise. Naturally, blind background calibration techniques are the most challenging to develop because only a limited prior knowledge about the signal is available. In reality, none of these methods is fully blind. They require some information about the input signal, usually expressed in term of spectral content or statistical properties. Among blind calibration techniques, *fully digital* calibration techniques are performed entirely in the digital domain while *mixed* mismatch calibration techniques partly operate in the digital domain and partly in the analog domain.

1.4.2.1 Mixed mismatch calibration

Most mixed mismatch calibration techniques have a structure where a digital unit controls a feedback to the analog front-end to trim some parts of the circuit. The goal of the digital unit is to detect the mismatches and to adaptively control a feebback to adjust the analog front-end.

The chip reported in [25] is an exception to this rule as it embeds an all-analog background skew mismatch calibration. The clock phases in the analog front-end can be shifted using delay lines based on a comparison with a reference clock signal.

The offset mismatches are often calibrated by equalizing the sub-ADC output averages [26, 27]. The offset mismatches can then be corrected in the analog domain, for example through digitally controlled current sources embedded in the track-and-hold amplifiers of the sub-ADC [28]. Other techniques involve auto-zeroing comparators that reduce the offset of each comparator individually, thereby reducing the offset mismatches [29, 30].

In a quite similar way, the gain mismatches can be calibrated by equalizing the output powers of the sub-ADCs [26]. Alternatively, it is possible to estimate the gain mismatches in the frequency domain by measuring the mismatch noise that appears in a frequency band free of signal [31]. In [32], the circuit uses a random chopper Sample and Hold Amplifier (SHA) in which the signal is multiplied in the analog domain by a random sequence. The offset mismatches are then estimated by equalizing the averages of the chopped sub-ADC outputs. The gain mismatches are estimated by equalizing the cross-products between the chopped output and the non-chopped output of each sub-ADC.

Skew mismatch calibration usually involves more complex techniques. The estimation of the timing mismatches can either be performed in the time domain or in the frequency domain. The techniques described in [33, 34, 35] are time based ones. The timing offsets are estimated through the minimization of a cost function calculated from cross-correlations between adjacent channels, and the sub-ADCs sampling instants are adjusted with variable delay lines. In [32], one of the two sub-ADC outputs of the 2-channel TIADC is digitally delayed to match the samples of the second channel. The technique however requires the input signal to be bandlimited to the Nyquist frequency of a single sub-ADC.

Some mixed calibration methods require the use of one or more redundant sub-ADCs. It the case of the circuit described in [20] that detects the timing mismatches in the digital domain, by minimizing the cross-covariance between each sub-ADC output and the ouput of a redunda. The timing offsets are adjusted in the analog domain by controlling adjustable delay lines. In [11, 27], the circuit has two redundant sub-ADCs, one that serves as a reference and the other one that is slightly delayed. The difference between the delayed reference ADC and the reference ADC gives a coarse approximate of the signal derivative, which is used in a correlation-based adaptive algorithm to adjust the

timing offsets through a bank of capacitors. In the TIADC circuit presented in [14], the skew mismatch calibration is performed in two phases. First, a time-to-digital signature is stored by inputting a known binary signal, generated by a DAC, into a reference ADC. Then the sub-ADC to be calibrated are disconnected from the array one after the other and replaced by the reference ADC. Their sampling times are adjusted in the analog domain by comparing their output to the reference ADC signature with the same binary signal at the input. In [17], the timing delays are adjusted through programmable delay lines by comparing the output of the sub-ADCs to the output of a flash ADC running at the TIADC overall sampling frequency.

Other approaches [36, 37, 18] propose to randomly reorganize the sampling order of the channels in order to spread out the mismatch noise across the entire spectrum. In the circuit described in [14], the offset mismatch noise is spread out by randomly connecting together unit differential pairs in the comparator pre-amplifier.

Not many mixed bandwidth mismatch calibration techniques have been proposed. One can cite the patent [38]. It describes a TIADC where the bandwidth of each sub-ADC can be adjusted through a variable boost capacitor contained in the bootstrapped switch.

1.4.2.2 Digital mismatch calibration

Digital mismatch calibration techniques have some advantages as compared to mixed calibration methods. First, they do not require any modification of the analog front-end, and they can therefore potentially be adapted to any TIADC architecture. Second, they benefit from technology scaling and from semi-automated CAD tools. However, certain functions that can be performed in the analog domain at a low cost – adjusting the delay of a sub-ADCs for example – require complex operations in the digital domain.

Correcting the offset mismatches and the gain mismatches in the digital domain is generally easy. Offset mismatch correction just requires to subtract each relative offset from each sub-ADC output [11, 39, 27]. Gain mismatch correction can be done by multiplying the output signal of each sub-ADC by the inverse of the corresponding estimated gain [35, 39]. The estimation of the offset and gain mismatches can be done similarly as in the mixed calibration case, by equalizing respectively the sub-ADC output averages, or the sub-ADC output squared average.

Solutions for fully digital calibration of the skew and bandwidth mismatches have been investigated a lot from a theoretical point of view but no integrated circuit demonstrating their efficiency has yet been published. In [40, 41], the authors propose to compensate the gain and the timing mismatches adaptively by retrieving the Wide Sense Stationarity of the signal at the TIADC output. The mismatches are digitally corrected by using a

FIR filter with variables coefficients. The work is then extended in [42, 43] to the case of arbitrary transfer function mismatches. Those methods are computationally costly as they require FIR filter with variable coefficients [44]. The skew mismatch calibration technique proposed in [45, 46] extends the work done in [47] to more than 2 interleaved ADCs. The idea is to pass the sub-ADC output signals through an Hadamard transform in order to separate the desired signal from the aliasing terms. This work is extended to the case of signals in arbitrary Nyquist zones in [48].

The skew mismatch calibration technique presented in [49, 50, 51, 52] assumes that the input signal is spectrally full, except for a small frequency band. The calibration is iteratively done by minimizing the power of the mismatch noise in this "out-of-band". The skew mismatch calibration method described in [31] uses the same principle, and the sampling errors are corrected using a first order Taylor approximation, which requires the computation of the signal derivative.

The principle of minimizing the out-of band noise was also extended to the case of generalized frequency mismatch calibration [53, 54]. In [42, 43], the frequency response of each sub-ADC is approximated by a polynom. Some papers solely deal with the problem of reconstructing a signal that is corrupted by the mismatch errors. The work done in [55, 56] uses linear interpolation to correct timing errors. An other solution, often more computationally complex, is to use fractional delay filters as explained in [57, 58]

1.5 Aim of this work and outline

This dissertation focuses on the fully digital mismatch calibration of the offset, gain, timing and bandwidth mismatches.

As reviewed above, the state-of-the-art solutions in this domain are, on many aspects, theoretical. Even, if numerical simulations prove their efficiency, most calibration techniques presented before would add an important power/area overhead to the TIADC if they were to be integrated on a circuit. This is especially true with the skew and bandwidth mismatch calibration techniques because of the complex correction filters that are required. Plus, the proposed calibration methods are always iterative and they would need an extremely long time to converge if applied on real telecommunication signals (instead of sine or multitone inputs). Finally, in most of the published work, it is assumed that the TIADC input signal is Wide Sense Stationary (WSS), which is generally not the case in practice. Real communication signals are at best Wide Sense Cyclostationary (WSCS) or non stationary. The work presented in this dissertation aims at filling these gaps.

The content of Chapter 2 is mostly a reformulation of previous work, explaining the

sources of the mismatches, and their effects on the output signal. It provides the reader with performance indicators (such as SNDR of SFDR) as a function of the mismatch levels.

Chapter 3 is the heart of this dissertation as it describes direct, i.e. non iterative, fully digital blind mismatch calibration techniques for reducing the effects of the offset mismatches, the gain mismatches, the skew mismatches and the bandwidth mismatches. The theoretical assumptions behind these techniques are carefully investigated, and it is shown that convergence is insured for non-stationary signals provided that certain conditions are met.

Chapter 4 describes the circuit realization of a TIADC embedding the offset, gain and skew mismatch calibration techniques detailed in Chapter 3. The measurements from the 1.62 GS/s TIADC test-chip are analyzed and prove the viability of fully digital mismatch calibration techniques.

Chapter 5 summarizes the findings given in this dissertation and presents some perspective to further improve calibration techniques for TIADCs.

Chapter 2

Mismatches in Time-Interleaved ADCs

2.1 Introduction

This chapter gives an overview of the sources of mismatch in TIADCs. The effects of the mismatches on the output signal are analyzed in order to provide performance metrics (SNDR and SFDR) based on the level of the mismatches. These performance indicators are useful both during the TIADC design process itself and for determining the accuracy of the mismatch calibration algorithms.

Although the mismatch problem was initially mentioned in the paper of Black and Hodges [4], the first efforts to analytically derive the effects of offset and gain mismatches on a TIADC performance were published in [59].

The analysis was later extended in [60] with a general frequency model that describes the effects of the mismatches. The model was also used to derive SNDR expressions as function of the level of the mismatches (offset, gain, skew).

The specific skew mismatch problem is also mentioned in papers dealing with non-uniform sampling (which includes timing jitter) such as [61, 19, 62, 63]. Those papers show the effects of non-uniform sampling for a sine input signal.

Later, an approach based on Hybrid Filter Banks (HFB) was proposed in order to model the mismatch effects for arbitrary input signals. This approach is extensively described in several articles [64, 65, 66] that include the effects of offset mismatches, gain mismatches, skew mismatches as well as bandwidth mismatches (first order or more). The specific case of a sine input is used to derive SNDR expressions as function of the mismatch level. HFB modeling is extended in [67, 68] to encompass non-linearity mismatches through the use of Non-Linear HFB (NL-HFB).

A probabilistic approach is proposed in [69]. The authors derive the probability

distribution of the SNDR assuming Gaussian random distributions for the offsets, gains and timing offsets.

Finally, whereas most of the literature analyzes the mismatch effects with sine input signals, the work presented in [70] analyzes skew mismatch effects for wide-band input signals. The authors show that for wide-band applications, evaluating the performances of a TIADC with sine inputs often leads to over-pessimistic requirements on the timing matching requirements.

This chapter re-demonstrates some of the results given in the literature by deriving a TIADC output spectrum when different types of mismatches are present. The reasoning that leads to these results is important because it gives a good sense of how the mismatches affect a TIADC performance. The chapter also introduces most of the notations that are used throughout this dissertation

2.2 Ideal Time-Interleaved ADC

First, the proposed modeling technique is applied to an ideal TIADC in order to introduce notations and explain the modeling principle in a simple case. With no imperfections nor mismatches, the derivation of the TIADC output spectrum unsurprisingly leads to the well-known output spectrum of an ideal ADC.

The continuous analog input signal of the TIADC is denoted $x(t)$. Depending on the applications $x(t)$ can either be a deterministic signal – for example a sine wave – or a realization of Wide Sense Stationary (WSS) random process (see in Chapter 3) – for example a complex wideband communication signal. This dissertation focuses on Nyquist converters, which means that the input signal is assumed to be bandlimited to the first Nyquist zone. If the signal is deterministic, its Continuous Time Fourier Transform (CTFT) [1], $X(\Omega)$, verifies

$$X(\Omega) = 0 \text{ for} |\Omega| > \frac{\Omega_s}{2} \tag{2.1}$$

where $\Omega_s = 2\pi F_s$ is the sampling pulsation [2].

In a TIADC, the analog input signal $x(t)$ is sampled by M sub-ADCs. Each sub-ADC is indexed by m with $m \in \mathcal{M} = \{0, \cdots, M-1\}$. The discrete sequence of samples at the output of each sub-ADC is denoted $\{x_m[k], m \in \mathcal{M}\}$. In the case of an ideal TIADC,

1. The CTFT is here defined as $X(\Omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\Omega t}dt$

2. If the signal is a WSS random process then its power spectrum density is zero outside the first Nyquist zone

$$R_{xx}(\Omega) = 0 \text{ for} |\Omega| > \frac{\Omega_s}{2}$$

each sub-ADC's output signal can be expressed

$$x_m[k] = x((kM + m)T_s) \text{ with } k \in \mathbb{N} \tag{2.2}$$

For a frequency analysis, it is easier to express the sampled signal in a continuous time form. Sampling a continuous signal at a frequency $F_s = 1/T_s$ can be done by multiplying the signal with a Dirac comb defined as follows:

$$c(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT_s) \tag{2.3}$$

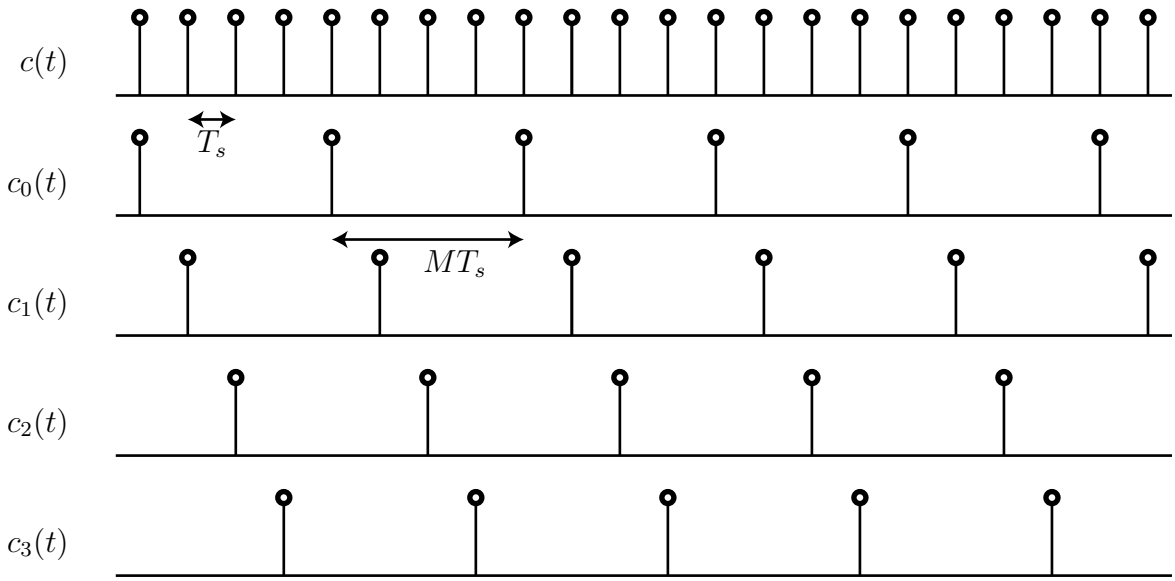


Figure 2.1 – Full rate and sub-sampled Dirac combs for $M = 4$ sub-ADCs

The sampled signal denoted $x_s(t)$ is therefore obtained by multiplying the continuous-time input signal $x(t)$ by the Dirac comb $c(t)$

$$x_s(t) = x(t)c(t) \tag{2.4}$$

In a TIADC, each sub-ADC samples the input signal at a rate F_s/M and each sub-ADC m has a delay mT_s. The sampled sub-ADC output signals, denoted $\{x_m(t), m \in \mathcal{M}\}$, are now equal the product between the input signal $x(t)$ and the respective sub-sampling Dirac combs $\{c_m(t), m \in \mathcal{M}\}$ (see illustration in Figure 2.1)

$$x_m(t) = x(t)c_m(t) \tag{2.5}$$

where

$$c_m(t) = \sum_{k=-\infty}^{+\infty} \delta(t - (kM + m)T_s) \tag{2.6}$$

The sampled TIADC output signal is obtained by adding together the sampled signals $\{x_m(t), m \in \mathcal{M}\}$ from all the sub-ADCs. When there are no mismatches, the sampled signal at the output of the TIADC is equal to the ideal sampled signal $x_s(t)$ defined in equation 2.4

$$x_s(t) = \sum_{m=0}^{M-1} x_m(t) \tag{2.7}$$

The output spectrum of an ideal TIADC is therefore the same as the output spectrum of a regular ADC. The output spectrum, denoted $X_s(\Omega)$, is obtained by taking the CTFT of the sampled signal $x_s(t)$ defined in equation 2.4. The product in the time domain becomes a convolution product in the frequency domain [3] , which yields

$$X_s(\Omega) = \frac{1}{2\pi} X(\Omega) * C(\Omega) \tag{2.8}$$

In the above equation 2.8, $C(\Omega)$ is the CTFT of the Dirac comb $c(t)$, which is also a Dirac comb in the frequency domain:

$$C(\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{+\infty} \delta(\Omega - k\Omega_s) \tag{2.9}$$

Replacing $C(\Omega)$ in the equation of the TIADC output spectrum 2.8 yields a spectrum that contains replica of the input spectrum centered at frequencies multiple of Ω_s, a well known phenomenon in sampling theory:

$$X_s(\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} X(\Omega - k\Omega_s) \tag{2.10}$$

The above expression can directly be linked to the Discrete Time Fourier Transform (DTFT) of $x[n]$. Indeed, an other way of writing the sampled signal $x_s(t)$ is

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x(t)\delta(t - nT_s) \tag{2.11}$$

$$= \sum_{n=-\infty}^{+\infty} x(nT_s)\delta(t - nT_s) \tag{2.12}$$

Taking the CTFT of the above expression and replacing $x(nT_s)$ by $x[n]$ leads to the

3. The normalization by $\frac{1}{2\pi}$ appears because of the use of angular frequencies in the CTFT.

DTFT of $x[n]$:

$$X_s(\Omega) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\Omega nT_s} \tag{2.13}$$

which is periodic of period $\Omega_s = \frac{2\pi}{T_s}$. The DTFT will be denoted $X_{2\pi}(\omega)$ where $\omega = \Omega T_s$ is the normalized angular frequency [4]

$$X_{2\pi}(\omega) = X_s(\Omega)|_{\Omega=\frac{\omega}{T_s}} = \sum_{k=-\infty}^{+\infty} x[n]e^{-j\omega n} \tag{2.14}$$

In the following sections, the mismatches are added to the model in order to characterize their effects on the TIADC output spectrum.

2.3 Mismatches in CMOS technology

Before diving into the problem of modeling TIADC mismatches, it is useful to do a quick summary of the origin of mismatches in circuits. According to Pelgrom [71]

> *Mismatch that can be observed between the parameters of a group of equally designed devices [...] is the result of several random processes which occur during every fabrication phase of the devices.*

The word "devices" can here be replaced by "transistor", "resistor", "capacitor" or "inductor" depending of the type of circuit under analysis. Mismatches can be classified according to the scale at which they happen. Thus, it is possible to distinguish

- Lot to lot mismatches, which consequence is to make devices of different lots have different physical properties.
- Wafer to wafer mismatches are mismatches that occur between devices from different wafers belonging to the same lot
- Die to die mismatches happen among devices from different dies of the same wafer
- Within die mismatches occur between devices in the same circuit

The first three types of mismatches can be excluded from our analysis because we are only interested in the mismatches between the sub-circuits (sub-ADCs) of a larger circuit (TIADC). More specifically, our interest goes toward transistors and capacitors, which are the main constituents of ADCs. In the remainder of this chapter, we will see that most of the TIADC mismatches originate from parameter variability in transistors and capacitors.

Among those parameters, the transistor threshold voltage has a particular role. The threshold voltage is indeed a parameter of the transistor that affects the properties of

4. From now on, each spectrum denoted with the subscript 2π correspond to a DTFT, and is consequently periodic in ω with period 2π.

numerous basic circuits, and from which other important parameters can be derived, such as the on-resistance of a switch. As we will see later in this chapter, understanding transistor threshold variability is useful to analyze the sources of offset mismatches, skew mismatches and bandwidth mismatches.

Threshold variability can be induced by

— Random dopant fluctuations in the channel [5] or in the source and drain [73]

— Effective channel length variability due to line edge roughness happening during the lithography process [74]

— Electron mobility variability caused by random strain variation [75]

— Surface roughness causing gate oxide thickness variations [76]

It is not the purpose of this document to analyze all the sources of variability. The simple (and old) model presented in [71] is used. It states that the variance of the threshold voltage fluctuations from one device to an other is:

— inversely proportional to the area of the devices

— proportional to the square distance between the two devices. This type of mismatch is called fixed mismatch and can be due for example to gradient effects on the wafer.

This model can be summarized by the following expression [6]:

$$\sigma^2_{V_{th}} = \frac{A^2_{V_{th}}}{WL} + S^2_{V_{th}} D^2 \tag{2.15}$$

where W is the gate width of the device, L the channel length, D the distance between two devices and $A_{V_{th}}$ and $S_{V_{th}}$ are proportionality constants that depend on the technology. We here assume that the distances between devices in the design are sufficiently small so that the fixed part of the mismatches can be neglected. With that approximation, the standard deviation of the threshold voltage variation becomes inversely proportional to the square root of the device area:

$$\sigma_{V_{th}} = \frac{A_{V_{th}}}{\sqrt{WL}} \tag{2.16}$$

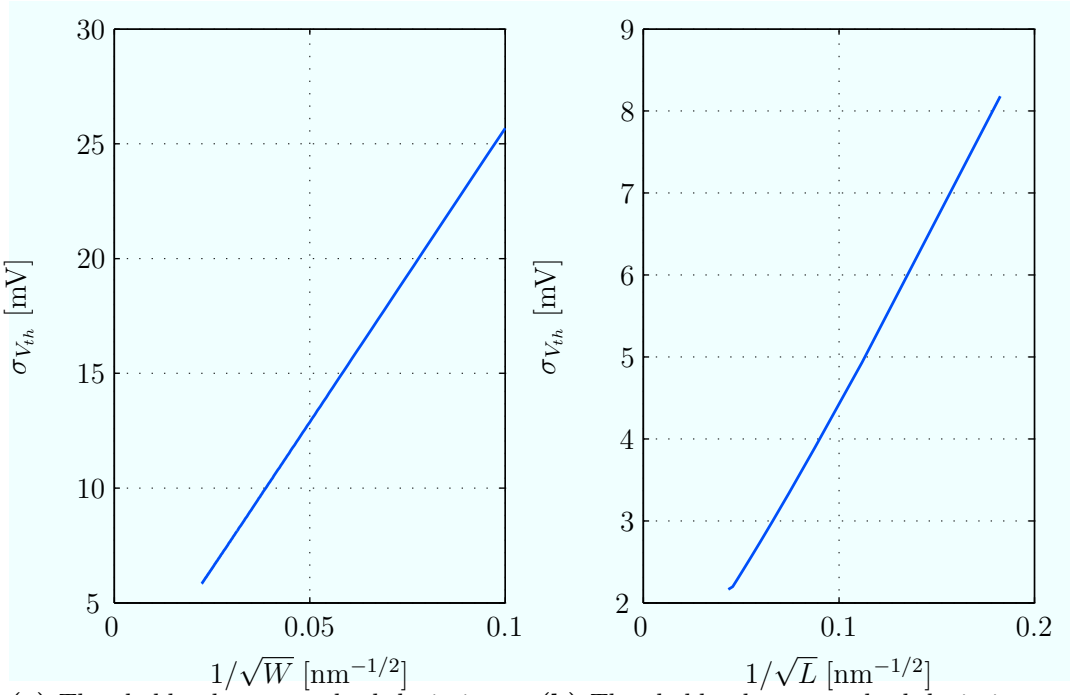
One can naturally wonder if this model, which dates from 1989, is still valid in today advanced CMOS technologies. The plots in Figure 2.2 show the standard deviation of the threshold voltage obtained from Monte-Carlo simulations of NMOS transistors in 28nm FDSOI CMOS technology, as a function of the width and the length of the transistor. It

5. In FDSOI technology, the channel is not doped, which decreases random dopant fluctuation [72]

6. The mismatches coming from the body-effect are here neglected, so that the standard deviation for V_{th} is the same as the standard deviation of V_{th0}. The formula in Pelgrom's model is actually $\sigma_{V_{th0}} = \frac{A_{V_{th0}}}{\sqrt{WL}} + S_{V_{th0}} D^2$ and $V_{th} = V_{th0} + K\left(\sqrt{|V_{sb}| + 2\phi_F} - \sqrt{2\phi_F}\right)$

turns out that, for sufficiently small devices ($L < 500$ nm and $W < 2\mu$m), the Pelgrom model seems to be valid with $A_{V_{th}} \approx 1.4$ mV $\cdot \mu$m.

The typical threshold voltage of a LVT NMOS transistor in 28nm FDSOI is around 360 mV for a minimum length device ($L = 30$ nm). It means that a minimum length transistor of 1 μm wide can expect to have RMS threshold variability $\sigma_{V_{th}} = 1.4/\sqrt{0.03 \times 1} = 8$ mV. This represents a mismatch of $8/360 = 2.2\%$.



(a) Threshold voltage standard deviation vs. width ($L = 30$nm)

(b) Threshold voltage standard deviation vs. length ($W = 1\mu$m)

Figure 2.2 – Threshold voltage standard deviation as a function of transistor dimensions. Calculated from 50 Monte-Carlo simulations of a LVT-NMOS transistor in 28nm FDSOI ($V_{gs} = 1$ V)

Of course, this is a simplified analysis that does not pretend to summarize the complex concept of transistor mismatches in advanced CMOS technology.

Capacitors are another fundamental element in ADC designs. For example, they act as the storage element during the sampling of an analog signal. Capacitors manufactured on silicon can be of different types. In the past, parallel plate capacitors made of two metal plates with oxide in between were commonly used. Nowadays, thanks to the increasing metal interconnect density and the increasing number of available metal layers, Metal-Oxide-Metal (MOM) capacitors become the norm. They are comprised of several interdigitated metal fingers that store electrical charges laterally.

Mismatches in capacitors have to main causes [77]:

— dielectric thickness variations
— edge effects

However, the capacitor mismatch behavior also obeys Pelgrom's law, thereby yielding a standard deviation of the variations inversely proportional to the area (and therefore to the capacitance) of the device [78, 79]:

$$\sigma_C = \frac{A_{C1}}{C} = \frac{A_{C2}}{\sqrt{WL}} \tag{2.17}$$

where A_{C1} and A_{C2} are constants that depend on the type of capacitor and the technology.

In the next sections, we will see how these technology variations cause mismatches in a TIADC.

2.4 Offset mismatches

The model of the TIADC presented in Section 2.2 does no take into account the mismatches between the converters. The aim of this section is to model the effects of offset mismatches.

Intuitively, the offset mismatches can be seen as a periodic sequence that is added to the signal. This sequence has a period M because the same offset is added to the output signal every M samples. Because of its periodicity, the sequence can be expressed in the frequency domain by its Fourier series, i.e. a sum of weighted tones at fixed frequencies. Since the mismatch sequence adds up to the signal, the output spectrum is therefore degraded by spurious tones. The mathematical framework developed for the ideal TIADC proves this intuition.

2.4.1 Circuit sources

Depending of the architecture of the sub-ADC, one or several comparators are used during the quantization phase. For example, a B-bit flash ADC requires 2^B comparators whereas a Successive Approximation Register (SAR) ADC only uses one. The offset mismatches happen when the comparators inside the sub-ADCs have different offsets. The offset sources in a comparator can be classified into two categories [30]:

— the static input offset, which is a static offset caused by the mismatches between the internal devices of the comparator (essentially transistors). Although the static input offset may slowly vary with temperature, it can be considered as a fixed offset.
— the input referred supply noise, which is an offset due to the comparator supply voltage variations. Input referred supply noise is by definition varying.

Comparators are usually made of several gain stages that amplify a voltage difference. Each of the stages has its own offset, which yield a global input referred offset. For instance, in high-speed ADCs, the comparator is usually made of one or two stages of pre-amplifiers followed by a latch comparator [15].

A simplest example of such an amplifier is a differential pair, which converts a voltage difference into a current difference. A NMOS differential pair is made of saturated NMOS transistors with sources connected to a current source I_0, as illustrated in Figure 2.3. If

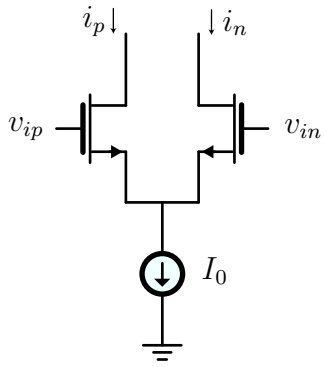


Figure 2.3 – Schematic of a NMOS differential pair

the two transistors of the pair have the same voltage on their gate, the current in each branch is $I_0/2$. On the other hand, different gate potentials yield different currents in each branch because changing the gate-source voltage changes the biasing points of the transistors. The offset of a differential pair is characterized by the fact that a current difference is observed at the output while no voltage difference is applied at the input. Or conversely, the offset is the opposite of the voltage difference that has to be applied at the input such that no current difference appears at the output. The offset can be caused by

— mismatches between the sizes of the two transistors of the pairs
— mismatches between the loads of each branch
— mismatches of threshold voltage between the two transistors
— mismatches of current factor between the two transistors

In advanced CMOS technology such as 28nm FDSOI, the V_{th} mismatches dominate. If there is a threshold voltage difference ΔV_{th} between the two transistors of the pair, the input referred offset is:

$$V_{os} = \Delta V_{th} \tag{2.18}$$

For example, a differential pair made of two minimum-length 1μm NMOS transistors would have an offset standard deviation

$$\sigma_{V_{os}} = \sigma_{V_{th}} \approx 8 \text{ mV} \tag{2.19}$$

This value corresponds to 4 LSB in a 9-bit ADC with a peak-to-peak input range of 1V.

In a single ADC, the comparator offsets yield different effects depending on the architecture that is used. In a flash ADC, they create DNL errors since different comparators with different offsets are used to compare the signal to the different reference voltages. In a SAR ADC, the same comparator is used to determine all the bits. A comparator offset therefore only creates a global offset.

When only a single ADC is used, a global offset is not a big issue because telecommunication signal do not contain any information at DC[7]. The offset mismatch problem arises when several converters, with different global offsets, are interleaved.

2.4.2 Effects on the output signal

When a TIADC has offset mismatches, a different offset is added to the output of each sub-converter, as illustrated in Figure 2.4. Assuming no quantization, the sampled

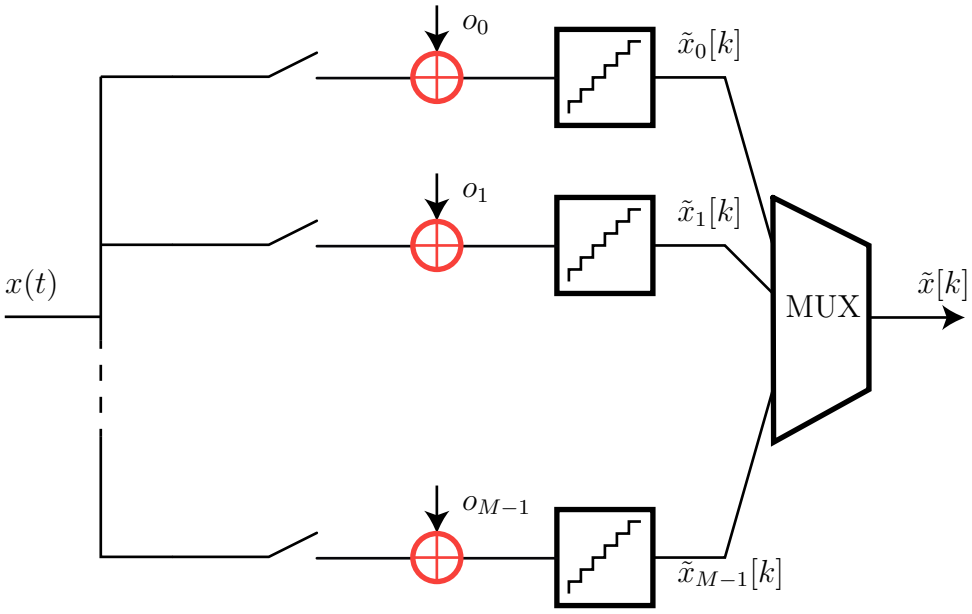


Figure 2.4 – Model of a TIADC with offset mismatches

signal at the output of each sub-ADC becomes

$$\tilde{x}_m[k] = x_m[k] + o_m \tag{2.20}$$

where o_m is the offset of the sub-ADC m.

Similar to the ideal TIADC case, the above expression can also be written in a continuous time form by adding an offset to the continuous-time signal, and by multiplying

7. However, an important offset can reduce the dynamic range of the ADC

it by the corresponding sub-sampling Dirac comb:

$$\tilde{x}_m(t) = (x(t) + o_m)c_m(t) = x_m(t) + o_m c_m(t) \tag{2.21}$$

The continuous TIADC output signal is obtained by adding the output signals of the sub-converters together:

$$\tilde{x}_s(t) = x_s(t) + \sum_{m=0}^{M-1} o_m c_m(t) \tag{2.22}$$

The above equation 2.22 shows that the TIADC output signal is a sum of the ideal output signal and weighted Dirac combs. Translating this expression into the frequency domain, through the CTFT, yields

$$\tilde{X}_s(\Omega) = X_s(\Omega) + \sum_{m=0}^{M-1} o_m C_m(\Omega) \tag{2.23}$$

The CTFT $C_m(\Omega)$ of the sub-sampling Dirac combs $c_m(t)$ are also Dirac combs:

$$C_m(\Omega) = \frac{2\pi}{MT_s} e^{-j\Omega m T_s} \sum_{k=-\infty}^{+\infty} \delta\left(\Omega - k\frac{\Omega_s}{M}\right) \tag{2.24}$$

Replacing $C_m(\Omega)$ by its expression in equation 2.23 leads to

$$\tilde{X}_s(\Omega) = X_s(\Omega) + \frac{2\pi}{MT_s} \sum_{m=0}^{M-1} o_m e^{-j\Omega m T_s} \sum_{k=-\infty}^{+\infty} \delta\left(\Omega - k\frac{\Omega_s}{M}\right) \tag{2.25}$$

The above expression can be rearranged by inverting the two sums and by noticing, because of the Dirac impulses, that the exponential terms $e^{j\Omega m T_s}$ only need to be evaluated at the frequencies $\Omega = k\frac{\Omega_s}{M}$ and can be rewritten $e^{-j2\pi\frac{km}{M}}$. The spectrum of the sampled output of the TIADC can thus be expressed

$$\tilde{X}_s(\Omega) = X_s(\Omega) + \frac{2\pi}{MT_s} \sum_{k=-\infty}^{+\infty} O_k \delta\left(\Omega - k\frac{\Omega_s}{M}\right) \tag{2.26}$$

where O_k are the Fourier coefficients associated to the sequence of offsets $\{o_0, o_1, \cdots, o_{M-1}\}$

$$O_k = \sum_{m=0}^{M-1} o_m e^{j2\pi\frac{km}{M}} \tag{2.27}$$

The DTFT of the TIADC output signal $\tilde{x}[n]$ is therefore

$$\tilde{X}_{2\pi}(\omega) = X_{2\pi}(\omega) + \frac{2\pi}{M} \sum_{k=0}^{M-1} O_k \delta_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.28}$$

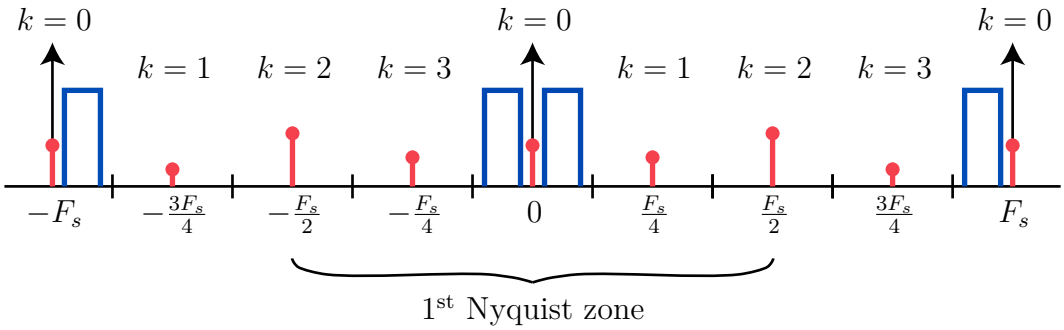


Figure 2.5 – Illustration of offset mismatch tones in TIADC output spectrum

where $\delta_{2\pi}(\omega)$ is the periodized Dirac comb DTFT, defined as follows:

$$\delta_{2\pi}(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} \delta\left(\frac{1}{T_s}(\omega - 2k\pi)\right) \tag{2.29}$$

As expected, the TIADC output spectrum is comprised of the ideal signal spectrum, and spurious tones at frequencies multiple of $2\pi/M$ (equivalent of F_s/M). This is illustrated in Figure 2.5.

2.4.3 Consequences on the TIADC performance

The output spectrum in Equation 2.28 can be used to derive important ADC performance metrics such as the SNDR and the SFDR. As those metrics are calculated with a sine input, let the input signal be $x(t) = A\cos(\Omega_0 t + \phi)$. The DTFT of $x[n] = x(nT_s)$

$$X_{2\pi}(\omega) = A\pi e^{j\omega\phi}\left(\delta_{2\pi}(\omega - \omega_0) + \delta_{2\pi}(\omega + \omega_0)\right) \tag{2.30}$$

where $\omega_0 = \Omega_0 T_s$.

2.4.3.1 SNDR degradation

Replacing the ideal signal DTFT by the expression given above in Equation 2.28 gives the TIADC output spectrum with offset mismatches:

$$\tilde{X}_{2\pi}(\omega) = \underbrace{A\pi e^{j\omega\phi}\left(\delta_{2\pi}(\omega - \omega_0) + \delta_{2\pi}(\omega + \omega_0)\right)}_{\text{Original sine tones}} + \underbrace{\frac{2\pi}{M}\sum_{k=0}^{M-1} O_k \delta_{2\pi}\left(\omega - k\frac{2\pi}{M}\right)}_{\text{Offset mismatch spurious tones}} \tag{2.31}$$

As illustrated in Figure 2.6, the output spectrum is comprised of the original sine tone, as well as offset mismatch spurious tones.

The SNDR is usually calculated for a full range sine input. It means that the sine wave

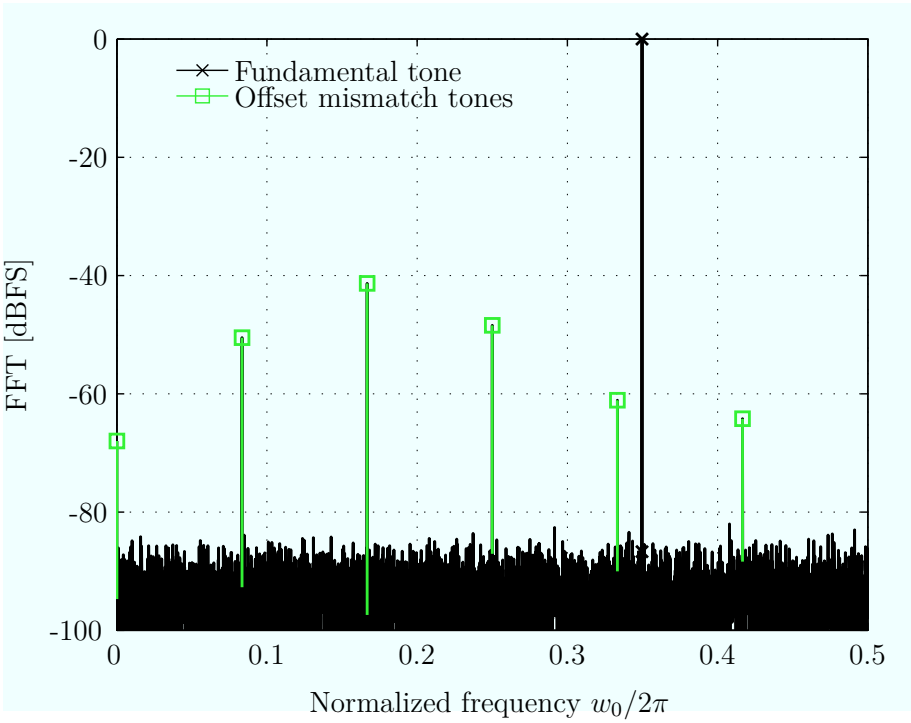


Figure 2.6 – Output spectrum of a 9-bit TIADC with offset mismatches ($M = 12$, $\omega_0 = 0.7\pi$, $\sigma_o = 2 \times \text{LSB}$, $N_{\text{FFT}} = 12 \times 4096$)

takes values between $-2^{B-1} \times \text{LSB}$ and $2^{B-1} \times \text{LSB}$, and consequently that $A = 2^{B-1}\text{LSB}$. The SNDR, in this case, is the ratio (in dB) between the signal power and the offset mismatch noise power [8]:

$$\text{SNDR} = 10 \log_{10} \left(\frac{2 \times 2^{2B-2} \times \text{LSB}^2}{\frac{4}{M^2} \sum_{k=0}^{M-1} O_k^2} \right) \tag{2.32}$$

The above expression 2.32 can be simplified using the Plancherel theorem that states that

$$\sum_{k=0}^{M-1} O_k^2 = M \sum_{m=0}^{M-1} o_m^2 = M^2 \sigma_o^2 \tag{2.33}$$

where $\sigma_o = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} o_m^2}$ is the Root Mean Square (RMS) offset mismatch level. Since the offsets are usually specified in number of LSBs, it is more convenient to express their RMS value in LSBs as well. For that, we define $\sigma_o^{\text{LSB}} = \sigma_o / \text{LSB}$ and the final SNDR expression is:

$$\text{SNDR} = 20 \log_{10} \left(\frac{2^{B-1}}{\sqrt{2} \sigma_o^{\text{LSB}}} \right) \tag{2.34}$$

Figure 2.7 illustrates the above expression for different RMS offset mismatch levels. Ba-

8. The quantization noise is not taken into account in the calculation of the noise because we are interesting in measuring the effects of the offset mismatches only

sically, multiplying the amount of offset mismatch by 2 decreases the SNDR by 6 dB, which is equivalent to loosing 1 bit of effective resolution.

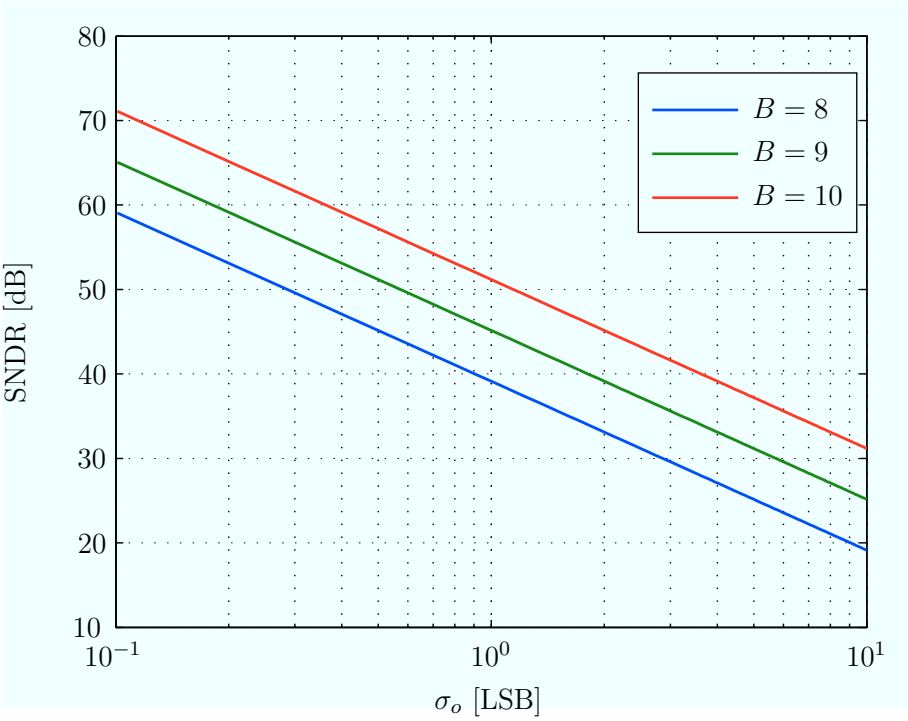


Figure 2.7 – SNDR as a function of RMS offset mismatch level

2.4.3.2 SFDR degradation

While the SNDR is an important metric to characterize an ADC performance, the SFDR is also often used to characterize offset mismatch noise. Indeed, since the offset mismatch noise takes the form of spurious tones, it is interesting to know the power of the highest tone relative to the power of the signal.

The worst case SFDR is obtained when the offset mismatch noise is concentrated in one spurious tone of index k_0. For a fixed σ_o, this translates into:

$$O_k = M\sigma_o\delta[k - k_0] \tag{2.35}$$

where $\delta[\cdot]$ is the Kronecker delta function. The values of the offsets $\{o_m, m \in \mathcal{M}\}$ are given by the inverse DFT of the $\{O_k, k \in \mathcal{M}\}$

$$o_m = \frac{1}{M}\sum_{k=0}^{M-1} O_k e^{j2\pi\frac{km}{M}} \tag{2.36}$$

When only O_{k_0} is non-zero, as expressed in equation 2.35, the above expression becomes:

$$o_m = \sigma_o e^{j2\pi \frac{k_0 m}{M}} \tag{2.37}$$

Since the offsets are real valued, k_0 can either be equal to 0 or to $\frac{M}{2}$. The case $k_0 = 0$ can be discarded because it corresponds to the spurious tone that falls at DC. The case $k_0 = \frac{M}{2}$, corresponding to a tone at the Nyquist frequency, is only possible if M is even, which yields the following offset values:

$$o_m = \sigma_o e^{j\pi m} = \sigma_o (-1)^m \tag{2.38}$$

In this case, the expression of the worst case SFDR is the ratio between the power of the positive side (or the negative side) of the fundamental, and the power of the unique offset mismatch tone

$$\text{SFDR}_{\text{worst}} = 20\log_{10}\left(\frac{2^{B-1}\text{LSB}}{2\left|O_{k_0}\right|/M}\right) = 20\log_{10}\left(\frac{2^{B-1}}{2\sigma_o^{\text{LSB}}}\right) = \text{SNDR} - 3 \text{ dB} \tag{2.39}$$

When M is even, the expression of the worst case SFDR is 3 dB lower than the SNDR (Equation 2.32) because even if all the offset mismatch noise is concentrated in one spurious tone, the calculation of the SFDR takes into account either the positive side, or the negative side of the spectrum.

When M is odd, it is not possible for the offset mismatch noise to be concentrated in only one spurious tone. In general, the spurious tones go in pairs, and the case where M is even is a particular case, where the two tones of the pairs are at half the Nyquist frequency. The fact that the offsets are real-valued implies that $O_k = O^*_{M-k}\ \forall k \in \mathcal{M}$. Consequently, the tone with index k and the tone with index $M-k$ have the same power. When M is odd, the worst case SFDR is obtained when the offset mismatch noise is only concentrated in one pair of tones, respectively indexed by k_0 and $M - k_0$ [9]. The two tones of the pair both have the same power since $|O_{k_0}| = \left|O^*_{M-k_0}\right|$, and according to the Plancherel theorem

$$|O_{k_0}|^2 = \left|O^*_{M-k_0}\right|^2 = \frac{M^2\sigma_o^2}{2} \tag{2.40}$$

The worst case SFDR is therefore expressed as

$$\text{SFDR}_{\text{worst}} = 20\log_{10}\left(\frac{2^{B-1}\text{LSB}}{2\left|O_{k_0}\right|/M}\right) = 20\log_{10}\left(\frac{2^{B-1}}{\sqrt{2}\sigma_o^{\text{LSB}}}\right) = \text{SNDR} \tag{2.41}$$

When the number of sub-ADCs is odd, the mismatch power, can at worse, be distributed

9. When M is even, the two tones can be mixed into one tone if $M - k_0 = k_0$, or alternatively $k_0 = \frac{M}{2}$

between two tones, which makes the worst case SFDR 3 dB better than when there is an even number of sub-ADCs. The offset sequences $\{o_m, m \in \mathcal{M}\}$ that lead to this worst SFDR are of the form

$$o_m = \frac{1}{M}\left(O_{k_0}e^{j2\pi\frac{mk_0}{M}} + O_{M-k_0}e^{j2\pi\frac{m(M-k_0)}{M}}\right) = \sigma_o\sqrt{2}\cos\left(2\pi\frac{mk_0}{M} + \varphi\right) \tag{2.42}$$

where $\varphi = \angle O_{k_0}$.

This worst case SFDR expression is used later in Chapter 3 to analyze the performance of the offset mismatch calibration algorithm.

2.5 Transfer function mismatches

The previous section explained how offset mismatches degrade the TIADC output spectrum. Because the offset mismatches have an additive effect in the time domain, they also have an additive effect in the frequency domain. Consequently, the offset mismatch noise is signal independent.

This is no longer the case for multiplicative mismatches, such as gain mismatches, skew mismatches or bandwidth mismatches. In this section, the same framework is used as before to model general transfer function mismatches, for which the gain, skew and bandwidth mismatches are particular cases. The choice of dealing with the general case before applying the results to the particular cases is made to avoid redundant calculations.

Let $h_m(t)$ be the impulse responses associated to each channel and $H_m(\Omega)$ the corresponding CTFT, as illustrated in Figure 2.8. The ideal case is when $h_m(t) = \delta(t)$, or equivalently $H_m(\Omega) = 1$ for all $m \in \mathcal{M}$, but in practice the transfer functions differ from one sub-ADC to an other. When there are transfer function mismatches, the continuous-time sampled sub-ADC output signals are [10]

$$\tilde{x}_m(t) = [x * h_m(t)]\, c_m(t) \tag{2.43}$$

and the continuous-time sampled output of the TIADC, obtained by adding the sub-ADC output signals together, is

$$\tilde{x}_s(t) = \sum_{m=0}^{M-1} [x(t) * h_m(t)]\, c_m(t) \tag{2.44}$$

10. The quantization noise is still neglected because we are interested in measuring the effects of the mismatches.

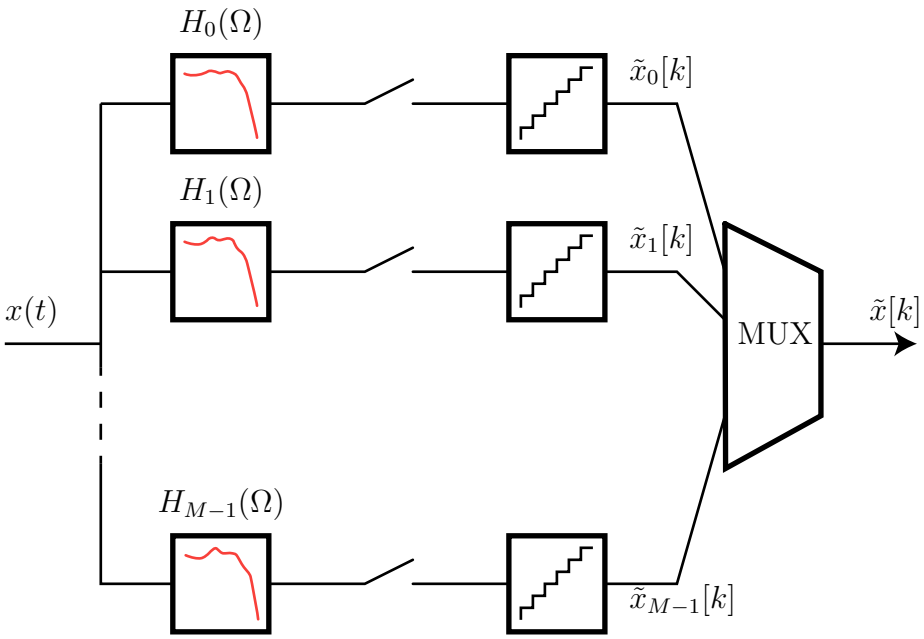


Figure 2.8 – Model of a TIADC with transfer function mismatches

In the frequency domain, the above expression translates into

$$\tilde{X}_s(\Omega) = \frac{1}{2\pi} \sum_{m=0}^{M-1} [X(\Omega) H_m(\Omega)] * C_m(\Omega) \tag{2.45}$$

The convolution product inside the sum can be simplified by inverting the sum and the integral, which leads to a sum of aliased delayed and filtered versions of the input signal:

$$\begin{aligned}[X(\Omega)H_m(\Omega)] * C_m(\Omega) =& \frac{2\pi}{MT_s} \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} X(\Omega-u) H_m(\Omega-u) \delta\left(u - k\frac{\Omega_s}{M}\right) e^{-jumT_s} du \\ =& \frac{2\pi}{MT_s} \sum_{k=-\infty}^{+\infty} X\left(\Omega - k\frac{\Omega_s}{MT_s}\right) H_m\left(\Omega - k\frac{\Omega_s}{M}\right) e^{-j2\pi\frac{km}{M}}\end{aligned} \tag{2.46}$$

Inserting the previous result back into 2.45 gives the TIADC output spectrum:

$$\tilde{X}_s(\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} \left[\frac{1}{M} \sum_{m=0}^{M-1} H_m\left(\Omega - k\frac{\Omega_s}{M}\right) e^{-j2\pi\frac{km}{M}}\right] X\left(\Omega - k\frac{\Omega_s}{M}\right) \tag{2.47}$$

which is equivalently expressed through its DTFT:

$$\tilde{X}_{2\pi}(\omega) = \sum_{k=0}^{M-1} \left[\frac{1}{M} \sum_{m=0}^{M-1} H_m\left(\frac{1}{T_s}\left(\omega - k\frac{2\pi}{M}\right)\right) e^{-j2\pi\frac{km}{M}}\right] X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.48}$$

In presence of transfer function mismatches, the TIADC output spectrum contains M aliases of the input signal spectrum around the frequencies $2\pi/M$. These aliases are

attenuated filtered versions of the input spectrum.

In the next sections, the above expression is used to derive the TIADC output spectrum when gain, skew or bandwidth mismatches are present.

2.6 Gain mismatches

This section analyzes the sources and effects of gain mismatches in TIADCs. Gain mismatches happen when the sub-ADCs have different gains. When there are gain mismatches, the sub-ADCs output samples $\{\tilde{x}_m[k], m \in \mathcal{M}\}$ are multiplied by the respective gain $\{g_m, m \in \mathcal{M}\}$ of each sub-ADC (see diagram in Figure 2.9):

$$\tilde{x}_m[k] = g_m x\left((kM + m)T_s\right) \tag{2.49}$$

Ideally, all the gains are all equal to unity.

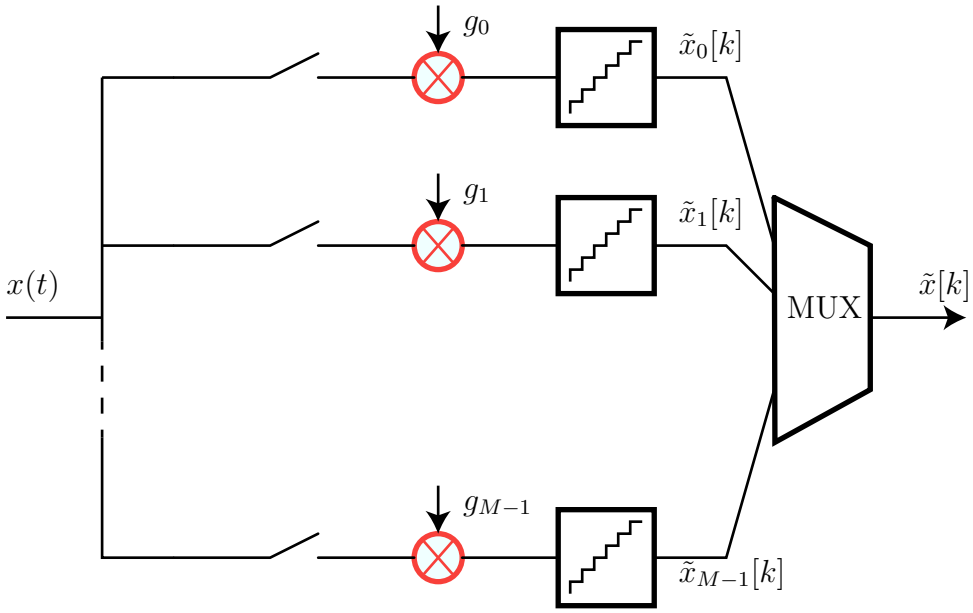


Figure 2.9 – Model of a TIADC with gain mismatches

Adding gain mismatches to the signal can be seen as multiplying the signal by a "rectangular" waveform, where each of the steps corresponds to a sub-ADC gain. This waveform has a period MT_s, and therefore its spectrum contains a fundamental at the frequency F_s/M and harmonics at frequencies integer multiples of F_s/M. It is therefore logical to expect that multiplying the input signal by this waveform will yield, in the frequency domain, replica of the input signal spectrum around frequencies multiple of F_s/M.

This intuition is formally proved in Section 2.6.2 by using the expression of the spectrum obtained in the general case of transfer function mismatches (see Section 2.5).

2.6.1 Circuit sources

The source of gain mismatches between the sub-ADCs highly depends on the sub-ADC architecture that is used. As for the case of offset mismatch, the list of gain mismatches sources presented here is not exhaustive. In this section, we take the example of a SAR ADC because it is the architecture that is used in the circuit presented in (Chapter 4) of this document.

In a SAR ADC, the input signal is sampled and held on a sampling capacitor C_s before being quantized. In this type of ADC architecture, gain mismatches can originate from the quantization process itself. As detailed later in 4, the sampling capacitor is usually comprised of binary weighted capacitors. During the quantization process the (bottom or top) plate of each of the capacitors can be connected to different reference voltages. Gain mismatches can appear if the reference voltages are different from sub-ADC to sub-ADC. In practice however, those reference voltages are usually derived from resistively loaded current sources that are common to all the sub-ADCs, which makes the variations rather small.

Another (potentially more important) source of gain mismatches is the parasitic capacitance seen by the input signal in parallel to the sampling capacitor. Due to charge sharing, the effective signal stored on the sampling capacitor is attenuated by the parasitic capacitance. If bottom place sampling (see 4) is used, the signal and the reference voltages see the same parasitic capacitance and are therefore attenuated similarly. This does not create an apparent gain during the quantization. If top plate sampling is used, the signal is connected to the top plate of the sampling capacitor whereas the reference voltages are connected to the bottom plates. It means that the reference voltages and the input signal see different parasitic capacitances, which can lead to an apparent gain during the quantization process. Consequently, gain mismatches can happen when the parasitic capacitances vary from one sub-ADC to an other.

2.6.2 Effects on the output signal

When a TIADC has gain mismatches, the gain of the sub-ADCs are different. The transfer function of each sub-ADC is:

$$h_m(t) = g_m \delta(t) \tag{2.50}$$

for which the CTFTs are expressed as

$$H_m(\Omega) = g_m \tag{2.51}$$

The TIADC output spectrum in presence of gain mismatches can be derived from the TIADC output spectrum in presence of arbitrary transfer function mismatches. This is done by replacing $H_m(\Omega)$ in Equation 2.48 by its expression given in Equation 2.51

$$\tilde{X}_{2\pi}(\omega) = \sum_{k=0}^{M-1} \left[\frac{1}{M} \sum_{m=0}^{M-1} g_m e^{-j2\pi \frac{km}{M}} \right] X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.52}$$

The obtained output spectrum is, not surprisingly, comprised of aliased versions of the input spectrum (see Figure 2.10. This is even better highlighted if we denote

$$G_k = \sum_{m=0}^{M-1} g_m e^{-2j\pi \frac{km}{M}} \tag{2.53}$$

and rewrite the output spectrum

$$\tilde{X}_{2\pi}(\omega) = \sum_{k=1}^{M-1} \frac{G_k}{M} X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.54}$$

It is interesting to notice that the aliasing terms are weighted by the DFT coefficients

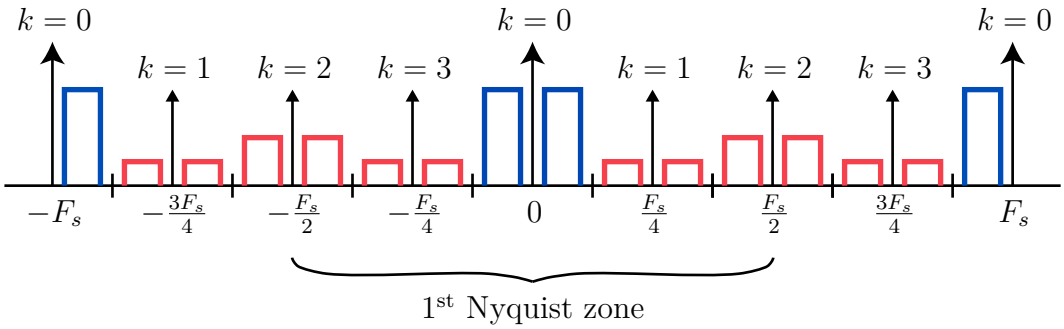


Figure 2.10 – Illustration of TIADC output spectrum with gain mismatch aliases

$\{G_k, k \in \mathcal{M}\}$ associated to the gain sequence $\{g_m, m \in \mathcal{M}\}$. In other words, the attenuation of each of the aliases depends on the frequency signature of the gain mismatch sequence. For example, if the sub-ADCs all have the same gain g (meaning that there are no gain mismatches), only the DC frequency component of the gain sequence has a non-zero value $G_k = Mg\delta[k]$. In this case, the output spectrum is just an attenuated version of the input spectrum and there are no aliases:

$$\tilde{X}_{2\pi}(\omega) = gX_{2\pi}(\omega) \tag{2.55}$$

This is expected, as a global gain just attenuates the signal without creating any other perturbation.

2.6.3 Consequences on the TIADC performance

Contrary to the offset mismatch case, the noise due to the gain mismatches depends on the frequency characteristics of the input signal. With a sine input, the output spectrum takes the following form

$$\tilde{X}_{2\pi}(\omega) \approx A\pi e^{j\omega\phi}\left[\frac{G_0}{M}\left(\delta_{2\pi}(\omega-\omega_0)+\delta_{2\pi}(\omega+\omega_0)\right)\right.$$
$$\left.+\sum_{k=1}^{M-1}\frac{G_k}{M}\left(\delta_{2\pi}\left(\omega-k\frac{2\pi}{M}+\omega_0\right)-\delta_{2\pi}\left(\omega-k\frac{2\pi}{M}-\omega_0\right)\right)\right] \tag{2.56}$$

It is comprised of the original sine spectrum and of attenuated aliased versions of it, as illustrated in Figure 2.11. The ratio between the signal power (first term in Equation 2.56)

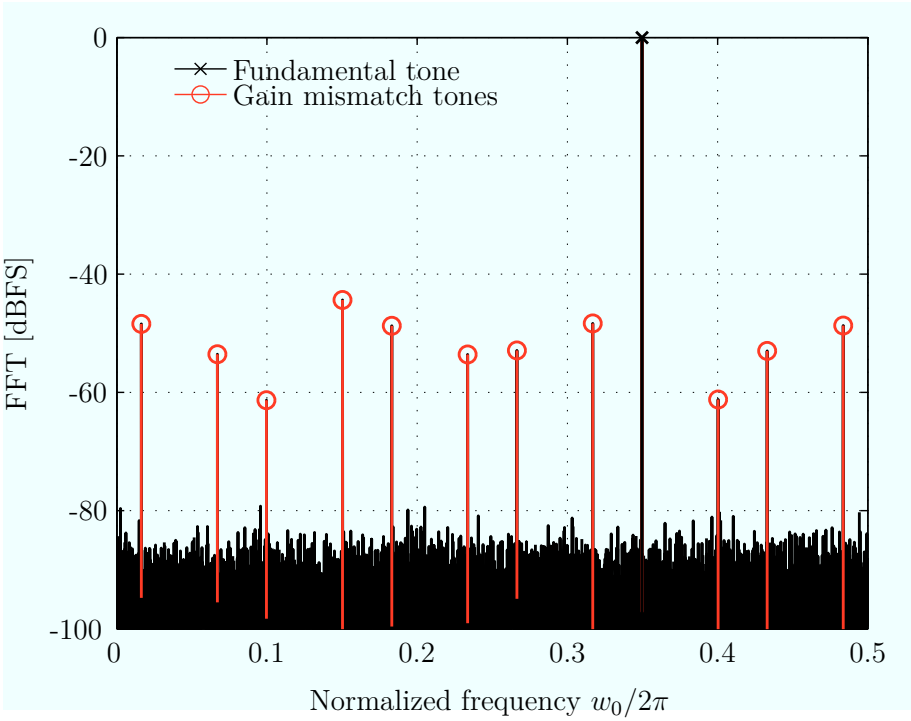


Figure 2.11 – Output spectrum of a 9-bit TIADC with of gain mismatches ($M = 12$, $\omega_0 = 0.7\pi$, $\sigma_g = 0.01$, $N_{\text{FFT}} = 12 \times 4096$)

and the noise power (sum term in Equation 2.56) is

$$\frac{\text{Power of the signal}}{\text{Power of the noise}} = \frac{\frac{|G_0|^2}{M^2}}{\frac{1}{M^2}\sum_{k=1}^{M-1}|G_k|^2} \tag{2.57}$$

In Equation 2.57, the numerator and the denominator are function of the gains of the sub-ADCs. The numerator corresponds to the squared average of the gains of the sub-ADCs because

$$\frac{G_0}{M} = \bar{g} = \frac{1}{M}\sum_{m=0}^{M-1} g_m \tag{2.58}$$

whereas the denominator can be rewritten using the Plancherel theorem. It states that

$$\sum_{k=0}^{M-1} |G_k|^2 = M \sum_{m=0}^{M-1} g_m^2 = \tag{2.59}$$

It follows that

$$\sum_{k=1}^{M-1} |G_k|^2 = M \sum_{m=0}^{M-1} g_m^2 - M\bar{g}^2 = M\sigma_g^2 \tag{2.60}$$

where $\sigma_g = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1}(g_m - \bar{g})^2}$ is the RMS gain mismatch level. Putting back the results of equation 2.58 and equation 2.60 into the SNDR expression in equation 2.57 yields:

$$\text{SNDR} = 10\log_{10}\left(\frac{\bar{g}^2}{\sigma_g^2}\right) = -20\log_{10}(\bar{\sigma}_g) \tag{2.61}$$

where $\bar{\sigma}_g = \sigma_g/\bar{g}$ is the normalized RMS gain mismatch level. The SNDR as a function of the RMS gain mismatch level is plotted in Figure 2.12

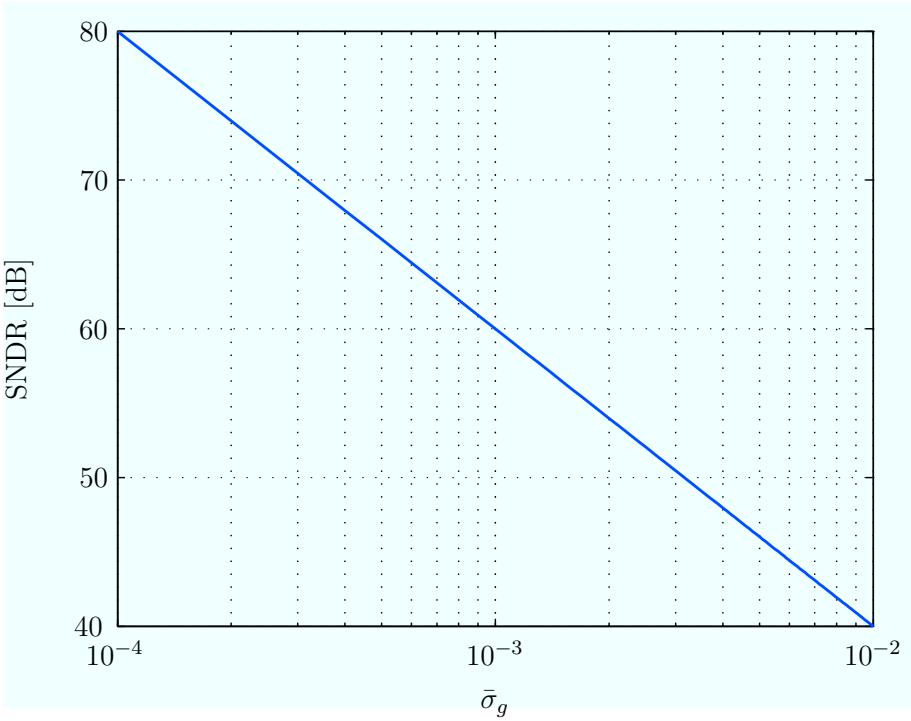


Figure 2.12 – SNDR as a function of RMS gain mismatch level

Although theoretically possible, analyzing the SFDR degradation caused by the gain

mismatches is not as useful as in the offset mismatch case. Indeed, telecommunication signals often have a wide frequency band, which has for consequence to spread the mismatch noise across the entire Nyquist band. It does not mean that the level of the most powerful gain mismatch tone is irrelevant. We will see in Chapter 4 that the SFDR is an important indicator when evaluating the mismatch reduction achieved by each of the calibration technique.

2.7 Skew mismatches

In a TIADC, skew mismatches appear when the sampling instants of the sub-ADCs are not perfectly adjusted. In theory, each sub-ADC samples exactly T_s seconds after the previous sub-ADC, which makes the overall sampling scheme uniform in time. In practice, technology variations or layout imperfections can reduce or increase the sampling interval between consecutive sub-ADCs. Normally, sub-ADC m samples the signal at

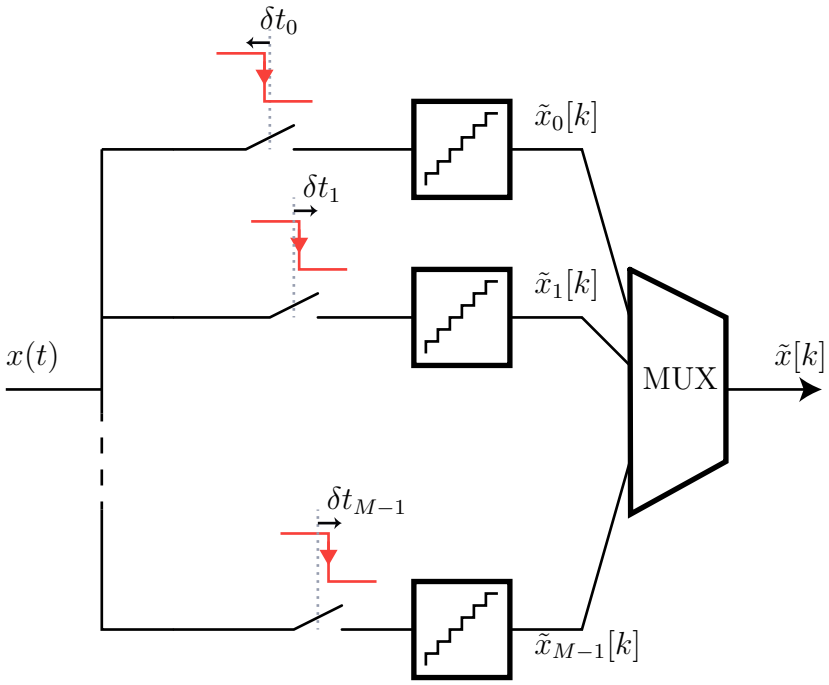


Figure 2.13 – Model of a TIADC with skew mismatches

times $(kM + m)T_s$ but because of the mismatches, a positive or negative delay δt_m, different for each sub-ADC, changes the sub-ADC sampling time to $(kM + m)T_s + \delta t_m$. This is illustrated by the block diagram in Figure 2.13.

Similar to gain mismatches, skew mismatches yield aliases of the input signal spectrum around frequencies multiple of F_s/M. This is demonstrated later in this section by using the general transfer function output spectrum derived in Section 2.5.

2.7.1 Circuit sources

In a TIADC, each of the sub-ADC receives its own sampling clock, generated by a phase generator. The sampling clock is used to close a switch (MOS transistor) and the closing instant determines the sampling time. The skew mismatches can either come from the clock signal path or from the switch itself.

Each clock signal is transmitted to each sub-ADC on metal wires and is usually passed through several inverters. The timing mismatches can originate from load differences between inverters from different clock paths. If two inverters are chained, the load of the second inverter affects the speed of the first inverter, thereby delaying the clock signal more or less. Mismatches can also come from metal trace thickness variations that change the capacitance and the resistance of the wire driving the clock signal.

The threshold voltage variations is however the most detrimental effect because it affects the inverters in the clock path and the sampling switch itself. Threshold voltage variations move the trimming point of an inverter, thereby increasing or reducing its propagation delay. If the sampling switch is a NMOS transistor, a reduced threshold voltage, delays the sampling instant, whereas an increased threshold voltage advances the sampling instant. For example, if the sampling clock slew rate is assumed to be 0.02 V/ps [11], a RMS threshold voltage variation $\sigma_{V_{th}} = 8$ mV (see example in Section 2.3) yields a RMS timing skew mismatch level of 0.4 ps.

2.7.2 Effects on the output signal

The frequency domain effects of the skew mismatches can be derived in a way similar to those of the gain mismatches (section 2.6). The skew mismatches occur when each sub-converter samples the input signal with a different delay $\{\Delta t_m, m \in \mathcal{M}\}$. The transfer function of each channel can be written

$$h_m(t) = \delta(t - \Delta t_m) \tag{2.62}$$

In the frequency domain, the transfer function becomes

$$H_m(\Omega) = e^{-j\Omega\Delta t_m} \tag{2.63}$$

11. it means that it takes 50 ps for the clock to go from 1 V down to 0 V.

The output spectrum with the skew mismatch effects is obtained by inserting the above expression in the general case expression of Equation 2.48:

$$\tilde{X}_{2\pi}(\omega) = \sum_{k=0}^{M-1} \left[\frac{1}{M} \sum_{m=0}^{M-1} e^{-j\left(\omega - k\frac{2\pi}{M}\right) r_m} e^{-j2\pi \frac{km}{M}} \right] X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.64}$$

where $\left\{ r_m = \frac{\Delta t_m}{T_s}, m \in \mathcal{M} \right\}$ are the timing delays relative to the sampling period. In practice, the timing mismatches are small as compared to the sampling period and we can assume that $r_m \ll 1$. The small timing skew assumption enables to further simplify expression 2.64 by noticing that

$$e^{-j\left(\omega - k\frac{2\pi}{M}\right) r_m} \approx 1 - j r_m \left(\omega - k\frac{2\pi}{M}\right) \tag{2.65}$$

Inserting the above first order approximation into expression 2.64 leads to the TIADC output spectrum expression:

$$\tilde{X}_{2\pi}(\omega) \approx (1 - j\omega\bar{r}) X\left(e^{j\omega}\right) - \sum_{k=1}^{M-1} \left[\frac{1}{M} \sum_{m=0}^{M-1} r_m e^{-j2\pi \frac{km}{M}} \right] j\left(\omega - k\frac{2\pi}{M}\right) X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.66}$$

where $\bar{r} = \frac{1}{M}\sum_{m=0}^{M-1} r_m$ is the average of the relative timing delays. An analysis of the

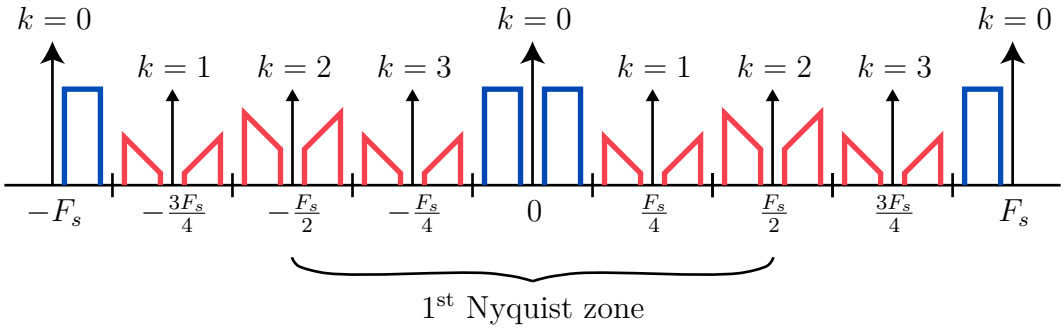


Figure 2.14 – Illustration of TIADC output spectrum with skew mismatch aliases

above expression show that, in presence of the timing skew mismatches, the TIADC output spectrum is comprised of

— the ideal signal term with a global delay $\bar{r}$, expressing the fact that the signal is globally delayed by the average of the sub-ADC timing delays. Since a global delay on the signal does not affect performance, $\bar{r}$ is assumed to be zero without loss of generality.

— aliasing terms of the derivative of the signal that are shifted in frequency (around frequency multiple of F_s/M) and weighted by the DFT of the timing delays, de-

noted $R_k = \sum_{m=0}^{M-1} r_m e^{-j2\pi \frac{km}{M}}$ with $k \in \{1, \cdots, M-1\}$ [12].

2.7.3 Consequences on the TIADC performance

For a sine input, the output spectrum (illustrated in Figure 2.15) can be written

$$\tilde{X}_{2\pi}(\omega) \approx A\pi e^{j\omega\phi} \Bigg[\delta_{2\pi}(\omega - \omega_0) + \delta_{2\pi}(\omega + \omega_0) + \sum_{k=1}^{M-1} \frac{R_k}{M} j\omega_0 \left(\delta_{2\pi}\left(\omega - k\frac{2\pi}{M} + \omega_0\right) - \delta_{2\pi}\left(\omega - k\frac{2\pi}{M} - \omega_0\right)\right)\Bigg] \quad (2.67)$$

The SNDR is the ratio (in dB) between the power of the signal and the power of the

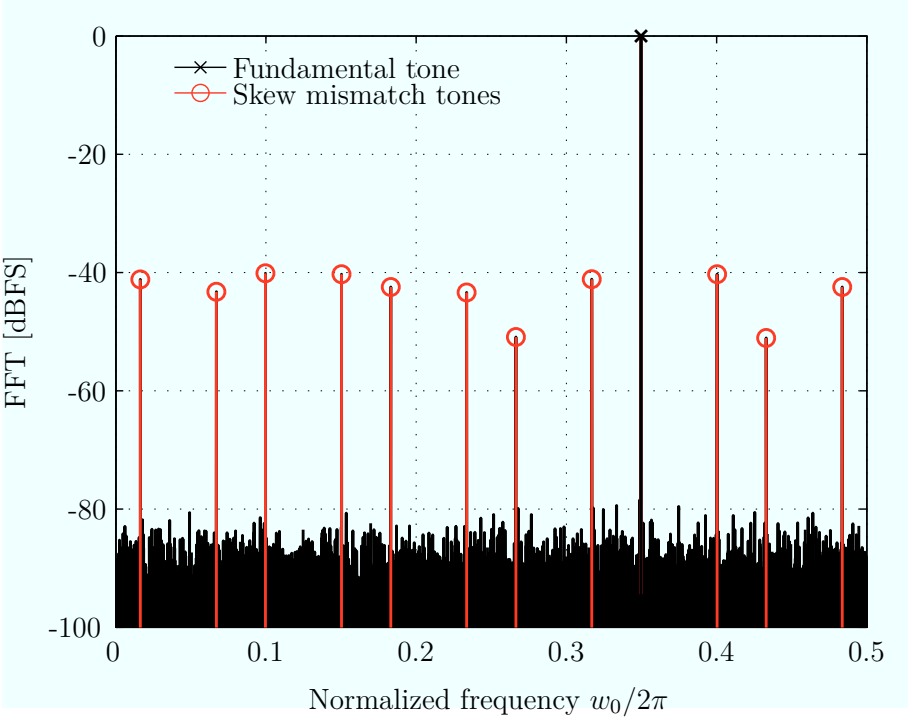


Figure 2.15 – Output spectrum of a 9-bits TIADC with skew mismatches ($M = 12$, $\omega_0 = 0.7\pi$, $\sigma_r = 0.01$, $N_{\text{FFT}} = 12 \times 4096$)

$M - 1$ aliases. Its expression is

$$\text{SNDR} = 10 \log_{10} \left(\frac{1}{\frac{w_0^2}{M^2} \sum_{k=0}^{M-1} R_k^2} \right) \quad (2.68)$$

The Plancherel theorem, which states that $\frac{1}{M} \sum_{k=0}^{M-1} R_k^2 = \sum_{m=0}^{M-1} r_m^2$, can be applied to Equation 2.68 to obtain the value of the SNDR as a function of the sub-ADC timing

12. $R_0 = M\bar{r} = 0$

offsets and the input frequency

$$\text{SNDR} = 20\log_{10}\left(\frac{1}{\omega_0\sigma_r}\right) \tag{2.69}$$

In the above equation, $\sigma_r = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1} r_m^2}$ is the RMS value of the timing delays. The SNDR is plotted in Figure 2.16 for different mismatch levels. The SNDR naturally

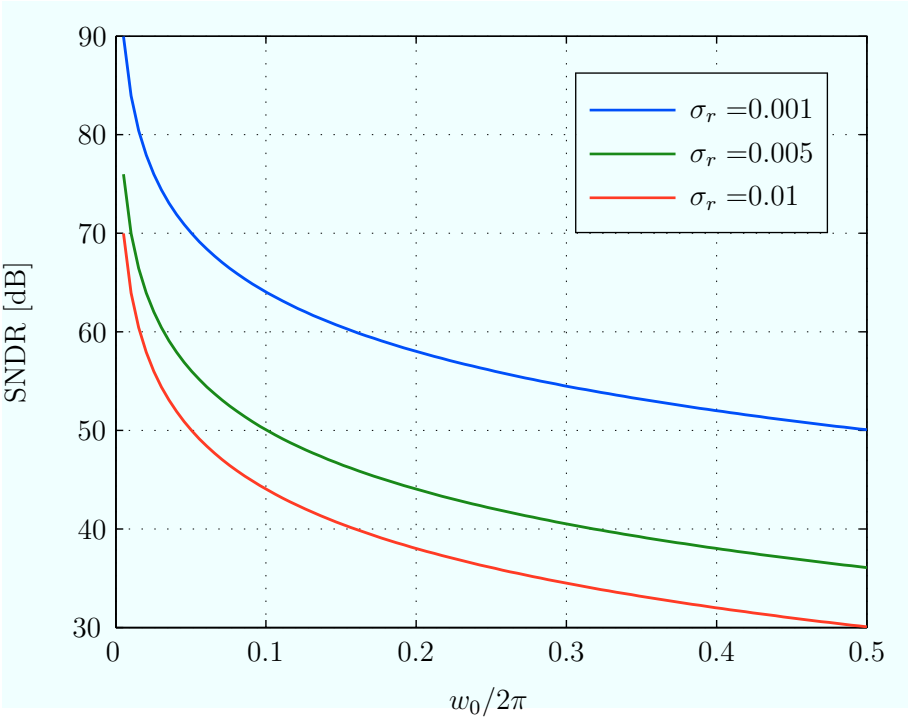


Figure 2.16 – SNDR as a function of input frequency for different RMS skew mismatch levels ($M = 12$)

decreases when the mismatch level increases. The SNDR also decreases when the input frequency increases. This is expected because a small delay affects more the value of the sample when the signal varies quickly than when it varies slowly.

Interestingly the skew mismatch SNDR expression is similar to the jitter SNDR. One way to look at it is to consider the skew mismatches as a periodic jitter. The difference between these two types of timing imperfections is in the shape that the noise takes. The periodicity of the timing skew mismatches yields aliased versions of the input signal, whereas random clock jitter yields a frequency white noise.

2.8 Bandwidth mismatches

The last mismatch type that is considered in this thesis is bandwidth mismatch. Actually, the case of transfer function mismatches mentioned in section 2.5 is relatively

close to the topic of bandwidth mismatches.

Bandwidth mismatches happen when each sub-ADC has a (first order) low-pass transfer function with a different cut-off frequency, as shown in Figure 2.17. Bandwidth mismatches create both frequency-dependent gain mismatches and frequency-dependent phase mismatches, thereby yielding aliases of the input signal spectrum around frequencies multiple of F_s/M. This will be analyzed more deeply later in this section.

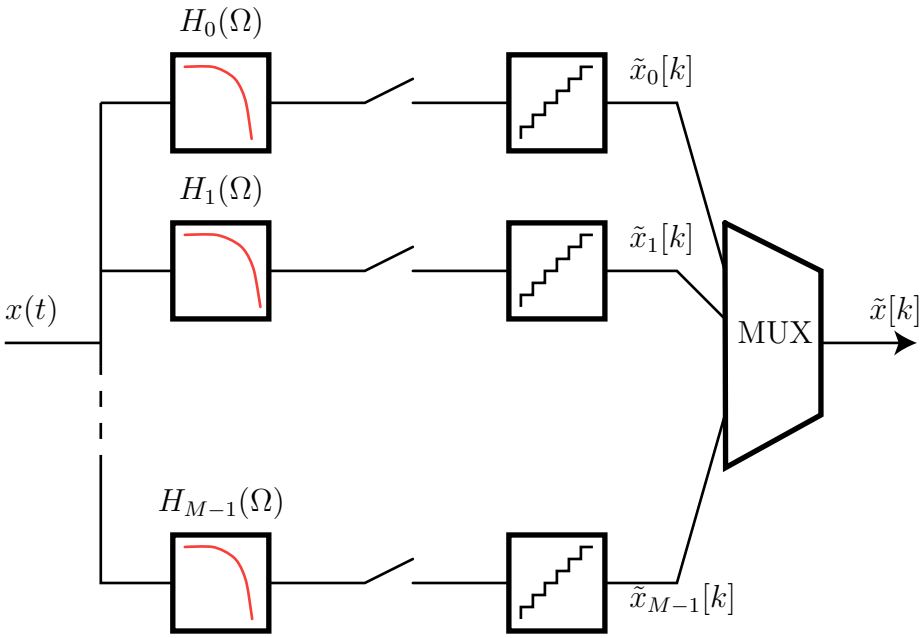


Figure 2.17 – Model of a TIADC with bandwidth mismatches

2.8.1 Circuit sources

In the previous sections, it was assumed that the input signal was not filtered before being sampled. In reality, the continuous signal is sampled by each sub-ADC using a Track & Hold circuit (T&H) that acts approximatively as a low-pass RC filter. Bandwidth mismatches can occur when the frequency characteristics differ of the T&H differ from sub-ADC to sub-ADC. Different architectures of T&H circuits exist but they can generally be conceptually modeled as a switch connected to a capacitor as illustrated in Figure 2.18. The operation of the T&H can be divided into two phases:

— The *tracking phase* during which the switch is closed and the input signal $V_i(t)$ is tracked on the top plate of the capacitor C_s. If R_s is the resistance of the closed switch, then the capacitor top plate voltage $V_o(t)$ verifies the differential equation $\frac{dV_o(t)}{dt} + \frac{1}{\tau}V_o(t) = V_i(t)$, where $\tau = \frac{1}{R_s C_s}$. The translation of this differential equation

into the frequency domain gives the transfer function of the T&H:

$$F(\Omega) = \frac{1}{1 + j\Omega\tau} \tag{2.70}$$

It corresponds to a first-order low-pass filter with a 3dB angular cut-off frequency $\omega_c = 1/\tau$.

— The *holding phase* during witch the switch is opened and the output voltage *held* on the top plate of the capacitor in order to be quantized.

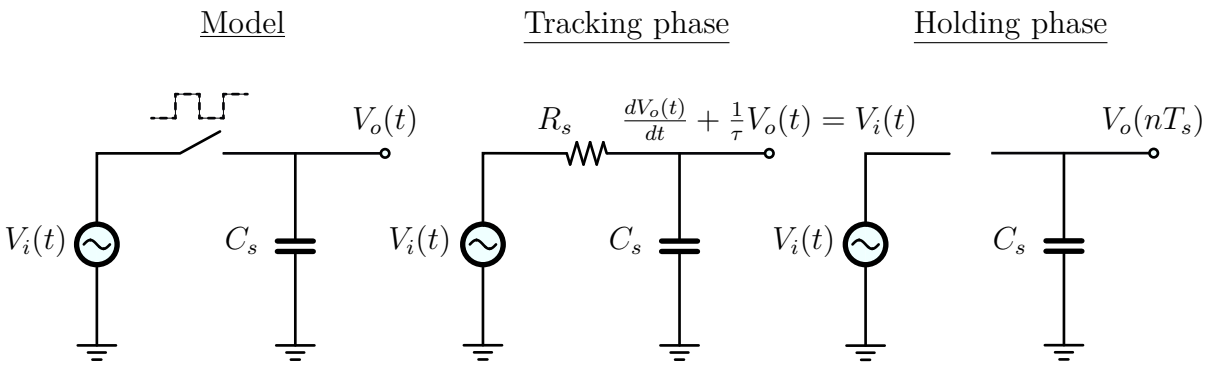


Figure 2.18 – Operation principle of a T&H circuit

Bandwidth mismatches happen when each sub-ADC T&H has different time-constant τ_m (or equivalently a different cut-off frequency). This can be caused either by variations of the switch resistance or by variations of the sampling capacitance from T&H to T&H. It means that the frequency response $\{H_m(\Omega), m \in \mathcal{M}\}$ of each channel is different

$$H_m(\Omega) = \frac{1}{1 + j\Omega\tau_m} \tag{2.71}$$

Different architectures using MOS transistors can be used to realize the switch in the T&H. One of the simplest architecture uses a single NMOS transistor to perform the switching operation, as shown in Figure 2.19. In advanced CMOS technologies, the capacitor is comprised of a bank of Metal-Oxide-Metal (MOM) capacitors or can also be specifically designed (see Chapter 4). The variations of time-constant from channel to channel can be due to both local variations of the on-resistance of the transistors and local variations of the capacitor values.

As seen in section 2.3, capacitor local variations follow Pelgrom's law, which makes them inversely proportional to the area of the capacitor. That is the reason why mismatches are more detrimental on fast designs, where the capacitors sizes are minimized. The on-resistance of a NMOS transistor R_{on} is defined as the slope of the $V_{gs}(I_{ds})$ curve

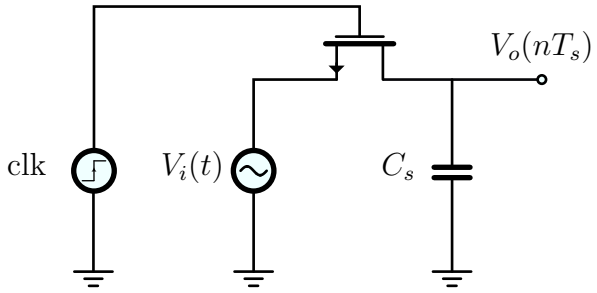


Figure 2.19 – Realization of a basic T&H circuit with a NMOS transistor

in the linear region ($V_{ds} = 0$):

$$R_{on} = \left.\frac{\partial V_{gs}}{\partial I_{ds}}\right|_{V_{ds}=0} \tag{2.72}$$

For square-law devices, the on-resistance is inversely proportional to the width of the transistor and proportional to its length

$$R_{on} = \frac{1}{\mu_n C_{ox} \frac{W}{L} (V_{gs} - V_{th})} \tag{2.73}$$

where μ_n is the mobility of the electrons, C_{ox} the gate oxide capacitance density, W the gate width, L the gate length, V_{gs} the grid source voltage and V_{th} the threshold voltage. Even though, short-channel devices in deep sub-micron technologies do not necessarily follow the square law any more, their on-resistance still increases when the length increases and when the width decreases. The on-resistance is also dependent on the overdrive voltage $V_{gs} - V_{th}$, and since the overdrive voltage depends on V_{th}, the on-resistance is affected by threshold voltage variations.

The plot in Figure 2.20 shows the RMS value of R_{on} as a function of the transistor width. For example, a 1 μm minimum length transistor has a on-resistance RMS value $\sigma_{R_{on}} = 5\ \Omega$, for a nominal resistance of 500 Ω. This represents a RMS mismatch of 1% on the time-constant $\tau = R_s C_s$ of the T&H circuit.

2.8.2 Effects on the output signal

The effects of the bandwidth mismatches on the output signal spectrum can be calculated by applying the same methodology as in Sections 2.6.2 and 2.7.2. Since, we know $H_m(\Omega)$, the output spectrum is gain found by using the general mismatch spectrum expression in Equation 2.48

$$\tilde{X}_{2\pi}(\omega) = \sum_{k=0}^{M-1} \left[\frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{1 + j\left(\omega - k\frac{2\pi}{M}\right) b_m} e^{-j2\pi\frac{km}{M}} \right] X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.74}$$

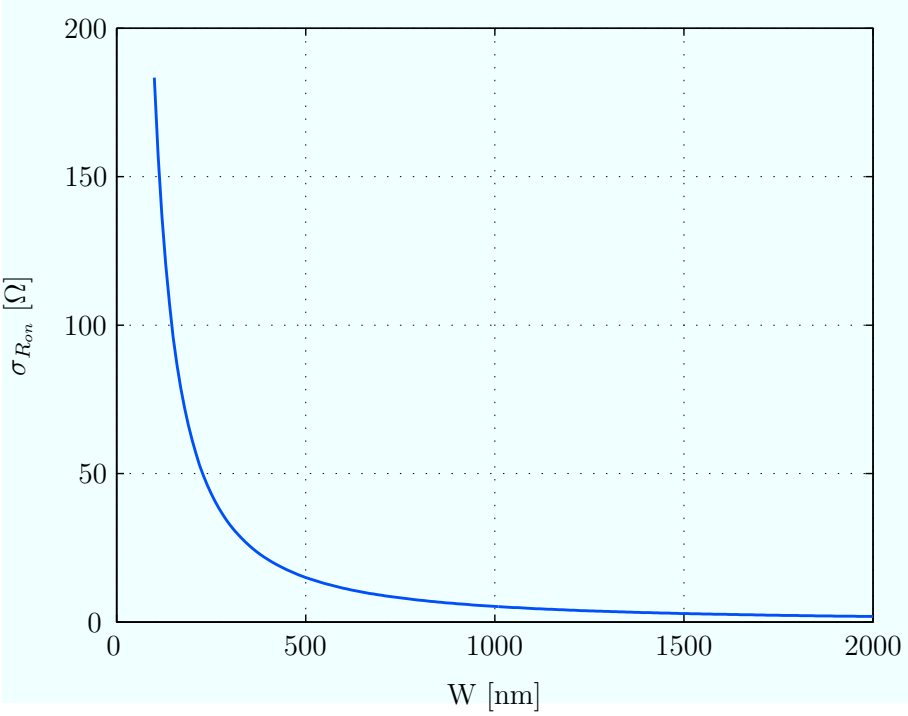


Figure 2.20 – Standard deviation of the on-resistance as a function of transistor dimensions. Calculated from 50 Monte-Carlo runs of a LVT-NMOS transistor in 28 nm FDSOI CMOS technology ($V_{gs} = 1$ V, $L = 30$ nm)

where $b_m = \frac{\tau_m}{T_s}$ represents the time constant of each channel low-pass filter relative to the sampling period.

Each time constant $\{b_m, m \in \mathcal{M}\}$ can be expressed as a sum between a nominal time constant $\bar{b} = \frac{1}{M}\sum_{m=0}^{M-1} b_m$ and a time-constant offset $\{\Delta b_m, m \in \mathcal{M}\}$:

$$b_m = \bar{b} + \Delta b_m \tag{2.75}$$

It is reasonable to suppose that the time-constant offsets are small as compared to the nominal time-constant ($\Delta b_m \ll \bar{b}$), such that the expression 2.74 can be approximated using a first order Taylor expansion:

$$\tilde{X}_{2\pi}(\omega) \approx \frac{X_{2\pi}(\omega)}{1+j\omega\bar{b}} - \sum_{k=1}^{M-1}\left[\frac{1}{M}\sum_{m=0}^{M-1}\frac{\Delta b_m}{\bar{b}}e^{-j2\pi\frac{km}{M}}\right]\frac{j\left(\omega - k\frac{2\pi}{M}\right)\bar{b}}{1+j\left(\omega - k\frac{2\pi}{M}\right)\bar{b}}\frac{X_{2\pi}\left(\omega - k\frac{2\pi}{M}\right)}{1+j\left(\omega - k\frac{2\pi}{M}\right)\bar{b}} \tag{2.76}$$

The previous expression can be made clearer by denoting $Y_{2\pi}(\omega)$ the spectrum of the signal that is filtered by the reference low-pass filter

$$Y_{2\pi}(\omega) = \frac{X_{2\pi}(\omega)}{1+j\omega\bar{b}} \tag{2.77}$$

and B_k the DFT of the normalized time-constant offsets $\beta_m = \frac{\Delta b_m}{b}$

$$B_k = \sum_{m=0}^{M-1} \beta_m e^{-j2\pi \frac{km}{M}} \tag{2.78}$$

The output signal spectrum of the TIADC defined in equation 2.76 can therefore be rewritten in a simplified way

$$\tilde{X}_{2\pi}(\omega) \approx Y_{2\pi}(\omega) - \sum_{k=1}^{M-1} \frac{B_k}{M} \frac{j\left(\omega - k\frac{2\pi}{M}\right)\bar{b}}{1 + j\left(\omega - k\frac{2\pi}{M}\right)\bar{b}} Y_{2\pi}\left(\omega - k\frac{2\pi}{M}\right) \tag{2.79}$$

The spectrum expression in Equation 2.79 has a lot of similarities with the expression of the output spectrum with skew mismatches in Equation 2.66. It contains two main

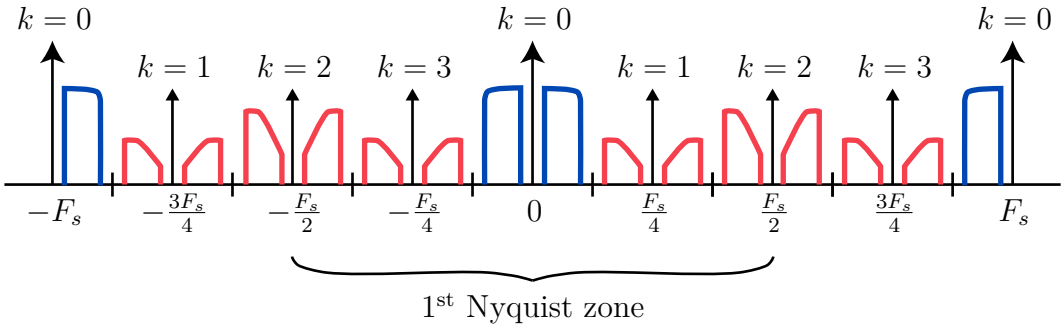


Figure 2.21 – Illustration of TIADC output spectrum with bandwidth mismatch aliases

terms:

- the spectrum of the ideal signal $Y_{2\pi}(\omega)$, which is a low-pass filtered version of the input signal (quite similar the globally delayed signal term in Equation 2.66).
- aliased versions of the ideal signal attenuated by the derivative of the reference low-pass filter, and weighted by the DFT coefficients B_k of the time-constants of the sub-ADCs (quite similar to the weighted and frequency-shifted versions of the derivative in Equation 2.66).

2.8.3 Consequences on the TIADC performance

The output spectrum defined in 2.79 can be used to derive the SNDR of the TIADC in presence of bandwidth mismatches. If the input signal is a sine input, the obtained

output spectrum is almost similar to the one found in the skew mismatch case

$$\tilde{X}_{2\pi}(\omega) \approx A\pi \frac{e^{j\omega\phi}}{1+j\omega_0\bar{b}} \Bigg[\delta_{2\pi}(\omega-\omega_0) + \delta_{2\pi}(\omega+\omega_0) + \sum_{k=1}^{M-1} \frac{B_k}{M} \frac{j\omega_0\bar{b}}{1+j\omega_0\bar{b}} \left(\delta_{2\pi}\left(\omega - k\frac{2\pi}{M} + \omega_0\right) - \delta_{2\pi}\left(\omega - k\frac{2\pi}{M} - \omega_0\right)\right)\Bigg] \tag{2.80}$$

If we compare the bandwidth mismatch spectrum expression to its counter part for skew mismatches, we notice that

— the spurious tones are now weighted by the coefficients B_k as opposed to R_k

— the spurious tones are multiplied by $\frac{j\omega_o\bar{b}}{1+j\omega_0\bar{b}}$ as opposed to $j\omega_0$

Using this similarity between the two expressions makes the calculation of the bandwidth mismatch SNDR straightforward. The calculation is done by using the SNDR expression in Equation 2.69 and by replacing the RMS skew mismatch level σ_r by the RMS bandwidth mismatch level $\sigma_\beta = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1}\beta_m^2}$ and by replacing $\omega_0 = |j\omega_0|$ by $\frac{\omega_0\bar{b}}{\sqrt{1+(\omega_0\bar{b})^2}} = \left|\frac{j\omega_0\bar{b}}{1+j\omega_0\bar{b}}\right|$ It yields the following SNDR expression for bandwidth mismatches:

$$\text{SNDR} = 20\log_{10}\left(\frac{\sqrt{1+(\omega_0\bar{b})^2}}{\omega_0\bar{b}\sigma_\beta}\right) \tag{2.81}$$

The above SNDR expression can equivalently be represented as a function of the nominal angular cut-off frequency $\bar{\omega}_c = 1/\bar{b}$ of the low-pass filter:

$$\text{SNDR} = 20\log_{10}\left(\frac{\sqrt{1+(\frac{\omega_0}{\bar{\omega}_c})^2}}{\frac{\omega_0}{\bar{\omega}_c}\sigma_\beta}\right) \tag{2.82}$$

It is interesting to notice that when the input frequency is small as compared to the cut-off frequency ($\omega_0 \ll \bar{\omega}_c$), the SNDR looks similar to the SNDR in presence of skew mismatches:

$$\text{SNDR} \approx -20\log_{10}\left(\omega_0\frac{\sigma_\beta}{\bar{\omega}_c}\right) \tag{2.83}$$

The reason is that, at a low input frequency, the bandwidth mismatch noise is dominated by phase mismatches. Conversely, if the input signal frequency is above the nominal cut-off frequency ($\omega_0 \gg \bar{\omega}_c$) [13], the SNDR becomes independent of the input signal frequency, and looks like the SNDR expression found for gain mismatches:

$$\text{SNDR} = -20\log_{10}(\sigma_\beta) \tag{2.84}$$

13. This is never the case in practice because this results in the input signal being very attenuated.

These observations tell us that when the input frequency increases the bandwidth mismatch noise becomes increasingly dominated by frequency-dependent gain mismatches.

The SNDR as function of the input signal frequency is plotted in Figure 2.22, for a nominal cut-off frequency equal to the Nyquist frequency.

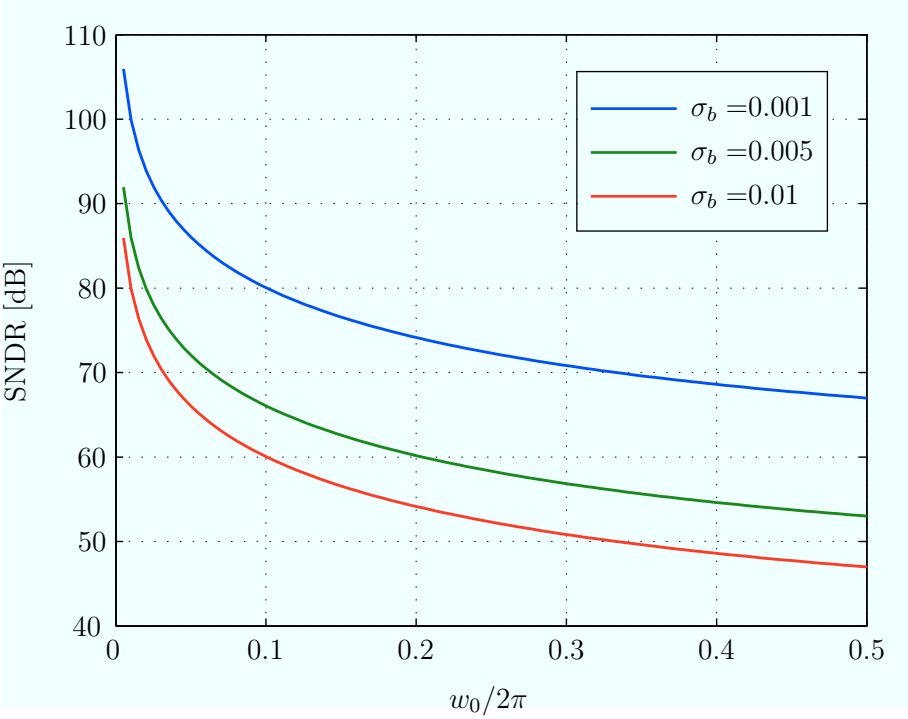


Figure 2.22 – SNDR vs. input frequency for different RMS bandwidth mismatch levels ($M = 12$, cut-off frequency $\bar{b} = 1/2\pi$)

2.9 Conclusion

This chapter presents the sources of the mismatches and analyzes their effects on the output signal. It was shown that the mismatches mostly come from random variations in the devices constituting the circuit, principally MOS transistors. The mismatches were separated in different categories: offset mismatches, gain mismatches, timing skew mismatches and bandwidth mismatches. For each of those categories, a close form expression of the SNDR as a function of the RMS mismatch level was derived in order to give insight on how the mismatches contribute to increase the noise in the TIADC output signal. For offset mismatches we also derived a close form expression of the SFDR in order to better analyze the spurious nature of offset mismatch noise. The expressions that were derived in this chapter will prove useful in the two following chapters to analyze the performance of the different mismatch calibration techniques.

Chapter 3

Background digital mismatch calibration

3.1 Introduction

This chapter constitutes the heart of this dissertation. It presents fully digital mismatch calibration techniques for TIADCs and perhaps, as importantly, deeply analyzes the input signal requirements that make those calibration techniques work. The introduction of each of the calibration techniques aims at giving the reader an intuition about their principle. However, intuitions are not proofs, which is why each technique is carefully analyzed with simulations or statistical tools when possible. The performance of each technique is voluntary analyzed under ideal conditions (i.e. no noise, only one mismatch at a time) in order to focus on the intrinsic limitations of each of the solutions. A more practical case is treated in Chapter 4 with a circuit implementation of the calibration techniques.

Offset and gain mismatch calibrations are presented first in Sections 3.4 and 3.5. Their concepts are rather simple and inspired from prior art. The offset mismatch calibration equalizes the sub-ADC output signal averages whereas the gain mismatch calibration equalizes the sub-ADC output signal variances. Both techniques require the signal to be non-stationary and to verify certain specific properties. Analyzing those requirements goes beyond existing research on the subject, which usually assumes that the input signal is a realization of a Wide-Sense Stationary (WSS) random process. We extend the – widely admitted – statement that says that blind digital mismatch calibration techniques do not work with a sine input signal with an input frequency multiple of F_s/M.

Skew mismatch calibration is presented in Section 3.6. The calibration is based on the equalization, among the sub-ADCs, of the covariances between each sub-ADC's output signal and its respective derivative. As opposed to previous work, the calibration is non

adaptive, which means that estimates of the timing offsets are available after a single iteration and no feedback loop is required. This greatly reduces the calibration time and eliminates the risk of instability that potentially occurs with adaptive techniques. The timing offsets of the sub-ADCs are adjusted digitally by doing a first-order interpolation of the signal. The interpolation only requires a derivative FIR filter with fixed coefficients.

Finally, a new bandwidth mismatch calibration technique is presented in Section 3.7. The technique is based on measuring the powers of two differently filtered versions of each of sub-ADC's output signal. It is possible to show that equalizing the ratios between those two powers among all the sub-ADCs enables to detect the bandwidth mismatches independently of the gain mismatches. The bandwidth mismatch calibration is also done in a single iteration and uses a fixed-coefficient FIR filter to adjust each sub-ADC's bandwidth.

3.2 Requirements on the input signal

3.2.1 Foreword

The calibration techniques presented in this chapter require the input signal $x(t)$ of the TIADC to be the realization of non stationary process $\{X_t, t \in \mathbb{R}\}$ that has certain constraints. The aim of this section is to provide a few mathematical tools and notations to analyze such processes. This study is a necessary preliminary step to understand the foundations of the calibration techniques presented thereafter. The published papers on blind background calibration techniques often omit to carefully analyze the assumptions that they make about the input signal. In my opinion, this analysis is essential because it highlights the limitations of fully digital mismatch calibration techniques and it helps analyze and solve the problems that arise when testing the algorithms with real circuits.

3.2.2 Wide Sense Stationary processes

Wide Sense Stationary (WSS) processes are a special case of random processes often used in signal processing because they are appropriate for calculations. A continuous time random process $\{X_t, t \in \mathbb{R}\}$ is said to be wide-sense stationary (WSS) if it verifies the two following conditions:

— Its expected value m_X is independent of time

$$\mathrm{E}(X_t) = m_X \ \ \forall t \in \mathbb{R} \tag{3.1}$$

— Its autocovariance $r_{XX}(t, \tau)$ is a function that only depends on the time lag τ

between X_t and $X_{t+\tau}$:

$$\mathrm{E}((X_t - m_X)(X_{t+\tau} - m_X)) = r_{XX}(t,\tau) = r_{XX}(\tau) \ \forall t \in \mathbb{R} \tag{3.2}$$

If the two conditions above are met, it is possible to define the power spectral density (PSD) $R_{XX}(\Omega)$ of the random process $\{X_t, t \in \mathbb{R}\}$. The PSD is the Fourier transform of the above expression with respect to the variable τ:

$$R_{XX}(\Omega) = \int_{-\infty}^{+\infty} r_{XX}(\tau) e^{-j\Omega\tau} d\tau \tag{3.3}$$

If a continuous-time process is WSS then any discrete process $\{X_n, n \in \mathbb{N}\} = \{X_t, t = nT_s\}$, obtained by sampling the process $\{X_t\}$, is also WSS and

— Its expected value is time invariant and is equal to the expected value of the continuous time process:

$$\mathrm{E}(X_n) = m_X \ \forall n \in \mathbb{N} \tag{3.4}$$

— Its autocovariance function only depends on the shift l between X_n and X_{n+l}

$$\mathrm{E}((X_n - m_X)(X_{n+l} - m_X)) = r_{XX}(nT_s, lT_s) = r_{XX}[l] \ \forall n \in \mathbb{N} \tag{3.5}$$

The discrete time PSD (DT-PSD) $\breve{R}_{XX}(\omega)$ of the process is the DTFT of the discrete autocovariance function $r_{XX}[l]$:

$$\breve{R}_{XX}(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} R_{XX}\left(\frac{1}{T_s}(\omega - k2\pi)\right) \tag{3.6}$$

where $\omega = \Omega T_s$.

3.2.3 Non-stationary processes

Although convenient for calculations, WSS random processes do not necessarily describe communication signals very well. Indeed, communication signals are often linear combinations of random processes and periodic functions, which destroys the stationarity property. In the next sections, we show that the proposed calibration algorithms are valid even if the signal is a realization of a non-stationary process provided that the signal verifies some properties.

As opposed to WSS random processes, of which they are a generalization, non-stationary processes have an expectation and an autocovariance that vary in time. If $\{X_t, t \in \mathbb{R}\}$ is a non-stationary process then

— $E(X_t) = m_X(t)$, where $m_X(t)$ is assumed to be a continuous function of time that

is square summable

— $\mathrm{E}\left(X_t X_{t+\tau}\right) = r_{XX}(t,\tau)$, where $r_{XX}(t,\tau)$ is assumed to be a continuous function of t and τ and square summable with respect to both variables

The expected value can be expressed in the frequency domain by taking the CTFT of $m_X(t)$:

$$M_X(\Omega) = \int_{-\infty}^{+\infty} m_X(t) e^{-j\Omega t} dt \tag{3.7}$$

The autocovariance function also has its counterpart in the frequency domain. However, since the autocovariance is a function of two time variables, two frequency domains can be defined: one associated to the variable t and one associated to the variable τ. The angular frequency associated with the time variable τ is denoted Ω for consistency with the PSD of a WSS random process. The angular frequency associated with the time variable t is denoted Υ. As a consequence, the PSD of the process now depends on the time t:

$$R^t_{XX}(\Omega) = \int_{-\infty}^{+\infty} r_{XX}(t,\tau) e^{-j\Omega\tau} d\tau \tag{3.8}$$

and the CTFT of the autocovariance function $r_{XX}(t,\tau)$ with respect to the variable t depends on the time lag τ:

$$R^\tau_{XX}(\Upsilon) = \int_{-\infty}^{+\infty} r_{XX}(t,\tau) e^{-j\Upsilon t} dt \tag{3.9}$$

3.2.4 Application to digital communications signals

In this section we show that how our probabilistic assumptions apply to wideband communication signals. Wideband digital communications signals generally consist of a linear combination of several channels modulated at different carrier frequencies [1, 2] .

$$x(t) = \sum_{j=0}^{N_c} c_j(t) \tag{3.10}$$

where N_c is the number of channels and $c_j(t)$ corresponds to the signal of a single channel. The signal of a single channel $c_j(t)$ is obtained by modulating an in-phase baseband waveform $i_j(t)$ and a quadrature baseband waveform $q_j(t)$ at the carrier frequency Ω_i such that

$$c_j(t) = i_j(t)\cos\left(\Omega_j t\right) - q_j(t)\sin\left(\Omega_j t\right) \tag{3.11}$$

The baseband waveforms correspond respectively to the real and imaginary parts of the complex baseband waveform $s_j(t)$, obtained by using spectral shaping on the stream of

complex-valued symbols $\{s_j[k]\}$ to be transmitted:

$$s_j(t) = \sum_{k=-\infty}^{+\infty} s_j[k] p(t - kT^j_{\text{symb}}) \tag{3.12}$$

where $p(t)$ is the pulse shaping filter and $1/T^j_{\text{symb}}$ is the symbol rate of channel j. Often, the pulse shaping filter is a Root-Raised Cosine (RRC) filter having an approximate bandwidth of $1/T^j_{\text{symb}}$.

The complex baseband waveform $s_j(t)$ can be modeled as being the realization of the random process

$$S^j_t = \sum_{k=-\infty}^{+\infty} S^j_k p(t - kT^j_{\text{symb}}) \tag{3.13}$$

where the $\{S^j_k, k \in \mathbb{N}\}$ are independent random variables with zero-mean [1] modeling the stream of symbols. Obviously, this process is stationary for the first order with expected value zero because

$$\mathrm{E}\left(S^j_t\right) = \sum_{k=-\infty}^{+\infty} \mathrm{E}\left(S^j_k\right) p(t - kT^j_{\text{symb}}) = 0 \tag{3.14}$$

The autocovariance function $r^j_{SS}(t, \tau)$ is however periodic of period T^j_{symb} if the pulse shaping filter has a bandwidth larger than $1/T^j_{\text{symb}}$ [80]. This is generally the case because pulse shaping filters classically have an excess bandwidth $(1 + \alpha)T_{\text{symb}}$, where α is the excess bandwidth factor with a typical value of 0.25 [1, 2]. Because of the periodicity, the random process $\left\{S^j_t, t \in \mathbb{R}\right\}$ is called cyclostationary for the second moment and its autocovariance function verifies

$$r^j_{SS}(t, \tau) = r^j_{SS}\left(t + T^j_{\text{symb}}, \tau\right) \tag{3.15}$$

As a consequence, the signal corresponding to a single channel is (in general) also cyclostationary. The autocovariance function of the associated random process is

$$r^j_{CC}(t, \tau) = \frac{1}{2} r^j_{II}(t, \tau) \left(\cos\left(\Omega_j(2t + \tau) + \cos(\tau)\right)\right) + \frac{1}{2} r^j_{QQ}(t, \tau) \left(\cos\left(\Omega_j(2t + \tau) - \cos(\tau)\right)\right) \tag{3.16}$$

where $r_{II}(t, \tau)$ and $r_{QQ}(t, \tau)$ are the T^j_{symb}-periodic autocovariance functions associated to the in-phase baseband waveform and the quadrature baseband waveform respectively. The channel autocovariance function is a product between a T^j_{symb}-periodic function and a $T^j_c/2$-periodic function, $T^j_c = 2\pi/\Omega_j$ being the period of the carrier. Therefore, the overall period $T^j = n_1 T^j_c/2 = n_2 T^j_{\text{symb}}$ exists only if $2T^j_{\text{symb}}/T^j_c$ is rational, i.e. if it can be

1. The constellations used in today's communication signals (mostly QAM) are centered around 0 such that the mean of the symbol stream can be considered to be 0 if all symbols are equally likely to be picked.

written

$$T^j = \frac{2T^j_{\text{symb}}}{T^j_c} = \frac{n_1}{n_2} \text{ with } (n_1, n_2) \in \mathbb{N}^{*2} \tag{3.17}$$

If $2T^j_{\text{symb}}/T^j_c$ is not rational then the random process C^j_t associated to the channel signal is not cyclostationary. However, when $2T^j_{\text{symb}}/T^j_c$ is rational, it is interesting to have an idea of the typical value for T^j. For that, we can take the example of the TV cable application using DOCSIS 3.0 [1] as a standard. In the DOCSIS 3.0 standard, the data is transmitted on contiguous 6 MHz channels on a frequency band that goes from 54 MHz to 1002 MHz. In the US, the carrier frequencies for cable TV are fixed and standardized by the EIA. The symbol rate can either be 5.360357 Msymb/s or 5.056941 Msymb/s depending on the type of modulation that is used (QAM64 or QAM256). For the example, let us consider the carrier frequency 993 MHz (channel 157) and the data rate 5.360357 Msymb/s. These parameters give the following irreducible n_1/n_2 ratio:

$$\frac{n_1}{n_2} = \frac{2 \times 993000000}{5360357} \tag{3.18}$$

which in turn leads to $T^j = 5360357 \times T_{\text{symb}} = 1$ s. This value is obtained assuming that the carrier frequency is perfectly adjusted. If there is a shift as small as 0.1 Hz on the value of the carrier frequency the ratio becomes

$$\frac{n_1}{n_2} = \frac{9930000001}{26801785} \tag{3.19}$$

which yields $T^j = 26801785 \times T_{\text{symb}} = 5$ s. This simple example shows that in practice the cyclostationarity period of a single channel can be of several seconds. It is easy to imagine that the cyclostationarity period of the whole signal can become very big. Indeed, the overall cyclostationarity period T must be the smallest common multiplier of all the channels cyclostationarity periods T^j_c such that $T = n_j T^j_c \quad \forall j \in \{1, \cdots, N_c\}$.

As we will see in the next sections, the calibration algorithms do not need more than 1 second of signal in order to reach a sufficient accuracy. Therefore, at this scale of observation, it is reasonable to assume that the input signal is non-stationary.

Below is a summary of the statistical assumptions that we make on the random process $\{X_t, t \in \mathbb{R}\}$ associated to the TIADC input signal $x(t)$:

— Zero expected value: $E(X_t) = m_X(t) = 0 \quad \forall t \in \mathbb{R}$ (stationarity for the first moment)

— Time-dependent autocovariance function $E(X_t X_{t+\tau}) = r_{XX}(t, \tau) \quad \forall t \in \mathbb{R}$

3.3 Overall mismatch calibration architecture

The next sections present each of the calibration techniques separately, assuming for each of them, that the other mismatches are absent, or have already been corrected. In reality, all the mismatches (or a subset of them) are present at the same time and they must be calibrated in a precise order. A block diagram illustrating the order of the calibration functions is shown in Figure 3.1.

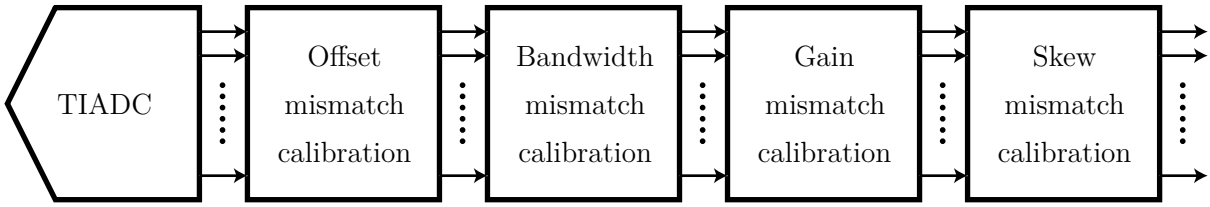


Figure 3.1 – Overall mismatch calibration structure

The offset mismatch calibration is done first because the other mismatch calibration techniques require the sub-ADC's output signals to have zero-mean. The bandwidth mismatch calibration is done next because bandwidth mismatches yield frequency-dependent phase and gain mismatches. These frequency-dependent mismatches must be corrected before skew and gain mismatch calibration happen. If the bandwidth mismatch calibration was not done before the gain mismatch calibration, the gain mismatch calibration unit would measure both the frequency dependent gain and the static gain without being able to distinguish them. Similarly, the skew mismatch calibration unit would measure both the frequency dependent phase mismatches and the timing mismatches. The bandwidth mismatch calibration, on the other hand, is designed to be insensitive to static gain mismatches and static timing mismatches.

The gain mismatch calibration has to be performed before the skew mismatch calibration. Indeed, the skew mismatch calibration relies on the calculation of the autocovariance function at each sub-ADC's output. Performing the gain mismatch calibration first guarantees that the calculation of the autocovariance function is not affected by the sub-ADC static gains.

This chapter does not follow the structural order of the calibration blocks. The bandwidth mismatch calibration technique is described last because it is better understood after gaining insight about the gain and skew mismatch calibration principles.

3.4 Offset mismatch calibration

3.4.1 Introduction

As explained in Chapter 2, the offset mismatch problem arises when the sub-ADCs have different offsets. The offset mismatch calibration can be done by equalizing the sub-ADC output averages. The overall architecture of the calibration is shown in Figure 3.2. It is divided into two phases: the first one is to estimate the offsets of the sub-ADCs; the second one is to correct the sub-ADC output signals given the estimated offsets.

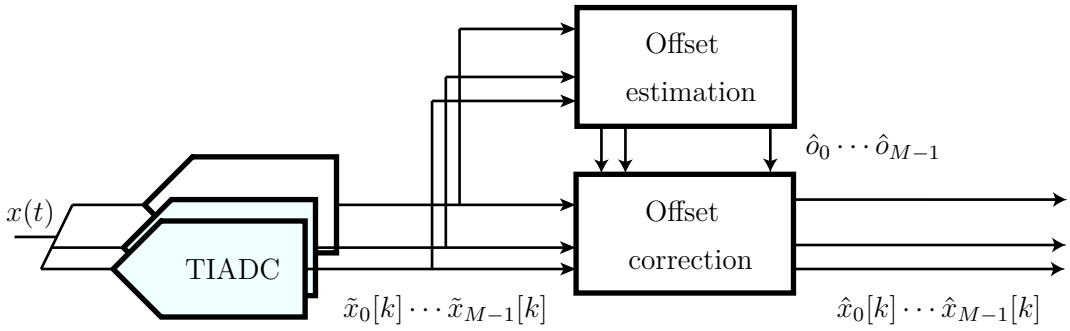


Figure 3.2 – Offset mismatch calibration block diagram

3.4.2 Offset mismatch correction

3.4.2.1 Principle

The offset estimation algorithm provides the estimated offsets $\{\hat{o}_m, m \in \mathcal{M}\}$ of the sub-ADCs. Correcting the output signal of the TIADC, knowing these offsets, is done by subtracting each offset from the output signal of the corresponding sub-ADC. The corrected sub-ADC output signals $\{\hat{x}_m[k], m \in \mathcal{M}\}$ become:

$$\hat{x}_m[k] = \tilde{x}_m[k] - \hat{o}_m \tag{3.20}$$

3.4.2.2 Circuit implementation

The block diagram of the offset mismatch correction is shown in Figure 3.3. The offset mismatch correction unit requires M adders that work at the rate F_s/M of the sub-ADCs.

Although the offset mismatch correction seems simple, quantization issues may limit its performance because the offset cannot be corrected with a precision of more than half a LSB. Indeed, let assume that the uncorrected signal and the corrected signal are coded

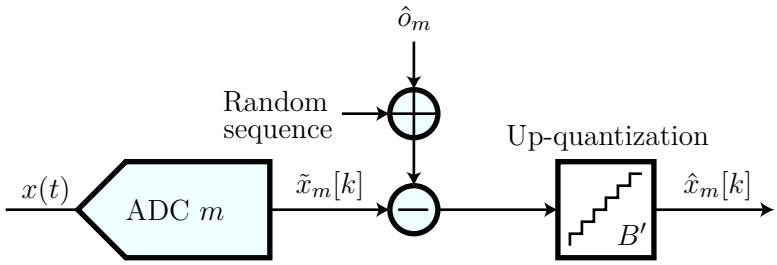


Figure 3.3 – Offset mismatch correction block diagram

with the same number of bits B, and that $\hat{o}_m$ is a perfect estimate of the real offset o_m, that is $\hat{o}_m = o_m$. Each offset estimate $\hat{o}_m$ can be decomposed as a sum of an integer number q_m of LSBs plus a fraction ξ_m of a LSB.

$$\hat{o}_m = q_m \text{LSB} + \xi_m \text{LSB} \tag{3.21}$$

where $q_m \in \mathbb{Z}$ and $0 \leq \xi_m < 1$. The correction operation consists in subtracting the above estimated offset from the quantized sub-ADC output signal $\tilde{x}_m^B[k] = Q_B(\tilde{x}_m[k]) \times \text{LSB}$ and then in quantizing the result with B bits:

$$\hat{x}_m^B[k] = Q_B(\hat{x}_m[k]) = \left(\tilde{x}_m^B[k] - q_m\right) \text{LSB} - Q_B(\xi_m)\text{LSB} \tag{3.22}$$

where $Q_B(\cdot)$ denotes the quantization function. Depending on the fractional part of the offsets, $Q_B(\xi_m)$ can either take the value 0 or 1:

$$Q_B(\xi_m) = \begin{cases} 0 \text{ if } \xi_m < 0.5 \\ 1 \text{ if } \xi_m \geq 0.5 \end{cases} \tag{3.23}$$

Thus, the difference between the signal that is quantized after offset correction and the signal that is not quantized after correction is:

$$\epsilon_m = \left[Q_B(\xi_m) - \xi_m\right] \text{LSB} \tag{3.24}$$

This error ϵ_m acts as a residual offset, different for each sub-ADC, and is comprised between -0.5 LSB and $+0.5$ LSB. This is a serious limitation as it creates – at worse – a noise with the same power as the quantization noise, thereby reducing the ENOB of the TIADC by half a bit (or equivalently reducing the SNDR by 3 dB). This limitation is illustrated in Figure 3.4a and Figure 3.4b where the output TIADC spectrums before and after mismatch correction are shown. Most of all, the quantization at the output of the offset correction stage degrades the SFDR, the worst case being when the noise power is concentrated in only one or two spurious tones (see chapter 2).

The most straightforward technique to alleviate this problem is to increase the number of bits at the output of the offset mismatch correction stage. Doing so enables to reduce the residual offset mismatch noise, thereby increasing the SNDR and the SFDR. For example, quantizing the corrected signal on $B' = B + 3$ bits yields residual offsets 8 times lower because the residual offsets become

$$\epsilon'_m = [Q_{B'}(\xi'_m) - \xi'_m]\,\mathrm{LSB}' \tag{3.25}$$

where $\mathrm{LSB}' = \mathrm{LSB}/8$ and ξ'_m is the fractional part of the estimated offset when the estimated offset is expressed as a function of LSB′. Figure 3.4c shows that 3 additional bits almost enable to reach the ideal SNDR of a 9-bit TIADC ($\approx$ 56 dB). Although conceptually easy, increasing the number of bits at the output of the offset correction stage is not a benign operation when it comes to manufacturing the circuit because increases the number of bits also increases the number of wires to route. Increasing the number of bits is acceptable only up to a certain level, for instance 2 to 3 additional bits for a 9-bit TIADC.

Digital dithering can be used to further increase the SFDR (while still maintaining the same level of residual offset mismatch noise). The idea is to render the residual offsets random in order to spread the mismatch noise across the entire spectrum. This can be done during the quantization to B' bits by making $Q_{B'}(\xi'_m)$ vary randomly in time, but with an average equal to ξ'_m. Such dithering can be done by adding a random sequence with mean 0 to ξ'_m before the quantization, such that the residual offsets become

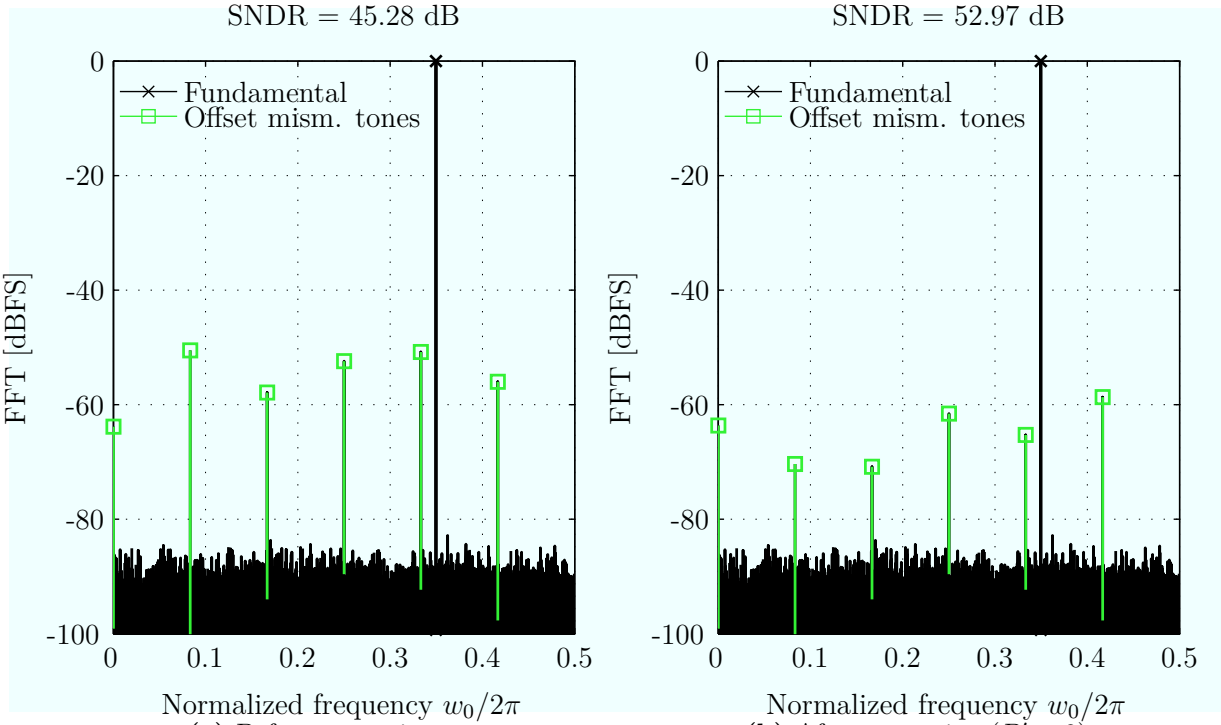
$$\epsilon'_m[k] = [Q_{B'}\left(\xi'_m + w[k]\right) - \xi'_m]\,\mathrm{LSB}' \tag{3.26}$$

where $w[k]$ is a random sequence generating 1 or 0 with equal probabilities. Using this digital dithering technique is equivalent to replacing a fixed residual offset by a varying residual offset with average 0. The TIADC output spectrum after offset mismatch correction and dithering is shown in Figure 3.4d. As predicted, the SNDR is similar to the correction without dithering but the SFDR is greatly increased.

3.4.3 Offset mismatch estimation

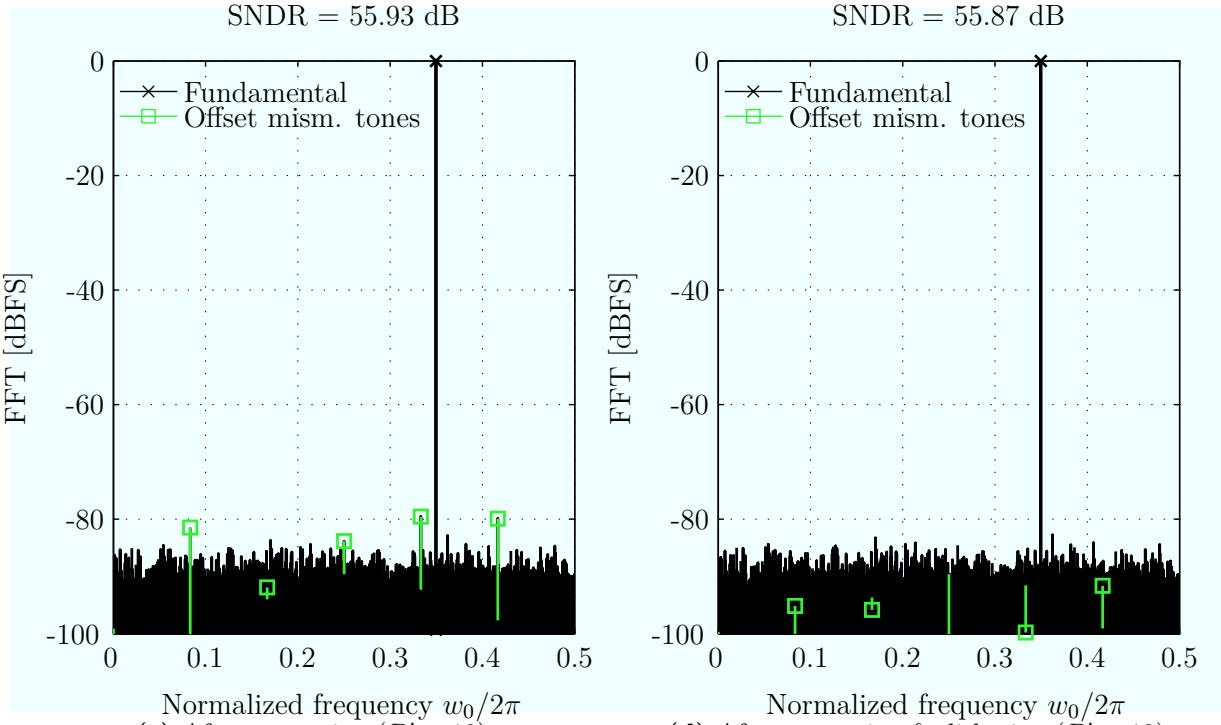
3.4.3.1 General idea

The offset mismatch estimation block uses the fact that the sub-ADCs output signals have the same average if there are no offset mismatches. If there are offset mismatches, the average of each sub-ADC's output signal is shifted by the offset of the sub-ADC. If, as assumed in Section 3.2, the input signal has a zero-mean, an estimated offset $\hat{o}_m$ of the



(a) Before correction

(b) After correction ($B' = 9$)



(c) After correction ($B' = 12$)

(d) After correction & dithering ($B' = 12$)

Figure 3.4 – Simulation of spectrums after offset mismatch correction for different quantization levels and with or without dithering ($B = 9$, $M = 12$, $N_{\text{FFT}} = 12 \times 4096$, $\sigma_o = 1$ LSB, true offset values used for correction)

offset o_m of sub-ADC m can be estimated by calculating the average of its output signal $\tilde{x}_m[k]$, as illustrated in Figure 3.5.

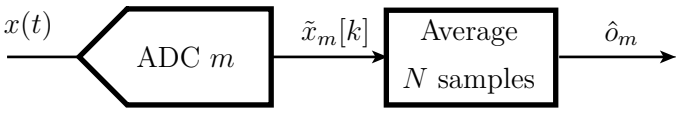


Figure 3.5 – Offset mismatch estimation block diagram

3.4.3.2 Theoretical explanation

The idea described above can be explained using theory. For that, we need to assume that the input signal is a realization of a random process that is stationary for the first moment. When there are no offset mismatches, the output signals of the sub-ADCs have the same expected value. Indeed, each sequence of output samples $\tilde{x}_m[k]$ associated to each sub-ADC is a sampled realization of the input first-order stationary process $\{X_t\}$. Each sequence is consequently a realization of a sub-process $\{X_k^m, m \in \mathcal{M}\}$ with an expected value equal to the expected value of the overall process (assumed to be zero):

$$\mathrm{E}\left(X_k^m\right) = \mathrm{E}\left(X_n\right) = \mathrm{E}\left(X_t\right) = 0 \text{ for } m \in \mathcal{M} \tag{3.27}$$

If there are offset mismatches the expected values of the sub-sequences are each shifted by the offset of the respective sub-ADC such that:

$$\mathrm{E}\left(\tilde{X}_k^m\right) = \mathrm{E}\left(X_k^m + o_m\right) = o_m \tag{3.28}$$

where $\{\tilde{X}_k^m, m \in \mathcal{M}\}$ are the sub-processes with offset mismatches. The above expression shows that knowing the expected value of one sub-ADCs output signal is equivalent to knowing the value of the sub-ADC offset.

In reality, it is impossible to know the theoretical value of the expected value in expression 3.28. It is however well known that the classical average is an unbiased estimator of the expected value. If we assume that the sub-process is ergodic for the first moment, the average of an infinitely long realization of the sub-process converges toward the value of the offset:

$$\lim_{N \to +\infty} \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_m[k] = \mathrm{E}\left(\tilde{X}_k^m\right) = o_m \tag{3.29}$$

This statement implies that estimates $\hat{o}_m$ of the sub-ADC offsets can be obtained by

averaging the sub-ADC output signals over a finite number N of samples:

$$\hat{o}_m = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_m[k] \tag{3.30}$$

3.4.3.3 Circuit implementation

Implementing the offset estimation scheme with digital circuits is simple because the offset estimation only requires one adder and one register per sub-ADC. The adder is used to perform the accumulation (the division by N is not necessary) and the register is used to store the accumulated value.

The next section describes how to use the offset estimates to correct the signal.

3.4.3.4 Accuracy analysis

Something important to know is how to determine the number N of samples that is required to get a sufficient estimation accuracy. A good way to analyze the accuracy of the offset estimator is to look at its variance. Indeed, it was shown in Chapter 2 that the SNDR is directly linked to the RMS value σ_o of the offsets. If we assume that we can correct the offset mismatches, the residual offset RMS value after correction $\sigma_{\tilde{o}}$ would be

$$\sigma_{\tilde{o}} = \sqrt{\frac{1}{M} \sum_{k=0}^{M-1} (\hat{o}_m - o_m)^2} \approx \sqrt{\text{var}\left(\hat{O}_m\right)} \; \forall m \in \mathcal{M} \tag{3.31}$$

where $\{\hat{O}_m = \frac{1}{N} \sum_{k=0}^{M-1} \tilde{X}_k^m, m \in \mathcal{M}\}$ are the offset estimators associated to each sub-ADC. We are therefore interested in calculating:

$$\text{var}\left(\hat{O}_m\right) = \text{E}\left(\hat{O}_m^2\right) - o_m^2 \tag{3.32}$$

Case $N = +\infty$ A possibility to calculate the variance of the offset estimator is to analyze the offset estimator in the frequency domain. For the analysis, we assume that the signal is also second-order stationary, which makes the calculation of its PSD possible [2]. In that case, the variance of the offset estimator is the integral of the PSD $R_{\hat{O}\hat{O}}^m(\omega')$ at the output of the estimator

$$\text{var}\left(\hat{O}_m\right) = \frac{1}{2\pi} \int_0^{2\pi} R_{\hat{O}\hat{O}}^m(\omega') d\omega' \tag{3.33}$$

2. The analysis would also hold for non second-order stationary signals but this makes the PSD dependent on the time.

where $\omega' = M\Omega T_s$ is the normalized angular frequency associated to the signal sampled at a rate F_s/M. To calculate the variance expression, we need to know the PSD at the output of the offset estimator, which is straight forward if we interpret the finite-length average done in the estimation as a FIR filter. The averaging filter has a z-transform:

$$H_{\hat{O}}(z) = \frac{1}{N}\sum_{k=0}^{N-1} z^k \tag{3.34}$$

The averaging filter is applied to the sub-process $\left\{\tilde{X}_k^m\right\}$, which PSD $\breve{R}_{\tilde{X}\tilde{X}}^m(\omega')$ contains the aliases of the full-rate PSD:

$$\breve{R}_{\tilde{X}\tilde{X}}^m(\omega') = \frac{1}{M}\sum_{k=0}^{M-1} \breve{R}_{XX}\left(\frac{\omega'}{M} - k\frac{2\pi}{M}\right) e^{-jm\left(\frac{\omega'}{M} - k\frac{2\pi}{M}\right)} \tag{3.35}$$

The frequency response of the averaging filter is

$$\breve{H}_{\hat{O}}(\omega') = \frac{\sin(N\omega'/2)}{N\sin(\omega'/2)} \tag{3.36}$$

It converges toward the Kronecker delta function when the average length goes toward infinity:

$$\lim_{N\to+\infty} \breve{H}_{\hat{O}}(\omega') = \delta[\omega'] \tag{3.37}$$

The PSD at the output of the estimation filter when $N \to +\infty$ therefore only contains information at DC:

$$\breve{R}_{\hat{O}\hat{O}}^m(\omega') = \breve{R}_{\tilde{X}\tilde{X}}^m(\omega')\left|\breve{H}_{\hat{O}}(\omega')\right|^2 \tag{3.38}$$

$$= \frac{1}{M}\sum_{k=0}^{M-1} \breve{R}_{XX}\left(\frac{2k\pi}{M}\right) e^{j2\pi\frac{km}{M}}\delta[\omega'] \tag{3.39}$$

The variance of the offset estimator is obtained by integrating the above expression (Equation 3.33). Due to the Kronecker function, the integration effectively happens between 0 and 0:

$$\mathrm{var}\left(\hat{O}_m\right) = \frac{1}{2\pi M}\sum_{k=0}^{M-1}\left[\int_0^0 \breve{R}_{XX}\left(\frac{2k\pi}{M}\right)d\omega'\right] e^{j2\pi\frac{km}{M}} \tag{3.40}$$

If the input signal does not contain periodic components at frequencies multiple of F_s/M then the power spectral density at these frequencies is bounded such that $\breve{R}_{XX}\left(\frac{2k\pi}{M}\right) < +\infty$, and the integral is zero. An opposite assumption would mean that the signal contains sine waves a frequencies multiple of F_s/M, which breaks the assumption that the signal is stationary for the first moment. Actually, making the assumption that the

signal is first-order stationary is a sufficient condition but not a necessary condition. If the signal is not stationary for the first moment but the Fourier transform of its expected value $m_X(t)$ does not contain Dirac impulses at frequencies multiple of F_s/M, the condition is also true. Under those conditions, the conclusion is that the variance of the offset estimator converges to zero when N grows infinity.

$$\text{var}\left(\hat{O}_m\right) = 0 \tag{3.41}$$

Case N finite The previous paragraph showed that the offset mismatch estimator was unbiased with variance 0 when N is infinite but what is interesting in practice is the case with a finite N. When N is finite, the frequency of the offset estimation filter $\breve{H}_{\hat{O}}$ is no longer a Kronecker delta function. The frequency response has a low pass characteristics as shown in Figure 3.6. The width of the pass band depends on the average length and it

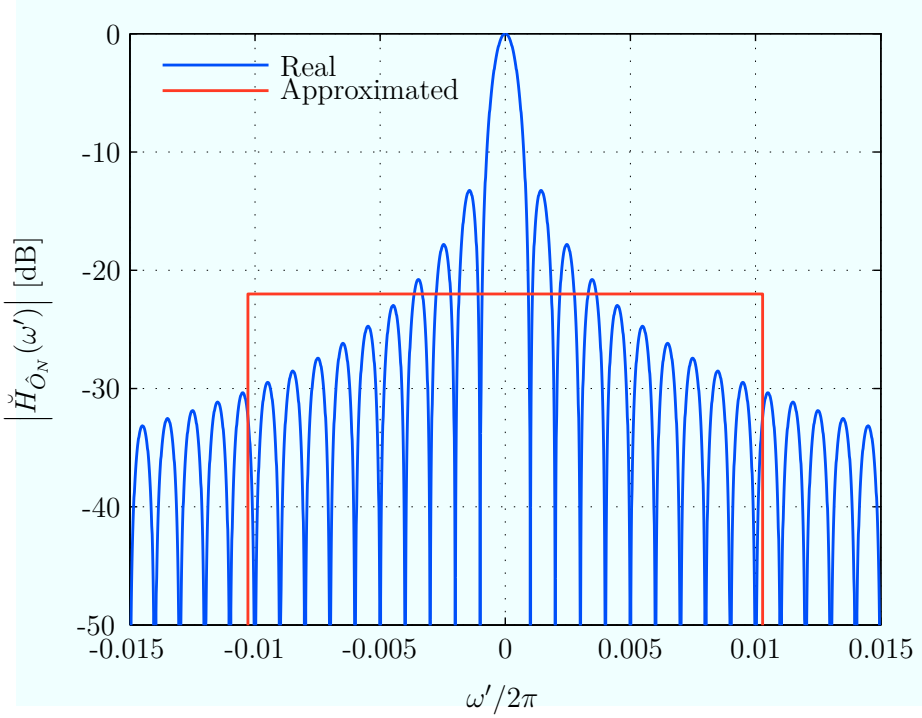


Figure 3.6 – Real and approximated frequency responses of the offset mismatch estimation filter ($N = 1000$)

can be shown that, *in case of a wide band input signal*, 99%[3]. of the output power comes from the part of the signal that lies in the band $[-\omega'_{99}, \omega'_{99}]$ where $\omega'_{99} \approx 2\pi \times \alpha_{99}/N$ (in practice $\alpha_{99} \approx 10.29$). For our analysis we consider that the square amplitude of the offset estimation filter frequency response can be approximated by the one of an ideal

3. The value of 99% is chosen quite arbitrarily to illustrate the concept.

low pass filter with cut-off angular frequency ω'_{99}.

$$\left|\breve{H}_{\hat{O}_N}(\omega')\right|^2 \approx \frac{0.99}{2\alpha_{99}} \times \text{rect}\left(\frac{\omega'}{2\omega'_{99}}\right) \tag{3.42}$$

Figure 3.6 shows this approximate response superimposed on the real frequency response.

Given this approximation, the power spectral density at the output of the offset estimation filter is

$$\begin{aligned}\breve{R}^m_{\hat{O}\hat{O}}(\omega') =& \breve{R}^m_{\tilde{X}\tilde{X}}(\omega')\left|\breve{H}_{\hat{O}_N}(\omega')\right|^2 \\ =& \frac{0.99}{2\alpha_{99}}\frac{1}{M}\sum_{k=0}^{M-1}\breve{R}_{XX}\left(\frac{\omega'}{M}-\frac{2k\pi}{M}\right)e^{-jm\left(\frac{\omega'}{M}-k\frac{2\pi}{M}\right)}\,\text{rect}\left(\frac{\omega'}{2\omega'_{99}}\right)\end{aligned} \tag{3.43}$$

Integrating the above spectral density expression over the entire Nyquist band leads to the variance of the offset estimator:

$$\text{var}\left(\hat{O}_m\right) = \frac{1}{2\pi}\frac{0.99}{2\alpha_{99}}\frac{1}{M}\sum_{k=0}^{M-1}\int_{-\omega'_{99}}^{\omega'_{99}}\breve{R}_{XX}\left(\frac{\omega'}{M}-\frac{2k\pi}{M}\right)d\omega' \tag{3.44}$$

Unlike the infinite average case, the variance of the estimator contains a non-zero term that makes the output of the offset estimator oscillate around its average o_m. The variance of the offset estimator now depends on the spectral properties of the input signal.

The best case (lowest variance) is when no signal is present in the frequency bands of width $2\omega'_{99}$ around the angular frequencies multiple of F_s/M. In that case, only the noise is integrated M times between $-\omega'_{99}$ and ω'_{99}. For example, if the noise is dominated by quantization noise only the power spectral density of the quantization noise is integrated. The power spectral density $R_{QQ}(\omega)$ of the quantization noise is given by

$$R_{QQ}(\omega) = \frac{\text{LSB}^2}{12} \tag{3.45}$$

Inserting the above expression of the quantization noise PSD into the offset estimator variance expression 3.44 gives after simplification:

$$\sigma^2_{\hat{O}_N} = 0.99 \times \frac{\text{LSB}^2}{12N} \tag{3.46}$$

If the specification is to have the offset mismatch noise after correction 10 times lower than the quantization noise, the variance of the estimator must be:

$$0.99 \times \frac{\text{LSB}^2}{12N} = \frac{1}{10}\frac{\text{LSB}^2}{12} \tag{3.47}$$

which corresponds to an average length $N \approx 120$. However, this requirement is too optimistic because when it comes to offset mismatch, one usually enforces limits on the SFDR rather than on the SNDR. We saw in Chapter 2 that specifying SFDR limits lead to more stringent requirements on the offset mismatch level. In a 9-bit TIADC, because of quantization noise, the maximum reachable SNDR for a full-scale sine input is 56dB. If the offset mismatch noise is specified to be a tenth of the power of the quantization noise, then the worst-case SFDR (equal to the SNDR as shown in Chapter 2) is 66dB, which is not necessarily an acceptable level. If for example we want a worst case SFDR of 86dB, the offset mismatch noise has to be 1000 times lower than the quantization noise, which yields an averaging length $N \approx 12000$.

Moreover, considering that only a relatively low power noise is present in the $2\omega'_{99}$-wide frequency bands around multiple of F_s/M yields a best case variance for the offset estimator. In real communication systems, the input signal is usually wide band and it possibly carries some non-negligible power in those frequency bands. A simple situation is to consider a signal that has a constant power spectral density over the entire Nyquist band [4]. The signal PSD can be expressed in terms of the Peak-to-Average-Power-Ratio (PAPR) (expressed in dB) of the signal, such that:

$$\breve{R}_{XX}(\omega) = 10^{-\frac{\text{PAPR}}{10}} 2^{2B-1}\text{LSB}^2 \tag{3.48}$$

Therefore, the variance of the offset estimator with a wide band input signal is

$$\sigma^2_{\hat{O}_N} = \frac{0.99}{N} \times 10^{-\frac{\text{PAPR}}{10}} 2^{2B-1}\text{LSB}^2 \tag{3.49}$$

Using the example of the 9-bit TIADC with a 86 dB worst case SFDR requirement and assuming that the signal PAPR is 15 dB [5], leads to the following condition on the average length:

$$\frac{0.99}{N} \times 10^{-\frac{\text{PAPR}}{10}} 2^{2B-1}\text{LSB}^2 = \frac{1}{1000}\frac{\text{LSB}^2}{12} \Rightarrow N \approx 50 \text{ millions} \tag{3.50}$$

To put things in perspective, the above averaging length requirement corresponds to an estimation time of 0.25 second if the sub-ADCs are sampling at a rate of 200 MS/s.

4. It is not a realistic case because communication signal are usually comprised of several narrow band channels centered at different carrier frequencies. The approximation of a full band signal can nonetheless give a good idea of the required averaging length.

5. This is a reasonable value for communication signals.

3.4.4 Conclusion

This section presents the concept of the offset mismatch calibration technique. It converges for first-order stationary signals but it can be applied on non-stationary signals if the time-dependent expected value of the signal does not contain spurious tones at frequencies multiples of F_s/M. After the offset mismatches have been calibrated, the mean of each sub-ADC's output signal becomes very close to zero, which is a requirement for the gain mismatch calibration stage.

3.5 Gain mismatch calibration

3.5.1 Introduction

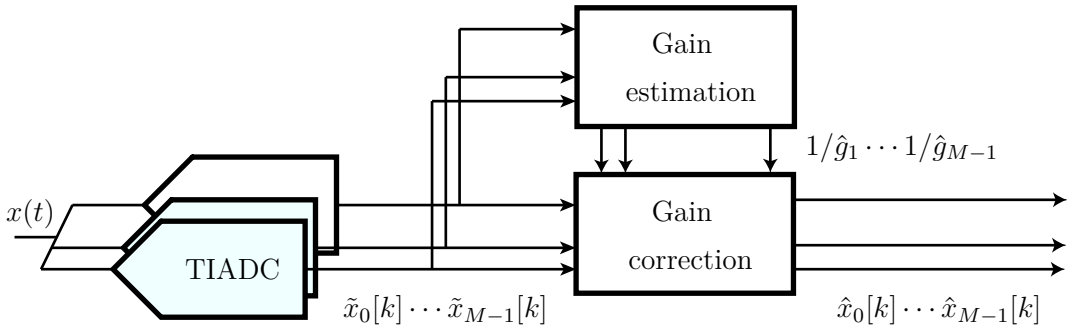


Figure 3.7 – Gain mismatch calibration block diagram

This section describes the principle of the proposed background gain mismatch calibration technique and analyzes its performance. The other mismatches are assumed to be zero.

The purpose of the gain mismatch calibration is to

— estimate the relative gains of the sub-ADCs with respect to a reference sub-ADC (sub-ADC 0)

— correct each sub-ADC's output signal once the estimated gains are known

Whereas the offset mismatch calibration algorithm tries to equalize the averages of the sub-ADC output signals, the gain mismatch calibration algorithm tries to equalize the sampled variance of the sub-ADC output signals. The block diagram of the gain mismatch calibration is given in Figure 3.7. As explained in section 3.5.3, the relative gain of one sub-ADC with respect to a reference sub-ADC is estimated by taking the ratio between the sub-ADC output power and the output power of the reference sub-ADC. The gain mismatch correction stage multiplies each sub-ADC's output signal by the inverse of the corresponding power ratio. The correction stage is explained in section 3.5.2.

3.5.2 Gain mismatch correction

3.5.2.1 Principle

If we know estimates $\{\hat{g}_m, m \in \mathcal{M}^*\}$ of the gains of the sub-ADCs with respect to the reference sub-ADC (sub-ADC 0), gain mismatch correction can be done by dividing each sub-ADC's output signal by its respective gain estimate as illustrated in Figure 3.8. Sub-ADC 0 is the reference ADC and therefore its gain is left unchanged (gain of 1).

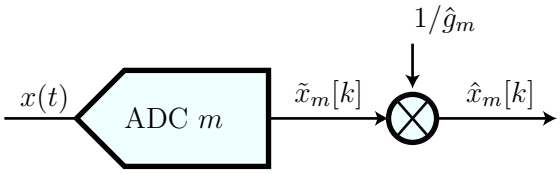


Figure 3.8 – Gain mismatch correction block diagram

After the gain mismatch correction, the sub-ADC output signals are

$$\hat{x}_m[k] = \frac{\tilde{x}_m[k]}{\hat{g}_m} = \tilde{g}_m x_m[k] \text{ for } m \in \mathcal{M}^* \tag{3.51}$$

where $\{\tilde{g}_m = g_m/\hat{g}_m, m \in \mathcal{M}^*\}$ are the residual gains after correction. Naturally, if the gain mismatch estimation is perfect $\hat{g}_m = g_m$ and the original signal is perfectly recovered (assuming no quantization):

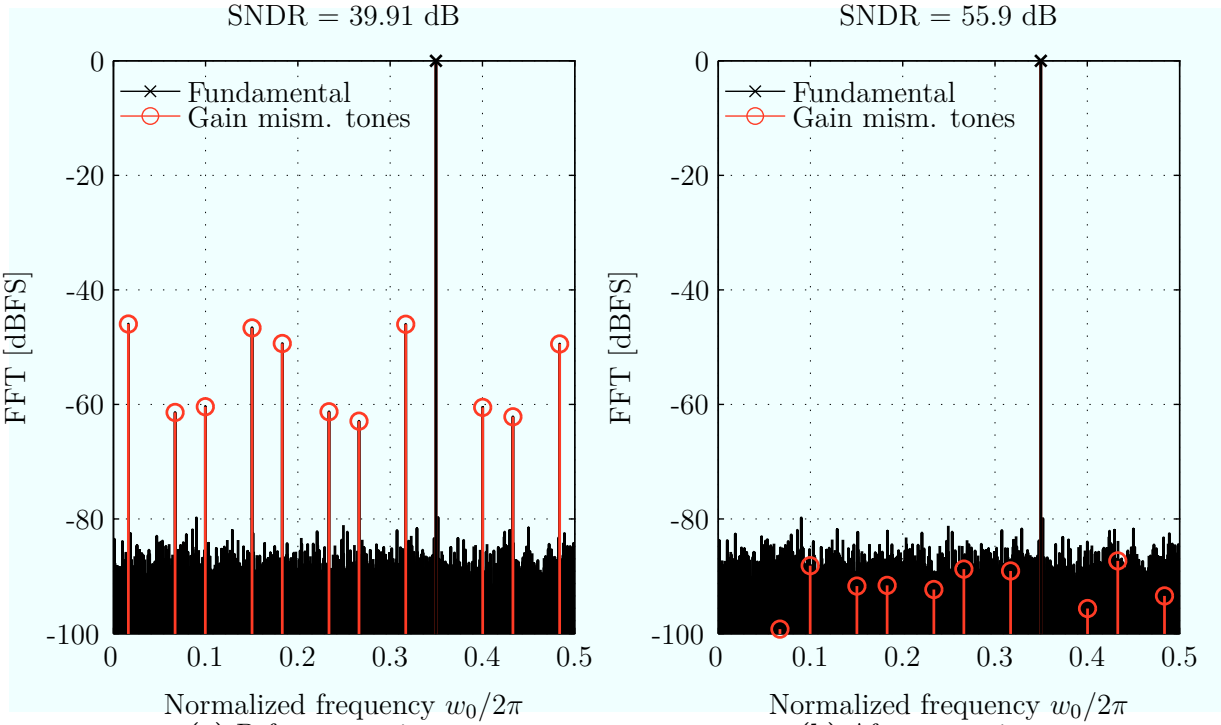
$$\hat{x}_m[k] = x_m[k] \text{ for } m \in \mathcal{M} \tag{3.52}$$

The gain mismatch correction effect on the TIADC output signal is demonstrated in Figure 3.9 where the TIADC output spectrums before and after gain mismatch correction are shown.

3.5.2.2 Circuit implementation

The architecture of the gain mismatch correction is illustrated in Figure 3.8. The required amount of hardware is relatively low because the correction only uses $M - 1$ multipliers (one per sub-ADC except sub-ADC 0 that serves as the reference).

Dividing each sub-ADC's output signal by its respective gain is done at the rate F_s/M of one sub-ADC but because implementing dividers at a high clock frequency is resource hungry, designers prefer to replace them by multipliers that have a lower hardware cost. In other words, it is preferable to multiply each sub-ADC's output signal by the inverse



(a) Before correction **(b)** After correction

Figure 3.9 – Output spectrum before and after gain mismatch correction ($B = 9$, $M = 12$, $N_{\text{FFT}} = 12 \times 4096$, $\bar{\sigma}_g = 0.01$, ideal gain values used for correction)

of its gain estimate rather than dividing by the gain estimate itself. As it will be later be explained in Section 3.5.3, the gain estimation unit can easily be implemented such that it provides the inverse of the gain estimates rather than the gain estimates themselves.

3.5.3 Gain mismatch estimation

3.5.3.1 General idea

The general idea of gain mismatch estimation relies on a simple principle. When there are no gain mismatches, each sub-ADC's output power is equal to the input signal power. With gain mismatches, each sub-ADC's output power is different. Each sub-ADC's output power is equal to the input signal power multiplied by the squared gain of the sub-ADC. Estimating the relative gain mismatches is easily done by measuring the ratio between each sub-ADC's output power and the reference sub-ADC output power, as shown in Figure 3.10.

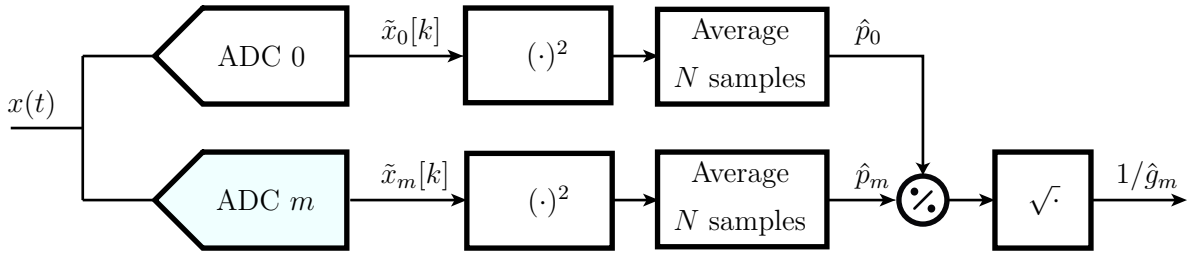


Figure 3.10 – Gain mismatch estimation block diagram

3.5.3.2 Theoretical explanation – WSS process

The gain mismatch estimation principle is easy to understand but it requires that the signal verifies certain statistical properties. The mathematical proof is first done assuming that the input signal is a realization of a WSS process with zero mean, and then extended to the slightly more complex case where the signal is a realization of a non-stationary process.

As explained in Section 3.4, the autocovariance function of a WSS process is time independent. In particular, the variance of the process is also time-independent. When there are no gain mismatches, each sub-ADC's output signal $x_m[k]$ is a realization of a sub-process $\{X_k^m, m \in \mathcal{M}\}$ that has the same variance as the input random process:

$$\text{var}\left(X_k^m\right) = \text{var}\left(X_n\right) = \text{var}\left(X_t\right) = r_{XX}[0] \text{ for } m \in \mathcal{M} \tag{3.53}$$

With gain mismatches, the above property is no longer verified, and each sub-ADC's output signal $\{\tilde{x}_m[k], m \in \mathcal{M}\}$ is a realization of a scaled version $\{\tilde{X}_k^m, m \in \mathcal{M}\}$ of the random sub-process $\{X_k^m, m \in \mathcal{M}\}$:

$$\text{var}\left(\tilde{X}_k^m\right) = g_m^2 \,\text{var}\left(X_k^m\right) = g_m^2 r_{XX}[0] \tag{3.54}$$

where $\{g_m, m \in \mathcal{M}\}$ are the gains of the sub-ADCs. Since we are only interested in the relative gains between the sub-ADCs, we assume that $g_0 = 1$ without loss of generality. It follows that each gain g_m can be obtained as the square root of the ratio between the output variance of sub-ADC m and the output variance of sub-ADC 0

$$g_m = \sqrt{\frac{\text{var}\left(\tilde{X}_k^m\right)}{\text{var}\left(\tilde{X}_k^0\right)}} \tag{3.55}$$

A natural way to obtain estimates $\{\hat{g}_m, m \in \mathcal{M}^*\}$ of the gains $\{g_m, m \in \mathcal{M}^*\}$ is to replace the variance terms in the above expression 3.55 by their estimates. Since the signal is assumed to have zero mean, the variance can be estimated by using the sampled mean square. Replacing the variance by the sampled mean square leads to the the gain mismatch estimation formula:

$$\hat{g}_m = \sqrt{\frac{\frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_m[k]^2}{\frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_0[k]^2}} \tag{3.56}$$

One can define the estimator $\hat{P}_m$ associated to the mean square of each sub-ADC's output signal:

$$\hat{P}_m = \frac{1}{N}\sum_{k=0}^{N-1} \left(\tilde{X}_k^m\right)^2 \tag{3.57}$$

For a process with zero expected value, the mean square is an unbiased estimator of the variance because

$$\mathrm{E}\left(\hat{P}_m\right) = \mathrm{var}\left(\tilde{X}_k^m\right) \tag{3.58}$$

If the random input process is assumed ergodic for the second moment, the estimates $\{\hat{p}_m, m \in \mathcal{M}\}$, provided by the variance estimators $\{\hat{P}_m, m \in \mathcal{M}\}$, converge toward their expected value when the average length N becomes infinite:

$$\lim_{N \to +\infty} \hat{p}_m = g_m^2 r_{XX}[0] \tag{3.59}$$

3.5.3.3 Theoretical explanation – non-stationary process

Section 3.2 of this document that communication signals are more accurately described as realizations of non-stationary random processes. As we saw, an importance consequence is that the autocovariance of the signal, and by extension the variance of the signal, is time-dependent. Nonetheless, the gain mismatch estimator described in the previous section remains valid provided that certain conditions on the autocovariance of the signal are met.

To clarify this affirmation, let us define the M random processes $\{Y_k^m, m \in \mathcal{M}\}$ verifying

$$Y_k^m = \left(\tilde{X}_k^m\right)^2 - g_m^2 r_{XX}((kM+m)T_s, 0) \tag{3.60}$$

These processes are first-order stationary because their expected value does not depend on the time:

$$\mathrm{E}\left(Y_k^m\right) = \mathrm{E}\left((X_k^m)^2\right) - g_m^2 r_{XX}[kM+m, 0] = 0 \tag{3.61}$$

As a consequence, the expected value of the estimator $\hat{P}'_m = \frac{1}{N}\sum_{k=0}^{N-1} Y_k^m$ is equal to zero.

Since the associated estimates $\hat{p}'_m = \frac{1}{N}\sum_{k=0}^{N-1}\left[\tilde{x}_m[k]^2 - r_{XX}\left[kM+m,0\right]\right]$ tend to their expected value when the averaging length becomes infinite, we obtain

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{k=0}^{N-1}\left(\tilde{x}_m[k]^2 - g_m^2 r_{XX}\left[kM+m,0\right]\right) = 0 \tag{3.62}$$

Assuming that $\lim_{N\to+\infty}\sum_{k=0}^{N-1} r_{XX}\left[kM+m,0\right]$ exists, the limit of the sum can be written as the sum of the limits, which implies that:

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{k=0}^{N-1}\tilde{x}_m[k]^2 = g_m^2\lim_{N\to+\infty}\frac{1}{N}\sum_{k=0}^{N-1} r_{XX}\left((kM+m)T_s,0\right) \tag{3.63}$$

In order to find an expression that is similar to that of Equation 3.59, it is necessary to demonstrate that the limit $\lim_{N\to+\infty}\frac{1}{N}\sum_{k=0}^{N-1} r_{XX}\left((kM+m)T_s,0\right)$ is the same for each sub-ADC, or in other words that the value of the limit is independent of m.

Interestingly, this problem is very similar the offset estimation case where we show that the average of each channel converges to zero. Indeed, averaging $r_{XX}\left[kM+m,0\right]$ is similar to averaging $\tilde{x}_m[k]$ in Equation 3.30. Once again this problem is more easily tackled in the frequency domain. First, we can express the continuous autocovariance function $r_{XX}(t,\tau)$ in terms of its Fourier transform along the time variable t:

The term $r_{XX}\left[kM+m,0\right]$ is a sampled version of $r_{XX}\left(t,0\right)$ and in the frequency domain becomes:

$$\breve{R}_{XX}^{0,m}(\upsilon') = \frac{1}{M}\sum_{l=0}^{M-1}\breve{R}_{XX}^{0}\left(\frac{\upsilon'}{M} - l\frac{2\pi}{M}\right)e^{-j\left(\frac{\upsilon'}{M}-l\frac{2\pi}{M}\right)m} \tag{3.64}$$

where $\upsilon' = M\Upsilon T_s$. When the average length becomes infinite, the right-hand side limit in Equation 3.63 can be interpreted in the frequency domain associated to the time variable t as the product of the DTFT in Equation 3.64 with the Kronecker delta function $\delta[\upsilon']$:

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{k=0}^{N-1} r_{XX}\left((kM+m)T_s,0\right) = \mathcal{F}^{-1}(\breve{R}_{XX}^{0,m}(\upsilon')\delta[\upsilon']) \tag{3.65}$$

The result is a sum of M terms corresponding to the F_s/M frequency components of the spectrum of $r_{XX}(t,0)$.

$$\breve{R}_{XX}^{0,m}(\upsilon')\delta[\upsilon'] = \frac{1}{M}\sum_{l=0}^{M-1}\breve{R}_{XX}^{0}\left(-2\pi\frac{l}{M}\right)e^{j2\pi\frac{lm}{M}}\delta[\upsilon'] \tag{3.66}$$

This expression is independent of the sub-ADC m if and only if the autocovariance function of the input random process $\{X_t\}$ does not contain non-stationary components

at multiples of F_s/M. This observation translates mathematically into:

$$\breve{R}_{XX}^{*0}\left(2\pi\frac{l}{M}\right) = 0 \text{ for } l \neq 0 \tag{3.67}$$

If the above condition is verified the power estimates $\hat{p}_m$ converge toward the DC value, i.e. the stationary component, of the autocovariance function:

$$\lim_{N\to+\infty} \hat{p}_m = g_m^2 \lim_{N\to+\infty} \frac{1}{N}\sum_{k=0}^{N-1} r_{XX}[k,0] = g_m^2 \bar{r}_{XX}[0] \tag{3.68}$$

This generalizes the situation where the input random process is WSS. When the input random process is WSS, all the non-stationary frequency components of the autocovariance function are zeros by definition (not only the ones that are multiple of F_s/M). The constraint for non-stationary signals is weaker. Only the frequency components of the autocovariance function at multiple of F_s/M have to be zero.

Whereas wideband communication signals are in general not WSS, they usually verify the above weaker condition. This convergence condition is however broken if the cyclostationarity period of the input signal is an integer fraction of MT_s because in that case the Fourier decomposition of $R_{XX}(t,0)$ has a strong harmonic tone at this frequency. In practice, this situation is unlikely. We saw in section 3.2 that the cyclostationarity period of typical communication signals is often of the order of the second, as opposed to the sampling period which is of the order of the nanosecond.

In general, if the input signal is cyclostationarity with period T its autocovariance can be decomposed with its Fourier series, which shows that $R_{XX}^{\tau}(\Upsilon)$ has harmonic tones at frequencies multiple of $1/T$. If the k-th harmonic falls on a frequency multiple of F_s/M, the above convergence condition is theoretically broken. However, in practice, k is very big [6], and consequently, the associated harmonic tone is very weak. Indeed, it is reasonable to assume that that $r_{XX}(t,\tau)$ varies smoothly with t (this is the case in the example of a multi-channel communication signal defined in Section 3.2), such that at least some of its derivatives with respect to t are defined. If we consider that derivatives are defined up to the n-th order, then the Fourier coefficients (the power of the harmonics) of $r_{XX}(t,\tau)$ decrease faster than $1/k^n$ and the harmonics at frequencies multiple of F_s/M are very weak as compared to the DC value. Therefore, the negative effect of a cyclostationary signal with large cyclostationary period on the gain estimation is minimal.

6. At least of the order of 1 billion if $T = 1$ s and $F_s = 1$ GS/s

3.5.3.4 Circuit implementation

The implementation of the gain mismatch estimation algorithm requires $M-1$ averaging units (one adder and one register per unit), $M-1$ squaring units (one multiplier per unit) and $M-1$ dividers. From a practical standpoint, the gain mismatch estimation unit is designed to provides the inverses of the gain estimates to the mismatch correction stage. It just means that in practice the inverse of the gain mismatch estimation formula is computed (Equation 3.56).

One can also reduce the hardware overhead by replacing the squaring operation by an absolute value operation, which suppresses one multiplier per sub-ADC (see Figure 3.11). It also avoids calculating the square root of $1/\hat{g}_m^2$. The ratio between the averages of the absolute values gives the value of the gain estimate directly. This implementation trick slightly decreases the estimation accuracy as shown in the next section.

The division is done in several clock cycles with only a few slow multipliers after the results of the averages are available.

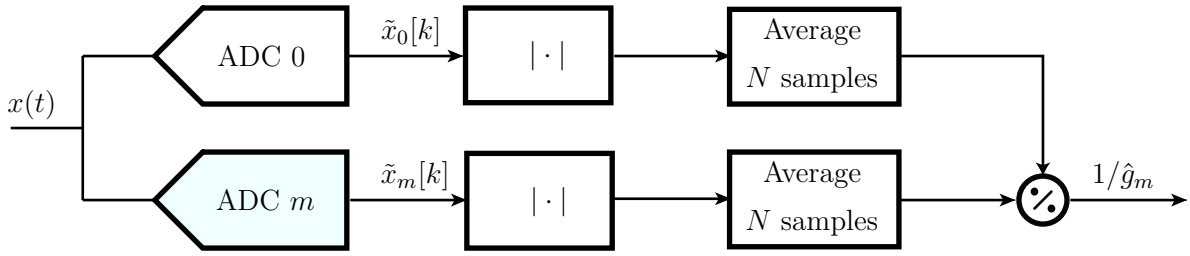


Figure 3.11 – Alternative implementation of the gain mismatch estimation

3.5.3.5 Accuracy analysis

In order to determine the required average length N, we need to study the behavior of the expected value and the variance of the gain mismatch estimator $\hat{G}_m$ as a function of N, where

$$\hat{G}_m = \sqrt{\frac{\hat{P}_m}{\hat{P}_0}} \tag{3.69}$$

Assuming an ideal gain mismatch correction (given a set of gain estimates), we want to know how big N needs to be such that the residual gains after correction are small enough to reach the wanted SNDR. We know from Equation 2.61 in Chapter 2, that the SNDR after correction is a function of the RMS value $\sigma_{\tilde{g}}$ of the residual gains $\{\tilde{g}_m = g_m/\hat{g}_m, m \in \mathcal{M}\}$. Therefore, it is natural to calculate the variance of the random variables

$\{\tilde{G}_m, m \in \mathcal{M}\}$ associated to those residual gains:

$$\text{var}\left(\tilde{G}_m\right) = \text{var}\left(\frac{g_m}{\hat{G}_m}\right) \tag{3.70}$$

Unfortunately, deriving a theoretical expression of the variance in the above equation is complex because it requires information about the 4th moment of the input signal, which in general is not a known quantity. Making the same type of analysis as we did with offset mismatch estimation is difficult here.

In the – non realistic – case where the signal is White Gaussian Noise (WGN), numerical simulations show that the residual gain variance is inversely proportional to N:

$$\text{var}\left(\tilde{G}_m\right) = \frac{1}{2N} \tag{3.71}$$

To evaluate the accuracy of the gain mismatch estimation algorithm, we can calculate the SNDR at the TIADC output with a sinewave at the input. The plots in Figure 3.13 show the SNDR as a function of the average length when the residual gain mismatches are the only source of noise. We observe that the theoretical formula for the WGN case matches numerical Monte-Carlo simulations. For example, 300 000 samples per sub-ADC are needed in order to achieve a SNDR of 56 dB, which is a level equivalent to the level of the quantization noise of 9-bit ADC.

However, a WGN input signal is not a realistic input signal, and it is legitimate to wonder how valid this expression of the variance is when the output samples of the sub-ADCs are not independent, as it is the case in a real communication signal. If the signal is generated from a list of random symbols that are passed through a pulse shaping filter and then modulated by a sine wave, the signal is correlated with itself on a duration t_c that depends on the length $K \times T_{\text{symb}}$ of the pulse shaping filter, where K is typically equal to 40.

When expressed as a function of the sampling period T_s, the duration during which the correlation is not zero depends on the oversampling ratio α and the pulse duration $t_c = K\alpha T_s$. For instance, the oversampling ratio for a 6 MHz wide carrier sampled at 1.6 GS/s is $\alpha = 266$. As a consequence, a sub-ADC output signal is correlated with itself on a length $L_c = K\alpha/M$. Another way to look at it is to consider that each sub-ADC's output signal is made of N/L_c subsets of uncorrelated samples on which the above theory can be applied. The worst-case scenario is to assume that the subsets are identical, i.e. perfectly correlated. In that case, averaging over $N \times L_c$ samples gives the same accuracy as averaging over N samples and a factor L_c more samples is required to get the estimation precision obtained for N uncorrelated samples.

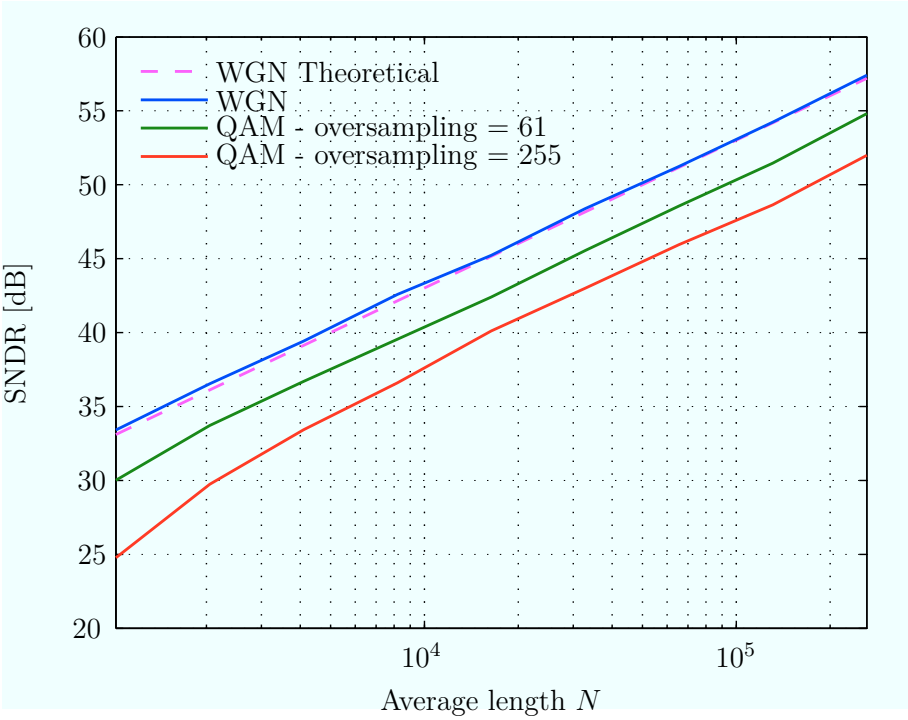


Figure 3.12 – Gain mismatch estimation accuracy as a function of average length, and for different types of modulated input signals ($M = 12$, $\sigma_g = 1\%$, residual gains averaged over 100 Monte-Carlo simulations, no quantization)

If we consider a 12-channel TIADC with an oversampling ratio $\alpha = 266$ and a filter length $K = 40$, the correlation length is $L_c = 266 \times 40/12 \approx 887$. In theory, at least 887 times more samples would be required to get the same estimation accuracy as with the uncorrelated case, i.e. around 9 billions samples per sub-ADC. In reality, this is an upper bound that is not reached because the L_c subsets of samples are not identical and each sub-set gives additional statistical information about the signal.

The plots in Figure 3.13 demonstrate this effect. As predicted, the gain mismatch estimation error becomes bigger when the correlation length increases. However, the upper bound that was mentioned above is far from being reached. A SNDR of 43 dB is reached for an average length $N = 10000$ if the input is WGN. If the input is a modulated signal with an oversampling ratio of 265 as in the example above, an average length $N = 30000$ is required to reach the same SNDR. It is only 3 times as many samples, which is rather far from the 887 factor mentioned before. The reason is that, in practice, the signal autocorrelation function is not equal to 1 for samples that are distant from each other (as it would be if the samples were perfectly correlated) but decreases when the distance between samples increases.

Figure 3.13 shows the gain mismatch estimation accuracy degradation that occurs when the ratio of the average absolute values is computed instead of the ratio of the

average squared values. It proves that using the absolute value incurs a minimal accuracy loss while saving hardware cost.

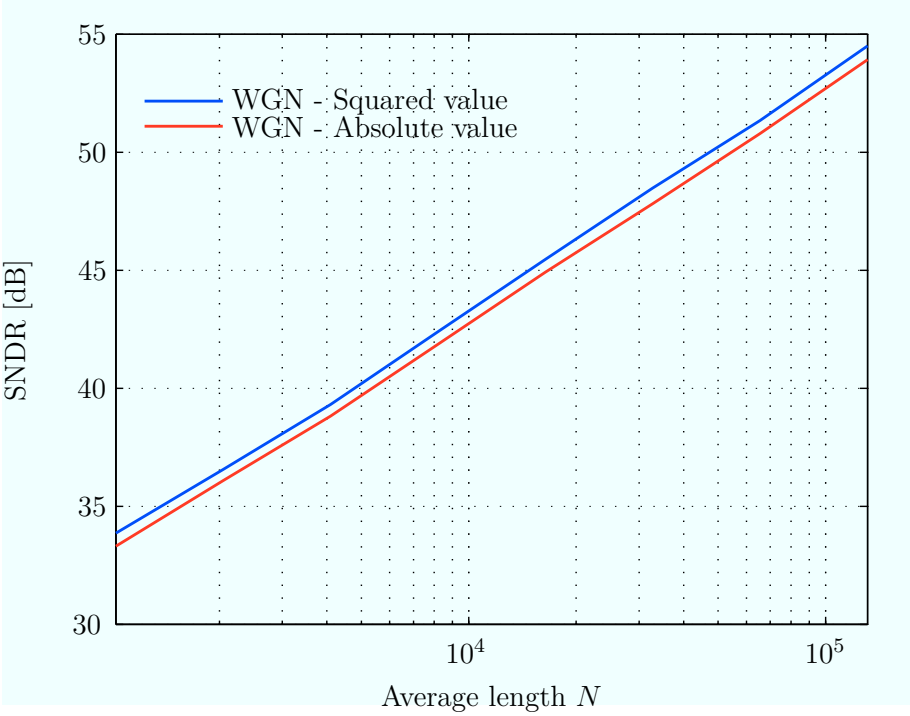


Figure 3.13 – Gain mismatch estimation accuracy comparison between using averaged absolute value or averaged squared value ($M = 12$, $\sigma_g = 1\%$, residual gains averaged over 100 Monte-Carlo simulations, no quantization)

3.5.4 Conclusion

We showed a background gain mismatch calibration technique that equalizes the sub-ADC output average powers. The calibration can successfully be run on input signals that are realizations of non-stationary processes as long as the time-dependent autocovariance of the process does not contain spurious tones at frequencies multiple of fs/M.

After gain mismatch calibration, the sub-ADC output signals have the same average power, which is a requirement for the skew mismatch calibration stage that is performed downstream.

3.6 Skew mismatch calibration

3.6.1 Introduction

This section describes an innovative skew mismatch calibration technique that performs a direct estimation of the sub-ADCs' timing offsets. This approach stands out of

the state of the art that often relies on adaptive estimation techniques such as gradient descent.

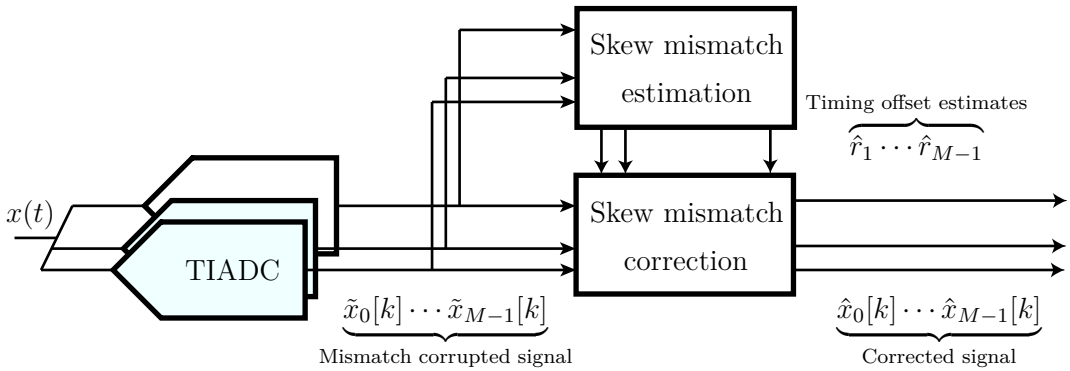


Figure 3.14 – Skew mismatch calibration block diagram

The calibration is done in two phases (see Figure 3.14) that consist in

— estimating each sub-ADC's timing offset with respect to a reference sub-ADC

— correcting each sub-ADC's output signal based on the result of the estimation

We assume that the timing offsets are small, which allows linearizing the delay transfer function.

The correction of each sub-ADC's output signal is done through linear interpolation based on the timing offset estimates and the signal derivative.

The timing mismatch estimation provides timing offset estimates after a single iteration with two distinct phases:

— First phase: calculation of the covariance between each sub-ADC's output signal and each sub-ADC's output signal derivative

— Second phase: calculation of the timing offset estimates by linearly post-processing the covariances obtained during the first phase

3.6.2 Skew mismatch correction

3.6.2.1 General idea

The Figure 3.15 shows how a sub-ADC sample is affected by a sampling error. If the delay is sufficiently small, the original sample can be recovered from the delayed sample by following the tangent of the signal around the sample to be corrected (linear interpolation).

In other words, the original sample can be recovered from the delayed sample by subtracting from it an error term proportional to the derivative of the signal and the

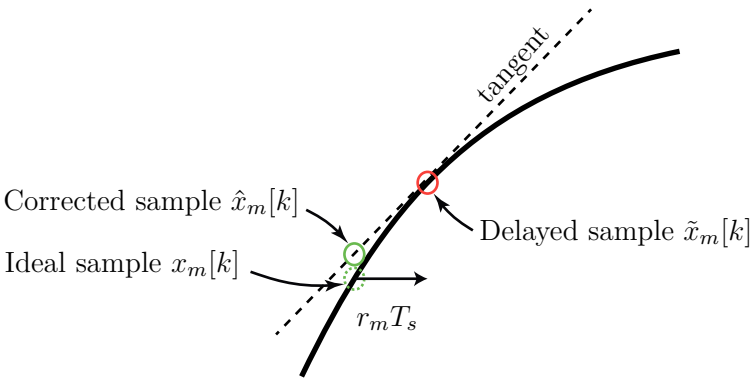


Figure 3.15 – Skew mismatch correction illustration

timing offset. This is illustrated in Figure 3.16 that shows the block diagram of the skew mismatch correction unit.

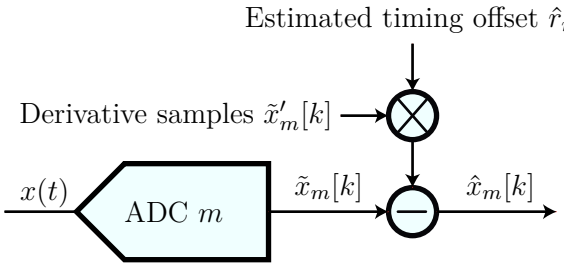


Figure 3.16 – Skew mismatch correction block diagram

3.6.2.2 Theoretical explanation

Assuming that timing offset estimates $\{\hat{r}_m, m \in \mathcal{M}\}$ of the real timing offsets $\{r_m, m \in \mathcal{M}\}$ are provided, one can recover the original signal through a first-order Taylor approximation, by using the fact that the timing offsets are small as compared to the sampling period. We showed in Chapter 2 that each sub-ADC's output signal $\{\tilde{x}_m[k], m \in \mathcal{M}\}$ is the sum of an ideal term and an error term:

$$\tilde{x}_m[k] \approx x_m[k] + r_m T_s x'_m[k] \tag{3.72}$$

where $x'_m[k]$ denotes the derivative of the input signal $x(t)$ taken at the instant $(kM + m)T_s$. It naturally follows that each sub-ADC's output signal $\{x_m[k], m \in \mathcal{M}\}$ can be recovered from its slightly-delayed version $\{\tilde{x}_m[k], m \in \mathcal{M}\}$ by subtracting the error term:

$$x_m[k] \approx \tilde{x}_m[k] - r_m T_s x'_m[k] \tag{3.73}$$

Obviously, this reconstruction formula requires some knowledge about the signal derivative. If the TIADC output signal is bandlimited to the first Nyquist zone, its derivative is obtained by filtering the TIADC output signal $x[n]$ with a derivative FIR filter with frequency response $F_d(e^{j\omega}) = j\omega$, which well-known impulse response is

$$f_d[n] = \begin{cases} 0 & \text{if } n = 0 \\ (-1)^n/n & \forall n \in \mathbb{Z}^* \end{cases} \tag{3.74}$$

The derivative terms associated to the sub-ADCs are therefore:

$$x'_m[k] = \frac{1}{T_s} x * h_d[kM + m] \tag{3.75}$$

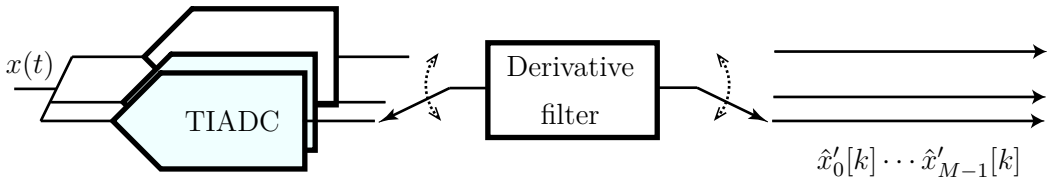


Figure 3.17 – Calculation of TIADC output signal derivative

In practice, the derivative cannot be calculated from the ideal TIADC output signal $x[n]$ because one only has access to the signal $\tilde{x}[n]$ containing timing mismatch errors. The best possible derivative estimate $\hat{x}'_m[k]$ is obtained by filtering the non-ideal signal with the derivative filter $f_d[n]$ as shown in Figure 3.17.

$$\hat{x}'_m[k] = \frac{1}{T_s} \tilde{x} * h_d[kM + m] \tag{3.76}$$

Therefore, in practice, each sub-ADC's output signal $\hat{x}_m[k]$ is recovered using the following reconstruction formula:

$$\hat{x}_m[k] = \tilde{x}_m[k] - \hat{r}_m \tilde{x} * h_d[kM + m] \tag{3.77}$$

In the above formula, both the timing offsets and the signal derivative have been replaced by their estimates.

3.6.2.3 Circuit implementation

The implementation of the derivative filter cannot use the infinite number of coefficients given in Equation **??**. If the input signal lies in the first Nyquist zone, the easiest way to implement the filter is to use a digital FIR structure with a finite number of taps

$2K_d + 1$:

$$\tilde{f}_d[n] = \begin{cases} 0 & \text{if } n = 0 \\ (-1)^n/n & \forall |n| < K_d \end{cases} \tag{3.78}$$

This filter architecture, shown in Figure 3.18, does not require general-purpose multipliers because the coefficients are fixed. Furthermore, the multiplications corresponding to the antisymmetric coefficients are grouped in pairs in order to reduce the number of fixed multipliers.

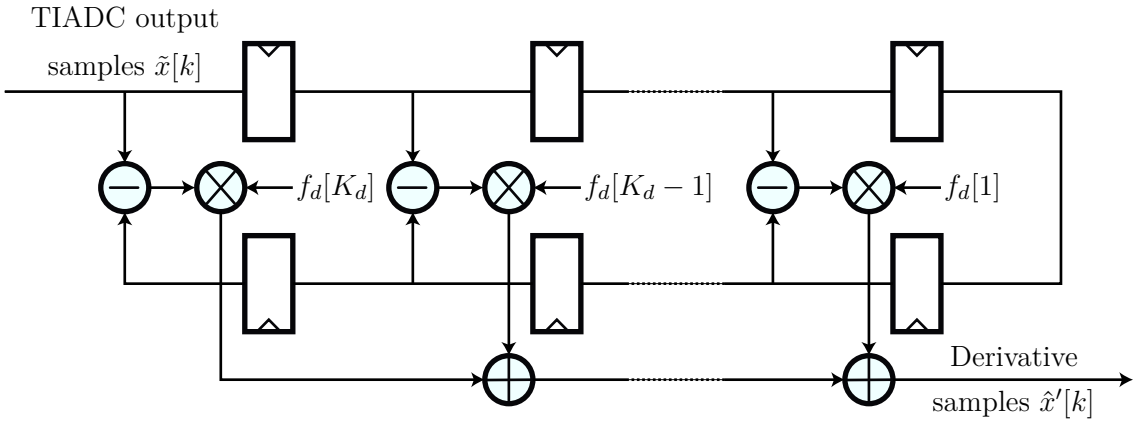


Figure 3.18 – Efficient derivative filter implementation

Naturally, the frequency response of the filter gets closer to the theoretical frequency response at the expense of more hardware by increasing the number of coefficients. There are several ways of limiting the number of coefficients of the filter. A simple truncation of the number of coefficients does not lead to a very optimal frequency response. Truncation is equivalent to the multiplication of the filter coefficients by a rectangular window, which is known to have big side lobes in the frequency domain. Other windows offer better frequency properties. The Blackman window enables the lowest side lobes and gives the best in-band accuracy. Figure 3.19 illustrates this property by showing the frequency response of a derivative filter with and without Blackman windowing. The frequency response of the windowed derivative filter is close to the ideal frequency response only up to a certain cut-off frequency ω_d. This cut-off frequency can be defined as the frequency at which the ratio between the ideal frequency response $F_d\left(e^{j\omega} = j\omega\right)$ and the frequency response $\tilde{F}_d\left(e^{j\omega}\right)$ of the truncated windowed filter falls under a certain threshold value ϵ_d:

$$\omega_d = \max\left\{\omega \in [0, \pi], \frac{\left|\tilde{F}_d(e^{j\omega})\right|}{w} < 1 - \epsilon_d\right\} \tag{3.79}$$

Increasing the number of filter coefficients increases the cut-off frequency. The simulation data shown in Figure 3.20 shows that the cut-off frequency is inversely proportional to

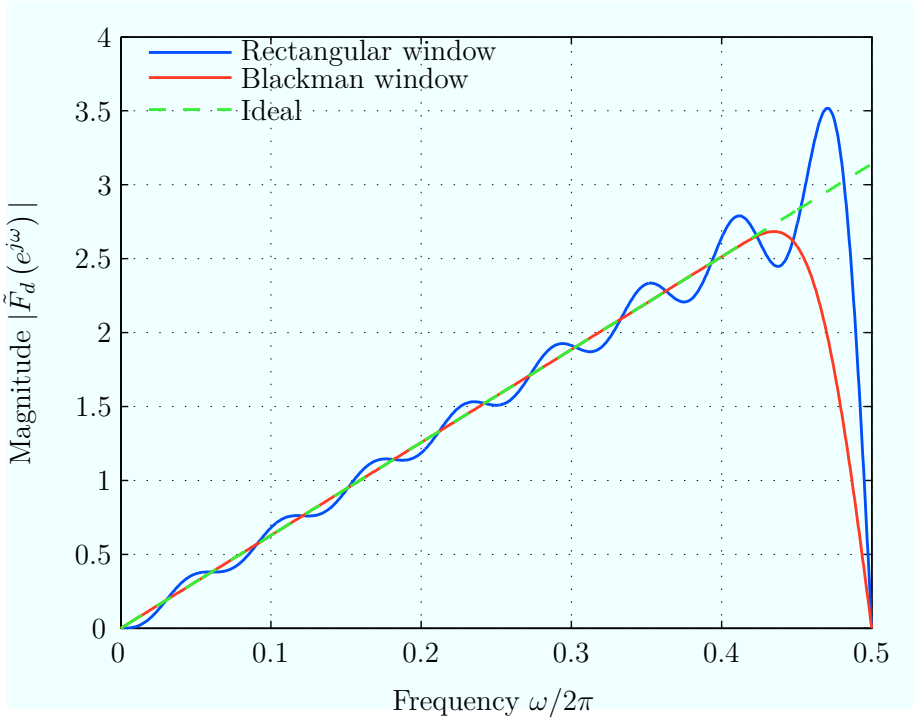


Figure 3.19 – Frequency response of the derivative filter for different windows ($K_d = 16$)

	$\epsilon_d = 0.1\%$	$\epsilon_d = 0.5\%$	$\epsilon_d = 1\%$
Value of $\alpha(\epsilon_d)$	17.18	15.41	14.49

Table 3.1 – Values of the proportionality constant $\alpha(\epsilon_d)$ for different threshold values

the number of coefficients $2K_d + 1$:

$$\omega_d = \pi - \frac{\alpha(\epsilon_d)}{2K_d + 1} \tag{3.80}$$

where $\alpha(\epsilon_d)$ is a constant that depends on the threshold value ϵ_d. The value of this constant for different threshold values is given in Table 3.1. The appropriate threshold value can be chosen through numerical simulations based on the skew mismatch correction accuracy that is needed.

In a real application, the cut-off frequency is determined based on the input signals spectrum, and the accuracy up to which the timing mismatches must be corrected. For example, a cable-TV signal has energy in the frequency band between 54 MHz and 1002 MHz. If such a signal is sampled at 2.2 GS/s, the derivative filter must be accurate up to $1.002/1.1 \approx 91\%$ of the Nyquist frequency. The yields a filter length of 61 ($K_d = 30$) if the threshold value is set to $\epsilon_d = 0.1\%$.

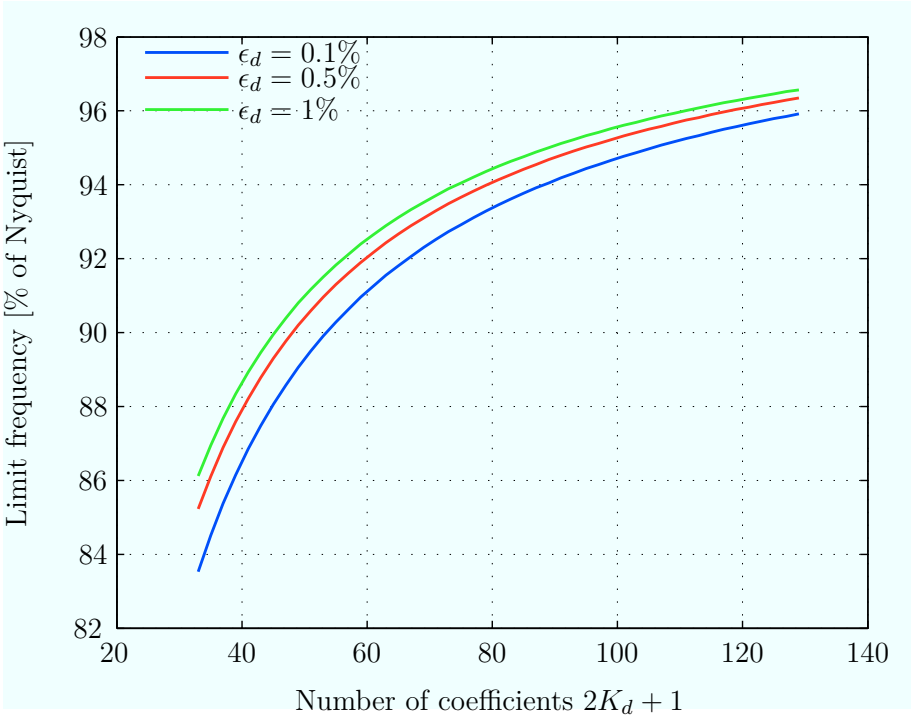


Figure 3.20 – Cut-off frequency of the derivative filter as a function of the number of coefficients, and for different levels of accuracy

3.6.2.4 Accuracy analysis

The proposed reconstruction technique is a linear approximation that only correct the timing skew errors down to a certain level. It has three types of limitations:

— the derivative-based correction assumes first-order sampling errors. This assumption does not hold when the timing offsets become large.

— the derivative filter is implemented using a limited number of coefficients, which limits the accuracy of the filter's frequency response (see previous section)

— the derivative is calculated by filtering the mismatch-corrupted TIADC output signal instead of the ideal signal

These limitations are illustrated in Figure 3.21 that compares the SNDR as a function of frequency before and after skew mismatch correction. The green dotted curve is obtained by using the theoretical expression of the signal derivative to perform the correction[7]. It demonstrates that even with a perfect derivative, the SNDR decreases with the input frequency, indicating that second-order timing mismatch errors limit the reconstruction accuracy. A similar curve is obtained when the ideal derivative is replaced by a derivative obtained by filtering the mismatch-free signal. The only difference happens at frequencies above the cut-off frequency of the derivative filter because of the limited number of

7. The input signal is of the form $x(t) = A\cos(\Omega t)$ and its derivative is $x'(t) = -A\Omega\sin(\Omega t)$

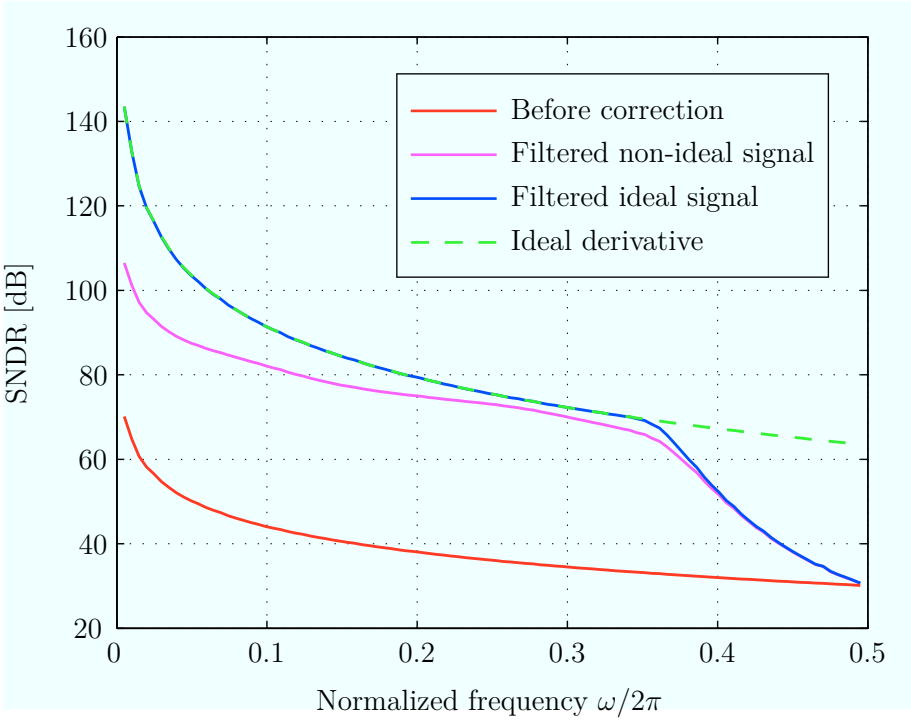
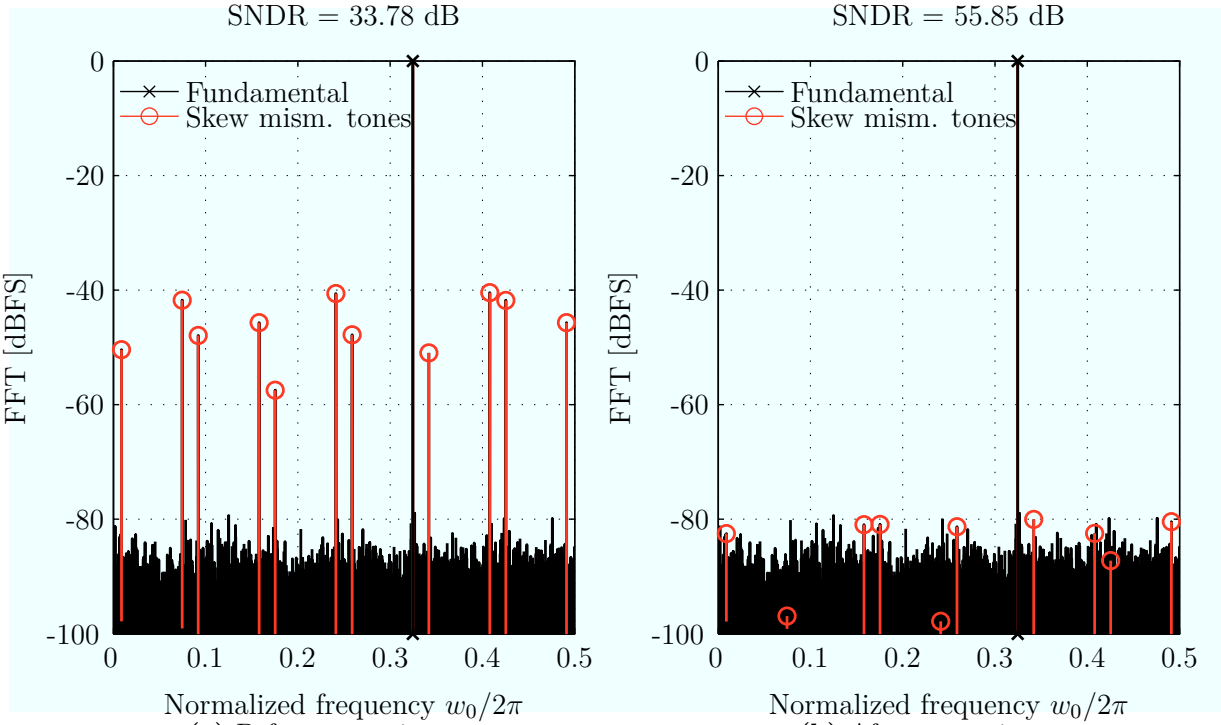


Figure 3.21 – SNDR before and after skew mismatch correction as a function of input frequency ($M = 12$, $\sigma_r = 1\%$, $K_d = 8$, no quantization, true timing offsets used as estimates)

coefficients in the filter. Finally, the magenta curve shows the performance of the skew mismatch correction as it is implemented in practice, i.e. with a derivative obtained by filtering the mismatch-corrupted TIADC output signal. These plots enable to draw a few conclusions:

- the problem of filtering the non-ideal signal is limiting the performance of the correction at low input frequency
- non-linear timing errors become non negligible when the input frequency increases
- for input frequencies above the derivative filter cut-off frequency, the finite impulse response of the filter limits the performance

Despite those limitations, the linear reconstruction method enables to drastically reduce the timing mismatch errors. The plots in Figure 3.21 show that, at 70% of Nyquist frequency, with a mismatch level of 1%, the SNDR goes from approximately 35 dB before correction to around 70dB after correction. This improvement can also be seen on the spectrums displayed in Figure 3.22. After skew mismatch correction, the SNDR is close to the maximum reachable SNDR for a 9-bit ADC (56 dB) even though spurious tones are not completely removed.



(a) Before correction **(b)** After correction

Figure 3.22 – Output spectrum before and after skew mismatch correction ($B = 9$, $M = 12$, $K_d = 8$, $N_{\text{FFT}} = 12 \times 4096$, $\sigma_r = 1\%$, true timing offsets used for correction)

3.6.3 Skew mismatch estimation

3.6.3.1 General idea

This section aims at giving an intuitive sense of the skew mismatch estimation algorithm, and we make some coarse assertions to serve this purpose. The rigorous mathematical demonstrations are provided in the next section.

Timing skew mismatches shift positively or negatively the sub-ADC sampling times, and therefore also create a small error in the sampled value. If a sub-ADC has a small positive timing offset, the error on the sampled value is positive if the slope of the signal is locally positive, and negative if the slope of the signal is locally negative. As a consequence, the average of the *delayed* samples corresponding to positive slopes of the signal will be in higher than the average of the *original* samples corresponding to positive slopes, and the average of the *delayed* samples corresponding to negative slopes will be lower than the average of the *original* samples corresponding to negative slopes. If the timing offset is small, it makes sense intuitively that the average shift is proportional to the timing offset and to the average slope. Without formalism, this can be written

$$\text{avg. of } \tilde{x}_m[k] \text{ in pos. slopes} \approx \text{avg. of } x_m[k] \text{ in pos. slopes} + \text{avg. pos. slope} \times r_m \tag{3.81}$$

$$\text{avg. of } \tilde{x}_m[k] \text{ in neg. slopes} \approx \text{avg. of } x_m[k] \text{ in neg. slopes} + \text{avg. neg. slope} \times r_m \tag{3.82}$$

If the signal is assumed to have a zero-mean, the average of the original samples corresponding to positive slopes and the average of the original samples corresponding to negative slopes are both equal to 0. Subtracting Equation 3.82 from Equation 3.81, consequently makes the two corresponding terms cancel out, yielding a right hand side that is proportional to the timing offset:

$$\begin{aligned}\text{avg. of } \tilde{x}_m[k] \text{ in pos. slopes} - \text{avg. of } \tilde{x}_m[k] \text{ in neg. slopes} \\ \approx (\text{avg. pos. slope} - \text{avg. neg. slope})\, r_m\end{aligned} \tag{3.83}$$

Of course, the slope is nothing more than the derivative of the signal, so we can rewrite Equation 3.83 as follows:

$$\frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_m[k] \times \text{sgn}(x'_m[k]) = r_m \frac{1}{N}\sum_{k=0}^{N-1} x'_m[k] \times \text{sgn}(x'_m[k]) \tag{3.84}$$

In the above equation, taking the sign of the derivative is equivalent to a 1-bit quantization of the derivative. A natural thing to do to generalize this equation is to remove the

sign function in order to get a "non quantized" version of the derivative. This yields an expression that indicates that the product average between each sub-ADC's output signal and each corresponding derivative is proportional to the timing offset and the average power of the derivative:

$$\frac{1}{N}\sum_{k=0}^{N-1}\tilde{x}_m[k]\times x'_m[k] = r_m\frac{1}{N}\sum_{k=0}^{N-1}x'_m[k]^2 \tag{3.85}$$

The timing offsets can straightforwardly be obtained from the above equation by calculating the following ratio:

$$\hat{r}_m = \frac{\sum_{k=0}^{N-1}\tilde{x}_m[k]\times x'_m[k]}{\sum_{k=0}^{N-1}x'_m[k]^2} \tag{3.86}$$

In the next section, we will demonstrate that this intuitive skew mismatch estimation formula is valid with small changes. For example, the true derivative must be replaced by its estimate, obtained by filtering the mismatch-corrupted TIADC output signal (see Section 3.6.2). We will see that this seemingly small change must be compensated to obtain accurate timing offset estimates.

3.6.3.2 Theoretical explanation – WSS process

In this section, we formally describe the skew mismatch estimation algorithm. We first assume that the input signal is a realization of random WSS process, and then extend the proofs to non-stationary random processes.

As a starting point, let us analyze the covariance between each sub-ADC's output signal and their corresponding derivative estimate. As shown in Section 3.6.2, the derivative can be estimated by filtering the TIADC output signal with the FIR filter with $2K_d+1$ coefficients $f_d[k]$. Therefore, the output of the derivative filter is itself a realization of a discrete random process $\left\{\tilde{X}'_n\right\}$:

$$\tilde{X}'_n = \sum_{l=1}^{K_d} f_d[k]\left(\tilde{X}_{n+l}-\tilde{X}_{n-l}\right) \tag{3.87}$$

Correlating the output signal of one sub-ADC to its derivative is the same as taking the expected value Γ_m of the product between the discrete random process $\left\{\tilde{X}_{kM+m}\right\} = \{X_t, t=(kM+m+r_m)T_s\}$ associated to this sub-ADC, and the random process $\tilde{X}'_{kM+m}$ associated to the derivative estimate:

$$\Gamma_m = \mathrm{E}\left(\tilde{X}_{kM+m}\tilde{X}'_{kM+m}\right) = \sum_{l=1}^{K_d} f_d[l]\left[\mathrm{E}\left(\tilde{X}_{kM+m}\tilde{X}_{kM+m+l}\right)-\mathrm{E}\left(\tilde{X}_{kM+m}\tilde{X}_{kM+m-l}\right)\right] \tag{3.88}$$

This expression can be rewritten in terms of the autocovariance function $r_{XX}(\tau)$ of the input random process:

$$\Gamma_m = \sum_{l=1}^{K_d} f_d[l] \left[r_{XX}((l + r_{m+l} - r_m)T_s) - r_{XX}((l - r_{m-l} + r_m)T_s) \right] \tag{3.89}$$

where r_{m+l} denotes the normalized timing offset associated to sub-ADC $m + l$ [8]. As mentioned before, the timing offsets are assumed to be small as compared to the sampling period, which means that $r_m \ll 1$. The autocovariance function in expression 3.89 can therefore be approximated by its 1st-order Taylor expansion:

$$\begin{aligned} \Gamma_m &= \sum_{l=1}^{K_d} f_d[l] \left(r_{XX}[l] + (r_{m+l} - r_m) r'_{XX}[l] - r_{XX}[l] - (r_m - r_{m-l}) r'_{XX}[l] \right) \\ &= -2r_m \sum_{l=1}^{K_d} f_d[l] r'_{XX}[l] + \sum_{l=1}^{K_d} f_d[l](r_{m+l} + r_{m-l}) r'_{XX}[l] \end{aligned} \tag{3.90}$$

The expression of Γ_m shows that the covariance between each sub-ADC's output signal and its estimated derivative is not only proportional to the timing offset r_m (contrary to what we thought in our intuitive analysis). The right-hand side of the equation is comprised of

— A term proportional to the timing offset r_m with a proportionality constant approximately equal to the average power of the derivative. The demonstration in Appendix **??**) indeed shows that

$$\sum_{l=1}^{K_d} f_d[l] r'_{XX}[l] \approx \frac{1}{2} \mathrm{E}\left(\left[\tilde{X}'_n \right]^2 \right) \tag{3.91}$$

— An "interference" term that is a linear combination of the surrounding sub-ADCs' timing offsets, the derivative filter coefficients $f_d[l]$, and the derivative of the autocovariance function $r'_{XX}[l]$ taken at different time lags l. The proof in Appendix **??** shows that the derivative of the autocovariance function can be expressed:

$$r'_{XX}[l] = \frac{1}{2} \mathrm{E}\left((X_{n+l} - X_{n-l}) X'_n \right) \approx \mathrm{E}\left(\left(\tilde{X}_{n+l} - \tilde{X}_{n-l} \right) \tilde{X}'_n \right) \tag{3.92}$$

In fact, this linear system of equations relating the covariance functions $\{\Gamma_m, m \in \mathcal{M}\}$ with the the timing offsets $\{r_m, m \in \mathcal{M}\}$ can be inverted, and the timing offsets can be expressed in terms of the covariance functions. This system of equations is better seen

8. Since there are only M sub-ADCs, the subscript $m + l$ must be interpreted as $(m + l) \bmod M$ but the modulo sign is here voluntary omitted in order to keep the notations simple.

by rewriting expression 3.90 in matrix form:

$$\mathbf{\Gamma} = -\sum_{l=1}^{K_d} f_d[l] r'_{XX}[l] \mathbf{C}_l \mathbf{\Delta} \tag{3.93}$$

where $\mathbf{\Gamma} = [\Gamma_0, \cdots, \Gamma_{M-1}]^T$ is a column vector containing the covariance functions, $\mathbf{\Delta} = [r_0, \cdots, r_{M-1}]^T$ is a column vector containing the timing offsets, and $\mathbf{C}_l$ is a $M \times M$-matrix which coefficients $\mathbf{C}_l(i,j)$ are defined as follows[9]:

— If $l \bmod M \neq 0$:

$$\mathbf{C}_l(i,j) = \begin{cases} 2 \text{ if } i \neq j \\ -1 \text{ if } j = (i+l) \bmod M \text{ or } j = (i-l) \bmod M \\ 0 \text{ otherwise} \end{cases} \tag{3.94}$$

— If $l \bmod M = 0$:

$$\mathbf{C}_l(i,j) = 0 \; \forall (i,j) \in \mathcal{M}^2 \tag{3.95}$$

Each matrix $\mathbf{C}_l$ is a circulant matrix because each of its row vectors is equal to the above row vector rotated one element to the right. In other words, the matrix $\mathbf{C}_l$ is comprised of shifted versions of its topmost row vector:

$$\mathbf{c}_l = \left[2, \underbrace{0 \cdots 0}_{(l \bmod M)-1 \text{ zeros}}, -1, \underbrace{0 \cdots 0}_{M-2(l \bmod M)-1 \text{ zeros}}, -1, \underbrace{0 \cdots 0}_{(l \bmod M)-1 \text{ zeros}} \right]^T \tag{3.96}$$

The $M \times M$ matrix $\mathbf{C} = -\sum_{l=1}^{K_d} f_d[l] r'_{XX}[l] \mathbf{C}_l$ is also a circulant matrix because circulant matrices form a commutative algebra. The matrix $\mathbf{C}$ is assumed to have a rank equal to $M-1$ even though it difficult to formally prove (see Appendix **??**). Having a rank of $M-1$ makes the matrix non-invertible, which means that the timing offset vector $\mathbf{\Delta}$ cannot be obtained by solving the system of equations $\mathbf{\Gamma} = \mathbf{C}\mathbf{\Delta}$.

Intuitively, this non-invertability is not too surprising because the timing offset vector $\mathbf{\Delta}$ contains the absolute timing offsets of the sub-ADCs. It consequently contains information about the global delay of the TIADC output signal with respect to the input signal. Of course, it is not possible to infer the absolute sampling delay from the TIADC output signal because the absolute time information does not exist anymore after sampling. The system has an infinite number of solutions. If a vector of timing offsets is a solution to the system, then adding a global delay to each of its coordinate, generates another solution to the system. This is not an issue for our problem because we only want to measure the timing offset difference between the sub-ADCs. We have no interest

9. i represents the row index, j represents the column index

in the global delay.

It is easy to slightly modify the system of equations in order to suppress the contribution of the global delay. For instance, we can arbitrarily choose that the timing offset of sub-ADC 0 is $r_0 = 0$ (or any other value) and solve the system for the remaining timing offsets $\{r_m, m \in \mathcal{M}^*\}$. In that case, the system of equation to solve becomes:

$$\mathbf{\Gamma} = \mathbf{C}_r \mathbf{\Delta}_r \tag{3.97}$$

where $\mathbf{\Delta}_r = [r_1, \cdots, r_{M-1}]^T$ and $\mathbf{C}_r$ is the $M \times M-1$ matrix obtained by removing the first column of $\mathbf{C}$. The reduced matrix $\mathbf{C}_r$ has $M-1$ independent columns because the rank of $\mathbf{C}$ is $M-1$. Consequently, $\mathbf{C}_r^T\mathbf{C}_r$ is non-singular and the system of equations defined in 3.97 can be solved using the pseudo-inverse of $\mathbf{C}_r$. Solving the system gives the value of the timing offsets as functions of the covariance functions [10]:

$$\mathbf{\Delta}_r = \left(\mathbf{C}_r^T\mathbf{C}_r\right)^{-1}\mathbf{C}_r^T\mathbf{\Gamma} \tag{3.98}$$

The last step is to replace the quantities in this equation by their estimates. Doing so leads to the skew mismatch estimation formula:

$$\hat{\mathbf{\Delta}}_r = \left(\hat{\mathbf{C}}_r^T\hat{\mathbf{C}}_r\right)^{-1}\hat{\mathbf{C}}_r^T\hat{\mathbf{\Gamma}} \tag{3.99}$$

where

— $\hat{\mathbf{\Delta}}_r = [\hat{r}_1, \cdots, \hat{r}_{M-1}]$ is a vector containing the timing delays estimates of sub-ADC 1 to sub-ADC $M-1$ (sub-ADC 0 is assumed to have a delay equal to zero)

— $\hat{\mathbf{C}}_r$ is a $M \times M-1$ matrix containing the $M-1$ last columns of $\hat{\mathbf{C}} = \sum_{l=1}^{K_d} f_d[l]\hat{r}'_{XX}[l]\mathbf{C}_l$. The sampled autocovariance derivative $r'_{XX}[l]$ can be estimated by replacing the random variables in Equation 3.92 by their realization, and by averaging over N samples:

$$\hat{r}'_{XX}[l] = \frac{1}{2N}\sum_{n=0}^{N-1}(\tilde{x}[n+l] - \tilde{x}[n-l])\tilde{x}'[n] \tag{3.100}$$

— $\hat{\mathbf{\Gamma}} = \left[\hat{\Gamma}_0, \cdots, \hat{\Gamma}_{M-1}\right]$ is a vector containing the estimates of the covariance between

10. The skew estimation formula 3.98 is easily obtained by multiplying each side of expression 3.97: $\mathbf{C}_r^T$:

$$\mathbf{C}_r^T\mathbf{\Gamma} = \mathbf{C}_r^T\mathbf{C}_r\mathbf{\Delta}_r$$

and then by $\left(\mathbf{C}_r^T\mathbf{C}_r\right)^{-1}$:

$$\mathbf{\Delta}_r = \left(\mathbf{C}_r^T\mathbf{C}_r\right)^{-1}\mathbf{C}_r^T\mathbf{\Gamma}$$

the signal derivative and the sub-ADCs' output signals:

$$\hat{\Gamma}_m = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_m[k]\tilde{x}'_m[k] \tag{3.101}$$

3.6.3.3 Theoretical explanation – non-stationary process

As with gain mismatch estimation, the skew mismatch estimation formula in Equation 3.101 is valid, with some constraints, when the input signal is a realization of a non-stationary random process. This is better understood by looking at the calculation of the covariance between each sub-ADC's output signal and its derivative (Equation 3.88) when the input random process $\{X_t\}$ is non-stationary:

$$\Gamma_m = \sum_{l=1}^{K_d} f_d[l]\left[\mathrm{E}\left(\tilde{X}_{kM+m}\tilde{X}_{kM+m+l}\right) - \mathrm{E}\left(\tilde{X}_{kM+m}\tilde{X}_{kM+m-l}\right)\right] \tag{3.102}$$

Now, the autocovariance function depends both on the time and the time lag:

$$\begin{aligned}\Gamma_m = \sum_{l=1}^{K_d} f_d[l]\,[&r_{XX}((kM+m+r_m)T_s, (l+r_{m+l}-r_m)T_s) \\ &- r_{XX}((kM+m+r_m)T_s, (l-r_{m-l}+r_m)T_s)]\end{aligned} \tag{3.103}$$

As with the WSS case, this expression can be expanded using a first-order Taylor approximation with respect to the time-lag τ. As in Equation 3.90, the two zero-order terms cancel out:

$$\Gamma_m \approx \sum_{l=1}^{K_d} f_d[l]T_s(-2r_m + r_{m+l} + r_{m-l})\frac{\partial r_{XX}}{\partial \tau}((kM+m+r_m)T_s, lT_s) \tag{3.104}$$

This equation can further be simplified by linearizing with respect to the time variable t, and by neglecting the second-order terms:

$$\begin{aligned}\Gamma_m \approx &\sum_{l=1}^{K_d} f_d[l]T_s(-2r_m + r_{m+l} + r_{m-l})\frac{\partial r_{XX}}{\partial \tau}((kM+m)T_s, lT_s) \\ &+ \sum_{l=1}^{K_d} f_d[l]\underbrace{T_s^2 r_m(-2r_m + r_{m+l} + r_{m-l})}_{\approx 0}\frac{\partial^2 r_{XX}}{\partial \tau \partial t}((kM+m)T_s, lT_s)\end{aligned} \tag{3.105}$$

Therefore, the covariance between each sub-ADC's output signal and its derivative can be written in terms of the partial derivative of the discrete autocovariance function and

the timing offsets:

$$\Gamma_m \approx \sum_{l=1}^{K_d} f_d[l] T_s(-2r_m + r_{m+l} + r_{m-l}) \frac{\partial r_{XX}}{\partial \tau}[kM + m, l] \tag{3.106}$$

where $\frac{\partial r_{XX}}{\partial \tau}[kM + m, l] = T_s \frac{\partial r_{XX}}{\partial \tau}((kM + m + r_m)T_s, lT_s)$. The $\{\Gamma_m, m \in \mathcal{M}\}$ are time-dependent, as opposed to the WSS case. However, the estimates $\{\hat{\Gamma}_m, m \in \mathcal{M}\}$ of $\{\Gamma_m, m \in \mathcal{M}\}$, obtained by replacing the expectation by a finite-length average, converge toward a value independent of the sub-ADC index m if certain conditions are verified. If we replace those estimates by their expression, we obtain:

$$\hat{\Gamma}_m = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_m[k] \tilde{x}'_m[k] = \sum_{l=1}^{K_d} f_d[l] T_s(-2r_m + r_{m+l} + r_{m-l}) \frac{1}{N} \sum_{k=0}^{N-1} \frac{\partial r_{XX}}{\partial \tau}[kM + m, l] \tag{3.107}$$

In order for the above expression to be independent of the sub-ADC index m, the average of the autocovariance function's derivative must converge to a value independent of m. This requirement enforces conditions on the autocovariance function that are similar to those enforced on the variance with gain mismatch estimation. Without diving into the same level of details as in section 3.5.3.3, one can say that the average $\frac{1}{N} \sum_{k=0}^{N-1} \frac{\partial r_{XX}}{\partial \tau}[kM + m, l]$ converges to a value independent of m if the *partial derivative* $\frac{\partial r_{XX}}{\partial \tau}(t, \tau)$ of the autocovariance function with respect to τ does not contain significant periodic components (with respect to the variable t) at frequencies multiples of F_s/M. If this is verified for the autocovariance function itself (see conditions in section 3.5.3.3), then it is also verified for its *partial derivative* with respect to τ. Indeed, differentiating with respect to τ is a linear operation that does not add new components in the frequency domain associated to the variable t.

Thus, when the average length N tends to infinity, the estimates $\{\hat{\Gamma}_m, m \in \mathcal{M}\}$ only depend on the sub-ADCs' timing offsets and the m-independent stationary components $\overline{\frac{\partial r_{XX}}{\partial \tau}}[l]$ of the autocovariance function partial derivatives:

$$\lim_{N \to +\infty} \frac{1}{N} \sum_{k=0}^{N-1} \tilde{x}_m[k] \tilde{x}'_m[k] = \sum_{l=1}^{K_d} f_d[l] T_s(-2r_m + r_{m+l} + r_{m-l}) \overline{\frac{\partial r_{XX}}{\partial \tau}}[l] \tag{3.108}$$

where $\overline{\frac{\partial r_{XX}}{\partial \tau}}[l] = \lim_{N \to +\infty} \frac{1}{N} \sum_{k=0}^{N-1} \frac{\partial r_{XX}}{\partial \tau}[kM + m, l]$.

This quick analysis demonstrates that the skew mismatch estimation formula in Equation 3.99 is valid when the input signal is a realization of a non-stationary random process if its autocovariance function verifies specific conditions. As seen in Section 3.5.3.3, those conditions are generally met for typical communication signals.

3.6.4 Generalization

Interestingly, if we have a closer look at Equation 3.93, we notice that the filter coefficients $\{f_d[l], l \in \{1, \cdots, K_d\}\}$ could be chosen arbitrarily, i.e. that they do no need to be the coefficients of a derivative filter. One could pick any set of coefficients $\{f_e[l], l \in \{1, \cdots, K_e\}\}$ without changing the skew estimation formula. As long as the values of the autovariance derivative $\{r'_{XX}[l], l \in \{1, \cdots, K_e\}\}$ are known, the skew mismatch estimation formula remains valid [11].

It means that correlating each sub-ADC's output signal $x_m[k]$ with the output signal $x^e_m[k]$ of any antisymmetric filter does not change the estimation technique. This is a big advantage when it comes to circuit implementation. The proof in Appendix **??** shows that one cannot be completely sure that the matrix $\mathbf{C}$ has a rank $M-1$, and therefore that the matrix $\mathbf{C}_r^T\mathbf{C}_r$ is invertible. It is technically possible to design a matrix inversion circuit that verifies the singularity of a matrix before trying to invert it but it is an additional burden that a designer prefers to avoid.

A way to avoid performing the matrix inversion is to use a one-coefficient filter, such that $K_e = 1$ and $f_e[1] = 1$. In this case, the matrix $\mathbf{C}$ has a simple form:

$$\mathbf{C} = -r'_{XX}[1]\mathbf{C}_1 \tag{3.109}$$

It can be proven (see Appendix **??**) that the matrix $\mathbf{C}$ has a rank $M-1$ if $r'_{XX}[1] \neq 0$. The skew mismatch estimation formula becomes

$$\hat{\mathbf{\Delta}}_r = \left(\hat{\mathbf{C}}_r^T\hat{\mathbf{C}}_r\right)^{-1}\hat{\mathbf{C}}_r^T\hat{\mathbf{\Gamma}} = \frac{1}{\hat{r}'_{XX}[1]}\left(\mathbf{C}_{1r}^T\mathbf{C}_{1r}\right)^{-1}\mathbf{C}_{1r}\hat{\mathbf{\Gamma}} \tag{3.110}$$

where $\mathbf{C}_{1r}$ is a $M \times M-1$ matrix obtained by taking $\mathbf{C}_1$ and removing its first column. As opposed to the multi-coefficient case, the matrix to invert is constant such that its inversion does not have to be performed on-chip. The calculation of the pseudo-inverse matrix $\left(\mathbf{C}_{1r}^T\mathbf{C}_{1r}\right)^{-1}\mathbf{C}_{1r}$ can be performed off-chip, and the result can be hard-coded in the circuit.

3.6.4.1 Circuit implementation

Skew mismatch estimation as described in Equation 3.110 is easy to implement in a circuit. The estimation is direct (as opposed to adaptive), which means that it can be performed with one iteration with no need for a feedback loop. The timing offset estimates are available after calculating a finite-length average over a single batch of

11. The derivative filter is still needed for the correction part and for the estimation of the covariance derivatives.

$M \times N$ samples. The estimation can be performed at start up, and then continuously or sparsely during the normal operation of the circuit.

Figure 3.23 shows the global block diagram of the circuit implementation. The design

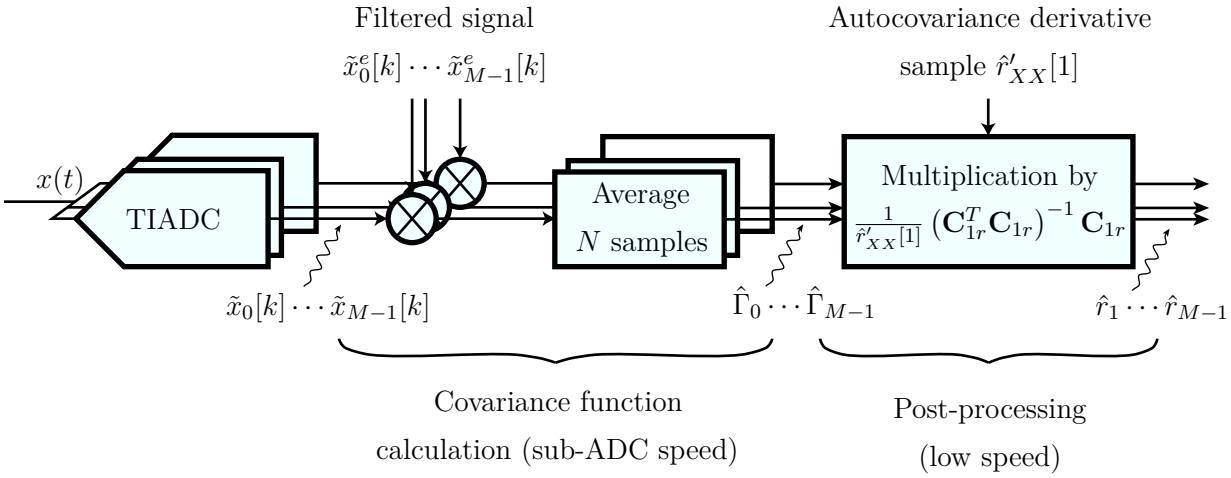


Figure 3.23 – Skew mismatch estimation block diagram

can be separated into a "high-speed" part and a "low-speed" part. The "high-speed" part corresponds to the calculation of the covariance between each sub-ADC's output signal and its corresponding derivative, which needs to have a throughput equal to the TIADC sampling rate. The matrix multiplication, on the other hand, can be done at a much lower rate.

The calculation of the post-processing matrix $\left(\mathbf{C}_r^T \mathbf{C}_r\right)^{-1} \mathbf{C}_r$ uses the estimate of the autocovariance derivative $\hat{r}'_{XX}[1]$. This estimate is obtained by adding a stage to the derivative filter, as illustrated in Figure 3.24. This is possible because the calculation reuses some terms that are also used for the calculation of the signal derivative. Indeed, Equation 3.100 shows that the calculation of $\hat{r}'_{XX}[1]$ contains the product between the signal derivative taken at time n, and the difference between adjacent signal samples taken respectively at time $n+1$ and time $n-1$. The former term is the output of the derivative filter while the latter term is an intermediate calculation done in one of the filter stages.

3.6.4.2 Accuracy analysis

The skew mismatch SNDR equation in Chapter 2 shows that when the skew mismatches are small the SNDR depends on the frequency of the input signal and the RMS skew mismatch level σ_r. The residual timing offsets $\{\tilde{r}_m = \hat{r}_m - r_m, m \in \mathcal{M}\}$ are defined as the difference between the true timing offsets and their estimates, and the RMS

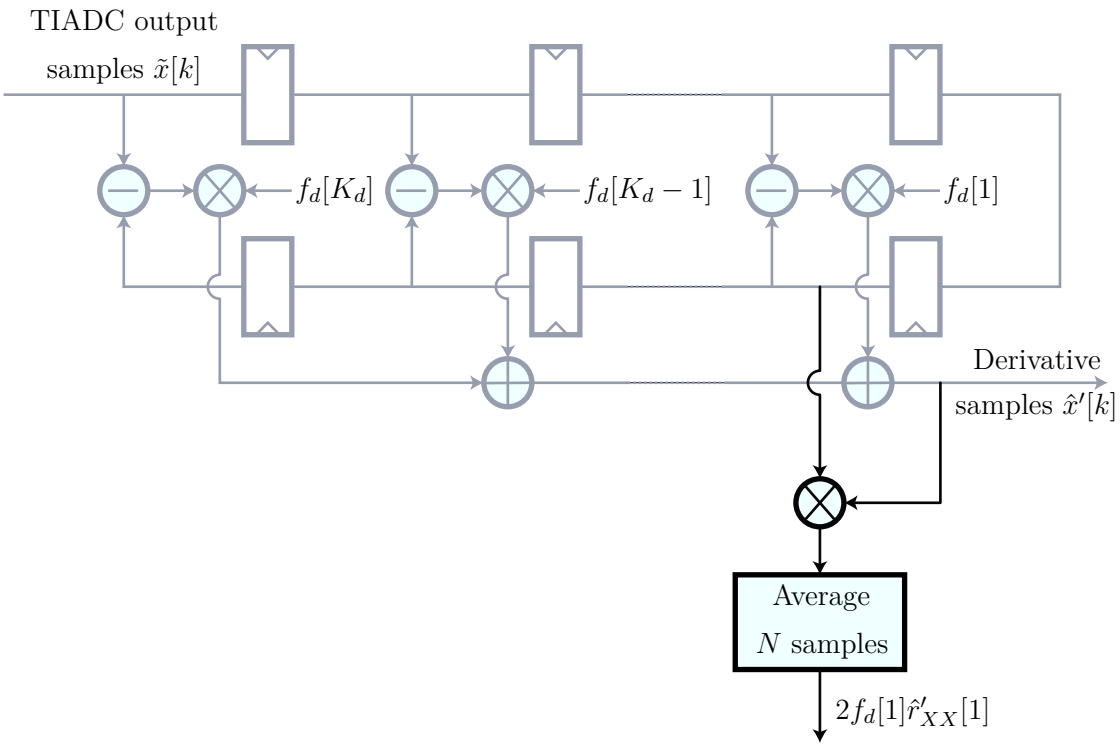


Figure 3.24 – Modification of the derivative filter implementation to calculate the autocovariance derivative

residual skew mismatch level $\sigma_{\tilde{r}}$ is defined as

$$\sigma_{\tilde{r}} = \frac{1}{M} \sum_{m=0}^{M-1} \tilde{r}_m^2 \tag{3.111}$$

For a sine input, the SNDR due to residual timing skew mismatches is

$$\text{SNDR} = 20 \log_{10} (\omega_0 \sigma_{\tilde{r}}) \tag{3.112}$$

where ω_0 is the angular frequency of the sinusoid.

The plots displayed in Figure 3.25 show the SNDR as a function of the average length N when the skew estimation is performed on a QAM modulated signal. The SNDR is calculated for a sine input with an input frequency equal to the cut-off frequency of the derivative filter, which in this example is 0.8π.

Not surprisingly, the accuracy of the estimation increases when the average length increases. It appears that the bandwidth of the QAM modulated signal (normalized to F_s) does not affect the performance much. However, the higher is the carrier frequency, the better is the estimation. This effect is normal because the mismatch effects are smaller at a lower carrier frequency, and it takes more time to average out the signal

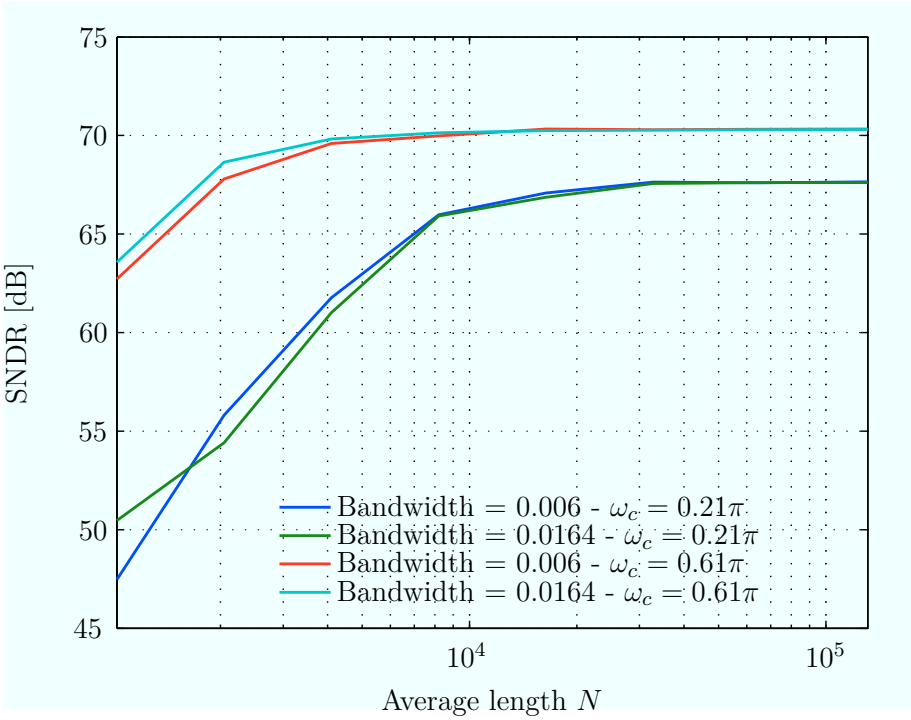


Figure 3.25 – [Skew mismatch estimation accuracy as a function of the average length ($M = 12$, $\sigma_r = 1\%$, sine input frequency $\omega_0 = 0.8\pi$, residual timing offset mismatches averaged over 50 Monte-Carlo simulations, no quantization)

contribution. Interestingly, above a certain average length, the SNDR reaches a plateau where the estimation accuracy does not improve anymore. This plateau happens when the second-order timing errors become limiting.

The residual level of skew mismatch noise reached when $N > 10^5$ is however equivalent to the quantization noise of a 11-bit ADC. In practice, the average length often needs to be higher in a real TIADC because other sources of noise also need to be averaged out (quantization noise or thermal noise for example).

3.6.5 Conclusion

This section presented a non-iterative skew mismatch calibration technique that relies on the timing offsets being small as compared to the sampling period. The estimation algorithm is guaranteed to converge when the input signal is a realization of a non-stationary random process verifying certain requirements. Specifically, the non-stationary component of the autocovariance function must not contain spurious tones at frequencies multiple of F_s/M. In practice, we showed that this requirement held true for pulse-shaped modulated signals.

In a real implementation, the skew mismatch calibration is the last stage of calibration because it must be performed after the correction of the static gain mismatches, the

frequency-dependent gain mismatches and the frequency-dependent phase mismatches.

The next section describes the bandwidth mismatch calibration technique. As we will see, it borrows some concepts from the skew mismatch calibration technique, particularly for the correction phase.

3.7 Bandwidth mismatch calibration

3.7.1 Introduction

Bandwidth mismatch calibration is complicated because bandwidth mismatches create frequency-dependent gain mismatches, and frequency dependent phase mismatches. To simplify the problem, we only look at first-order bandwidth mismatches, which are a good approximation of mismatches that occur in practice. The bandwidth mismatch calibration algorithm is performed in two distinct phases:

— estimating the mismatches by calculating the time-constant associated to each sub-ADC (or equivalently the cut-off frequencies)

— correcting the mismatches by equalizing the sub-ADC frequency responses

The bandwidth mismatch estimation algorithm, described in section 3.7.3, is done by measuring the frequency dependent gain mismatches that are induced by the bandwidth mismatches. The bandwidth mismatch correction technique is detailed in Section 3.7.2 and is very similar to the timing skew mismatch correction technique.

3.7.2 Bandwidth mismatch correction

3.7.2.1 General idea

The aim of the bandwidth mismatch correction stage is to correct the TIADC output signal such that the sub-ADCs appear to have been passed through filters with equal frequency responses. This step requires the sub-ADC time-constants to be known. The bandwidth mismatch estimation stage, which operation is explained in section 3.7.3, provides estimates of those time constants.

The structure of the bandwidth mismatch correction stage, shown in Figure 3.26, is similar to the structure of the skew mismatch correction stage, albeit with a different filter.

We assume that the time-constant offsets are small as compared to the original nominal time-constant so that the sampling error at each sub-ADC's output can be linearized. With this linearization, a sub-ADC's sampling error due to bandwidth mismatches is proportional to the sub-ADC's time-constant offset and the low-pass filtered signal's

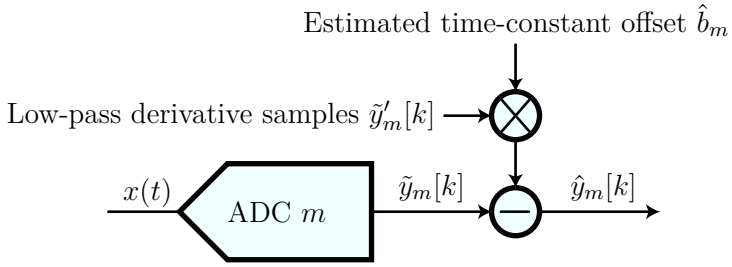


Figure 3.26 – Bandwidth mismatch correction block diagram

derivative. Once the time-constant offset is known, the original signal is recovered by calculating the error signal and by subtracting it from each sub-ADC's output signal.

3.7.2.2 Theoretical explanation

In a TIADC, each sub-ADC's input signal is filtered by a different first-order low-pass filter (see explanation in Chapter 2). Each filter's continuous frequency response $\{H_m(\Omega), m \in \mathcal{M}\}$ is defined as:

$$H_m(\Omega) = \frac{1}{1 + j\Omega\tau_m} \tag{3.113}$$

where $\{\tau_m, m \in \mathcal{M}\}$ is the time-constant of the low-pass filter. Each filter's time-constant can also be expressed as a sum between the average (or nominal) time-constant $\bar{\tau}$ and a time-constant offset $\{\Delta\tau_m, m \in M\}$. Using this notation, the filter's frequency response becomes:

$$H_m(\Omega) = \frac{1}{1 + j\Omega(\bar{\tau} + \Delta\tau_m)} \tag{3.114}$$

If the mismatches are small, i.e. if the time-constant offsets are small as compared to the nominal time constant ($\Delta\tau_m \ll \bar{\tau}$), the filter frequency response can be expanded by its Taylor series around the nominal time constant $\bar{\tau}$:

$$H_m(\Omega) \approx \frac{1}{1 + j\Omega\bar{\tau}} \left(1 + \frac{\Delta\tau_m}{\bar{\tau}} \frac{j\Omega\bar{\tau}}{1 + j\Omega\bar{\tau}}\right) \tag{3.115}$$

The CTFT of each sub-ADC's output signal (before sampling) is a sum of an ideal term $Y(\Omega) = \frac{X(\Omega)}{1+j\Omega\bar{\tau}}$ corresponding to the TIADC input signal filtered by the nominal low-pass

filter, and an error term:

$$\begin{aligned}\tilde{Y}_m(\Omega) &= \frac{X(\Omega)}{1+j\Omega\bar{\tau}}\left(1+\frac{\Delta\tau_m}{\bar{\tau}}\frac{j\Omega\bar{\tau}}{1+j\Omega\bar{\tau}}\right) \\ &= \underbrace{Y(\Omega)}_{\text{desired signal}} + \underbrace{\frac{\Delta\tau_m}{\bar{\tau}}Y(\Omega)\frac{j\Omega\bar{\tau}}{1+j\Omega\bar{\tau}}}_{\text{error term}}\end{aligned} \tag{3.116}$$

The error signal is proportional to the normalized time-constant offset and the desired signal filtered by a low-pass derivative filter. The frequency response of the low-pass derivative filter is:

$$H_b(\Omega) = \frac{j\Omega\bar{\tau}}{1+j\Omega\bar{\tau}} \tag{3.117}$$

The denomination "low-pass derivative filter" stems from the fact that the frequency response is a product between a derivative filter (response $j\Omega$) and a first-order low-pass filter (response $\frac{1}{1+j\Omega\bar{\tau}}$).

It is clear from Equation **??** that the bandwidth mismatch correction problem is similar to the skew mismatch correction problem. The only differences are that a low-pass derivative filter H_b replaces the derivative filter, and the time-constant offsets replace the timing offsets. The bandwidth mismatch reconstruction equation can consequently be expressed as follows:

$$\hat{y}_m[k] = \tilde{y}_m[k] - \hat{\beta}_m\tilde{y} * f_b[kM+m] \tag{3.118}$$

where $f_b[n]$ is the discrete impulse response of the low-pass derivative filter, $\tilde{y}[n]$ represents the mismatch-corrupted TIADC output signal, and $\hat{\beta}_m = \frac{\hat{\Delta}\tau_m}{\bar{\tau}}$ is the estimated normalized time-constant offset. The low-pass derivative filter output signal associated to each sub-ADC is denoted $\tilde{y}'_m[n] = \tilde{y} * f_b[kM+m]$.

The discrete impulse response of the low-pass derivative filter is obtained by convolving the continuous time impulse response $h_b(t)$ with an antialiasing filter, and by sampling the result at interval nT_s:

$$f_b[n] = \int_{-\infty}^{+\infty} h_b(u)\,\mathrm{sinc}\left(\frac{nT_s-u}{T_s}\right)du \tag{3.119}$$

The term $h_b(t)$ is the continuous-time impulse response of the low-pass derivative filter, and it is equal to the derivative of the continuous impulse response $h(t) = e^{-t/\bar{\tau}}$ of the nominal low-pass filter:

$$h_b(t) = \frac{d}{dt}h(t) = -\frac{1}{\bar{\tau}}e^{-t/\bar{\tau}} \tag{3.120}$$

Inserting the expression of $h_b(t)$ into Equation 3.119 gives the discrete impulse response

of the low-pass derivative filter:

$$f_b[n] = -\frac{1}{\bar{\tau}} \int_{-\infty}^{+\infty} e^{-u/\bar{\tau}} \operatorname{sinc}\left(\frac{nT_s - u}{T_s}\right) du \tag{3.121}$$

The right-hand side of this equation does not have a close-form expression but the coefficients can be approximated with the help of mathematical tools such as Matlab (see in next section).

3.7.2.3 Circuit implementation

The $2K_b + 1$ low-pass derivative filter coefficients are obtained similarly to those of the derivative filter, by windowing the infinite filter impulse response in Equation 3.121. Since no close form of the impulse response exists, the filter coefficients are calculated using the symbolic calculation toolbox of Matlab. An example of those coefficients is plotted in Figure 3.28b.

The hardware implementation of the low-pass derivative filter is shown in 3.28. As

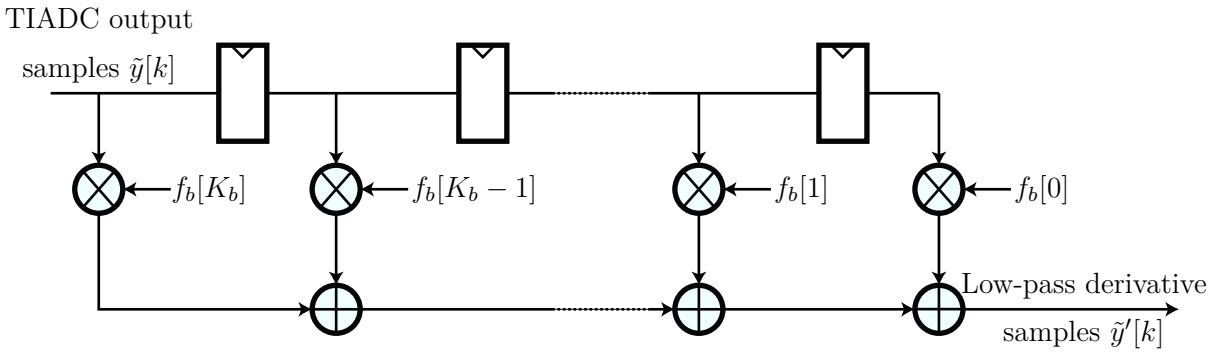
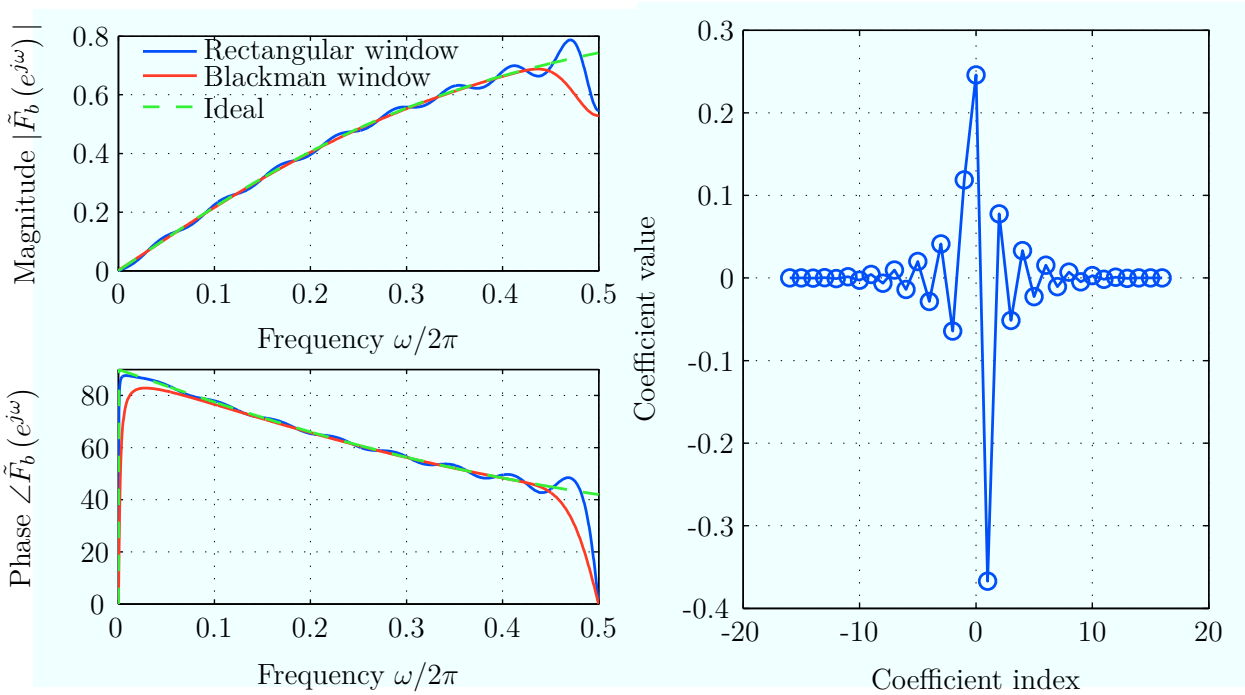


Figure 3.27 – Efficient low-pass derivative filter implementation

opposed to the derivative filter, the coefficients of the low-pass derivative filter are not symmetric. They cannot be regrouped in order to reduce the number of fixed multipliers. The frequency response of the filter is shown in Figure 3.28a. Once again, one can notice that the windowing improves the accuracy of the in-band response by reducing the ripples.

Naturally, increasing the number of coefficients leads to a frequency response closer the ideal frequency response. Increasing the number of coefficients enables to increase the filter's cut-off frequency, and an analysis similar to that of the derivative filter accuracy could be done here.

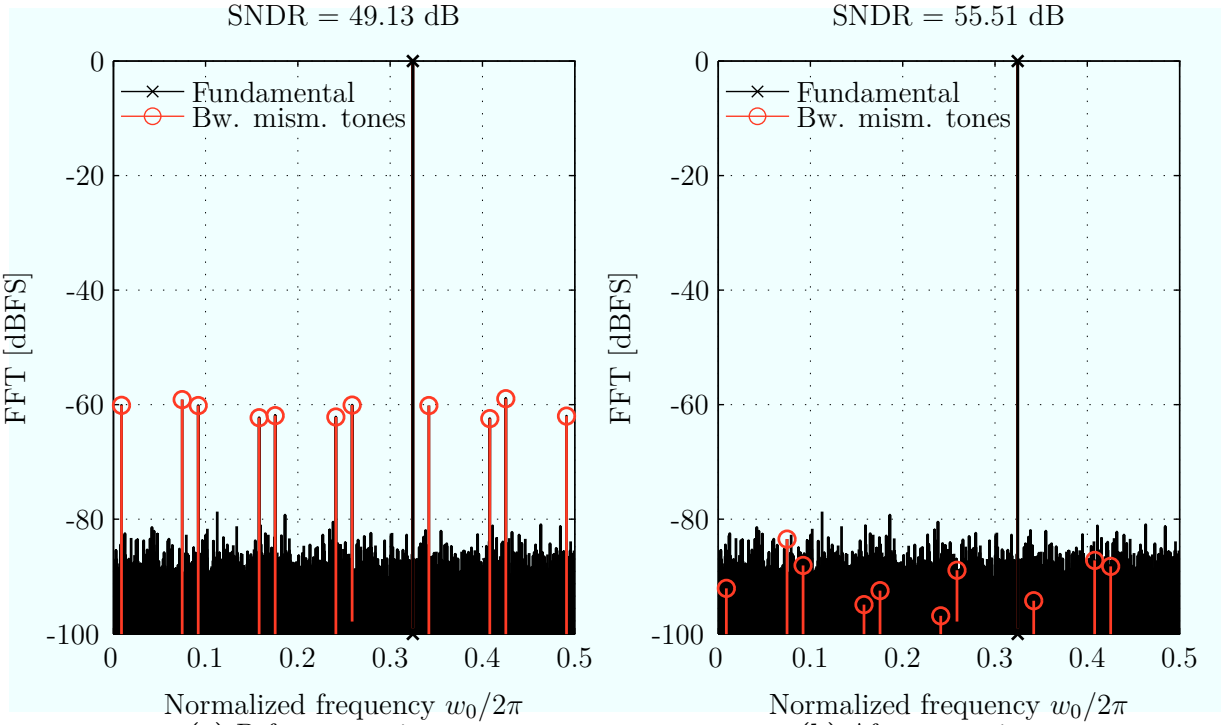
Figure 3.29 illustrates the performance of the bandwidth mismatch correction technique by showing the TIADC output spectrum before and after correction. The bandwidth mismatch spurious tones are reduced from -60 dBFS level down to the noise floor.



(a) Frequency response **(b)** Impulse response

Figure 3.28 – Frequency response of the low-pass derivative filter for different windows ($K_b = 16$)

The design of the low-pass derivative filter requires to know the nominal time-constant of the sub-ADCs. This average time-constant can be obtained through Monte-Carlo simulations of the analog front-end. However, in practice, nothing guarantees that the sub-ADC time-constants of one particular chip will be centered around this nominal time-constant. It is possible that for one particular chip, the filter time-constant is underestimated or overestimated. The impact of this problem on the correction accuracy is illustrated in Figure 3.30. The figure shows that the SNDR after correction decreases when the input signal frequency increases. This degradation at high frequency stems from the fact that the second-order bandwidth mismatch errors become dominant at high input signal frequency. Overestimating or underestimating the nominal time-constant primarily degrades the performance at low frequencies, where the second-order errors are not dominant.



(a) Before correction **(b)** After correction

Figure 3.29 – Output spectrum before and after bandwidth mismatch correction ($B = 9$, $M = 12$, $K_b = 32$, $N_{\text{FFT}} = 12 \times 4096$, $\sigma_\beta = 1\%$ $\bar{b} = 1/2\pi$, true time-constant offsets used for correction)

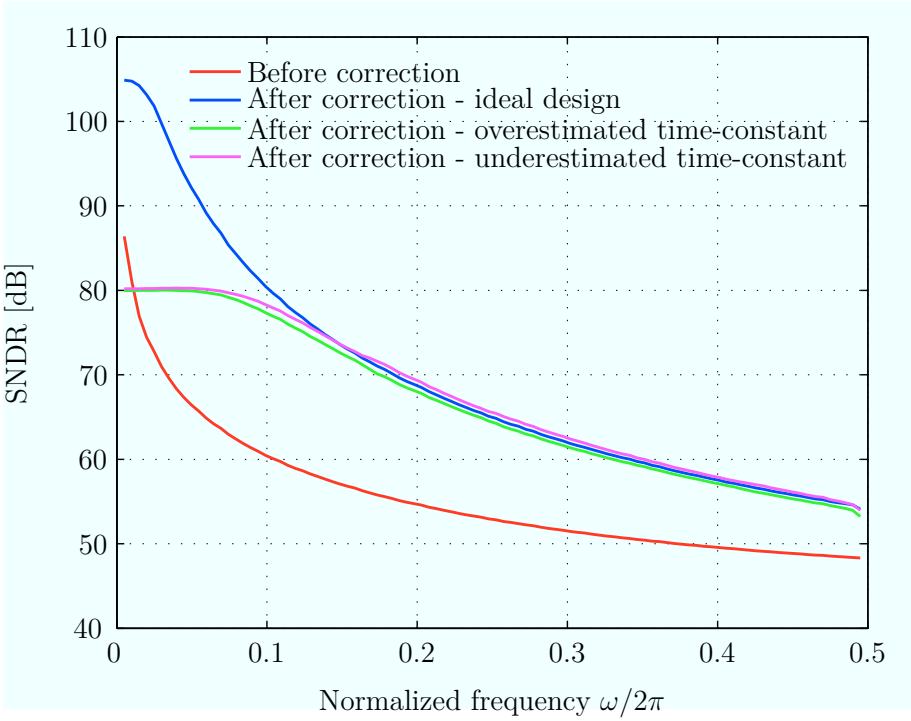


Figure 3.30 – Bandwidth mismatch estimation accuracy as a function of the input frequency ($M = 12$, $\sigma_\beta = 1\%$, $K_b = 128$, $\bar{b} = 1/2\pi$, true time-constant offsets used for correction)

3.7.3 Bandwidth mismatch estimation

3.7.3.1 General idea

Bandwidth mismatches create both frequency-dependent gain and phase mismatches. When the input signal $x(t) = A\cos(\Omega t + \phi)$ is a sine wave, the signal $y(t)$ seen by each sub-ADC is

$$y(t) = \frac{A}{\sqrt{1 + (\Omega\tau)^2}} \cos\left(\Omega t + \phi - \operatorname{atan}(\Omega\tau)\right) \tag{3.122}$$

When there are bandwidth mismatches, the sub-ADC output signals $\{\tilde{y}_m[k], m \in \mathcal{M}\}$ are therefore:

$$\tilde{y}_m[k] = A\gamma_m(\omega) \cos\left(\omega(kM + m) + \phi - \varphi_m(\omega)\right) \tag{3.123}$$

where $\gamma_m(\omega)$ is a frequency-dependent gain

$$\gamma_m(\omega) = \frac{1}{\sqrt{1 + \omega(\bar{b} + \Delta b_m)^2}} \tag{3.124}$$

and $\varphi_m(\omega)$ is a frequency-dependent phase

$$\varphi_m(\omega) = -\operatorname{atan}(\omega(\bar{b} + \Delta b_m) \tag{3.125}$$

Let us remind that $\bar{b} = \bar{\tau}/T_s$ is the normalized average time-constant and $\{\Delta b_m = \Delta\tau/T_s, m \in \mathcal{M}\}$ are the normalized time-constant offsets.

It seems that the bandwidth mismatches can be estimated by looking either at a gain-dependent quantity or at a phase dependent quantity (or a combination of both). The approach that was chosen in this work was to measure a gain-dependent quantity. However, it is realistic to think that it is possible to extend the findings in this chapter by including measurements of a phase-dependent quantity.

A natural idea is to use the gain mismatch estimation algorithm presented earlier to estimate the relative gains between the sub-ADC, and deduce the amount of bandwidth mismatch from the measurement. However, this approach is only possible if the static gain mismatches, i.e. the non frequency-dependent gain mismatches, are absent or small. If there are static gain mismatches, the estimation of the variance of each sub-ADC's output signal is affected both by the frequency-dependent gain mismatches and the static gain mismatches, and it is impossible to distinguish between the two types of mismatches.

Therefore, it is necessary to isolate the frequency-dependent gain mismatch effects from the static gain mismatch effects. A solution to tackle this problem is to measure the variance of each sub-ADC's output signal for different frequency bands. Conceptually, the idea is to measure each sub-ADC's output power for low input signal frequencies, and then each sub-ADC's output power for high input signal frequencies. These two powers supposedly differ from each other, and taking their ratio eliminates the static gain contribution.

In order for these quantities to be non-zero, the signal must have frequency content both at high and low input frequencies. For example, let us say that the signal is a sum of a sine input at a low frequency Ω_L with amplitude A_L, and a sine input a high frequency Ω_H with amplitude A_H:

$$x(t) = A_L \cos(\Omega_L t) + A_H \cos(\Omega_H t) \tag{3.126}$$

In presence of both static gain mismatches and bandwidth mismatches, each sub-ADC's output signals can be expressed:

$$\begin{aligned}\tilde{y}_m[k] = {} & A_L g_m \gamma_m(\omega_L) \cos\left(\omega_L(kM+m) - \varphi_m(\omega_L)\right) \\ & + A_H g_m \gamma_m(\omega_H) \cos\left(\omega_H(kM+m) - \varphi_m(\omega_H)\right)\end{aligned} \tag{3.127}$$

Let us suppose that there is a way to measure each sub-ADC's output power $\{P_m^L, m \in \mathcal{M}\}$ corresponding to low input frequencies, and each sub-ADC's output power $\{P_m^H, m \in$

$\mathcal{M}\}$ corresponding to high input frequencies, such that

$$P_m^L = \frac{1}{2} A_L^2 g_m^2 \gamma_m^2(\omega_L) \tag{3.128}$$

$$P_m^H = \frac{1}{2} A_H^2 g_m^2 \gamma_m^2(\omega_H) \tag{3.129}$$

For each sub-ADC, the ratio β_m of these powers is independent of the static gain:

$$\beta_m = \frac{A_L^2 \gamma_m^2(\omega_L)}{A_H^2 \gamma_m^2(\omega_H)} \tag{3.130}$$

These power ratios can serve as a measure of the bandwidth mismatches because they only contain information about the frequency-dependent gains of the sub-ADCs. In expression 3.130, it is interesting to notice that the ratios become independent of the bandwidth mismatches if the frequencies of the two sine waves are the same, or if the amplitude of one of the sine waves if zero. It tells us that estimating the bandwidth mismatches requires the input signal to span a sufficient frequency range, which seems quite intuitive given that we are trying to detect frequency-dependent mismatches. Doing so on a single sine wave with unknown frequency is obviously not possible.

The next section provides an analytical approach to explain how to extract the time-constant offset estimates from the power ratios.

3.7.3.2 Cosine and sine filtering

The idea presented in the previous section requires measuring the power of the signal in two different frequency regions. However, it we filter the TIADC output signal, the filter output signal is a linear combination of of the sub-ADC output signals, and it contains information about each single sub-ADC. The only viable solution is to filter each sub-ADC's output signal separately, which constrains a lot the frequency response of the filter that it is possible to design. Since each sub-ADC's output signal is a subsampled version of the TIADC output signal, the equivalent frequency response of the filter at the TIADC full rate is comprised of aliases at multiple of F_s/M of the frequency response of the filter at the sub-ADC sampling rate, as shown in Figure 3.31. In other words, if the frequency response of the filter at the output of the sub-ADC is $H(e^{j\omega'})$, its equivalent frequency response the TIADC sampling rate is $H(e^{j\omega}) = \frac{1}{M} H\left(e^{j\omega'}\right)\Big|_{\omega'=M\omega}$

The original idea of extracting low and high frequency components of the entire signal by filtering only one sub-ADC output is therefore compromised. Instead, we can design sub-ADC filters which frequency responses are somewhat complementary, such that one filter one filter has its passband where the other one has its stopband. Sine and cosine filters loosely correspond to this description. Their z-transforms are $H_{\sin}(z) = 0.5z -$

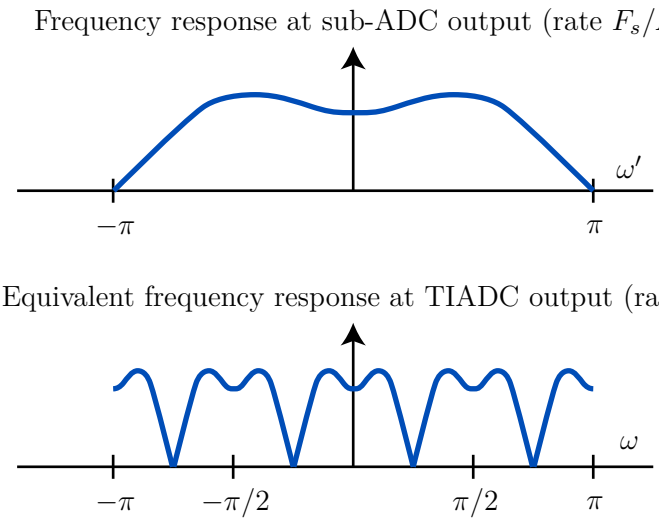


Figure 3.31 – Effects of upsampling on the frequency response of a filter ($M = 4$)

$0.5z^{-1}$ and $H_{\cos}(z) = 0.5z + 0.5z^{-1}$. We call these filters *sine* and *cosine* filters because their respective frequency responses are:

$$H_{\sin}(e^{j\omega'}) = j\sin(\omega') \tag{3.131}$$

$$H_{\cos}(e^{j\omega'}) = \cos(\omega') \tag{3.132}$$

Their frequency response magnitudes, shown in Figure 3.32, illustrate the complementarity criterion. The sine filter has its zeroes were the cosine filter has its peaks and conversely.

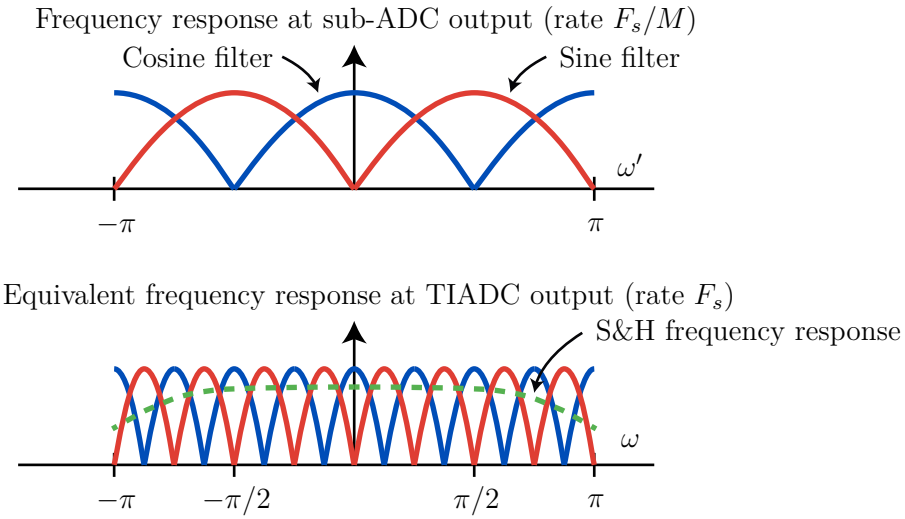


Figure 3.32 – Spectral interpretation of sine and cosine filtering ($M = 4$)

3.7.3.3 Theoretical explanation – WSS process

The general idea behind the bandwidth mismatch estimation technique is to measure the variance of each sub-ADC's output signal after cosine filtering and after sine filtering, and then calculate the ratio between those values (Figure 3.33). In order to more formally prove the idea, we first make the assumption that the input signal is a realization of a WSS random process. If the PSD of the TIADC input random process $\{X_t, t \in \mathbb{R}\}$ is $R_{XX}(\Omega)$

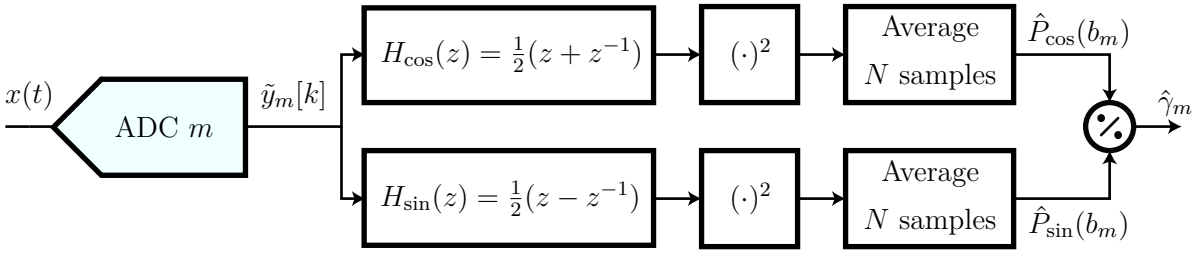


Figure 3.33 – Bandwidth mismatch estimation block diagram

then the PSD associated to the continuous input random processes $\{Y_t^m, t \in \mathbb{R}, m \in \mathcal{M}\}$ of each sub-ADC is

$$R_{YY}^m(\Omega) = \frac{R_{XX}(\Omega)}{1 + (\Omega\tau_m)^2} \tag{3.133}$$

where each sub-ADC's time-constant is of the form $\tau_m = \bar{\tau} + \Delta\tau_m$. If the signal is bandlimited to the first Nyquist region, the PSD corresponding to the discretization of the signal at rate F_s is naturally [12]

$$\breve{R}_{YY}^m(\omega) = \frac{\breve{R}_{XX}(\omega)}{1 + (\omega b_m)^2} \tag{3.134}$$

and the PSD of the processes $\{Y_t^m, t \in \mathbb{R}, m \in \mathcal{M}\}$ sampled at a rate F_s/M are aliased versions of the full rate PSD:

$$\breve{R}_{YY}^m(\omega') = \frac{1}{M} \sum_{k=0}^{M-1} \breve{R}_{YY}^m \left(\frac{\omega'}{M} - k\frac{2\pi}{M} \right) \tag{3.135}$$

The PSDs after sine and cosine filtering ($\breve{R}_{YY}^{m,\cos}(\omega')$ and $\breve{R}_{YY}^{m,\sin}(\omega')$ respectively) correspond sub-sampled PSDs (Equation 3.135) multiplied by the frequency response magnitude of the filters:

$$\breve{R}_{YY}^{m,\cos}(\omega') = \breve{R}_{YY}^m(\omega') \cos(\omega')^2 \tag{3.136}$$

$$\breve{R}_{YY}^{m,\sin}(\omega') = \breve{R}_{YY}^m(\omega') \sin(\omega')^2 \tag{3.137}$$

12. The symbol $\breve{}$ is here used significate that the spectrum is made periodic with period 2π (DT-PSD).

We are trying to extract the time-constant offsets from the sine-filtered and cosine-filtered variances. The variance (or power) $P_{\cos}(b_m)$ (respectively $P_{\sin}(b_m)$) can be obtained by integrating the PSD $\breve{R}_{YY}^{m,\cos}(\omega')$ (respectively $\breve{R}_{YY}^{m,\sin}(\omega')$) over the frequency interval $[-\pi, \pi]$.

$$P_{\cos}(b_m) = \int_{-\pi}^{\pi} \breve{R}_{YY}^{m}(\omega') \cos(\omega')^2 d\omega' \tag{3.138}$$

$$P_{\sin}(b_m) = \int_{-\pi}^{\pi} \breve{R}_{YY}^{m}(\omega') \sin(\omega')^2 d\omega' \tag{3.139}$$

which can be further developed by replacing the sub-sampled PSD by its expression:

$$P_{\cos}(b_m) = \frac{1}{M} \int_{-\pi}^{\pi} \breve{R}_{YY}^{m}(\omega) \cos(M\omega)^2 d\omega' \tag{3.140}$$

$$P_{\sin}(b_m) = \frac{1}{M} \int_{-\pi}^{\pi} \breve{R}_{YY}^{m}(\omega) \sin(M\omega)^2 d\omega' \tag{3.141}$$

If the normalized time-constant offsets $\{\Delta b_m, m \in \mathcal{M}\}$ are small as compared to the average normalized time-constant $\bar{b}$, the sine and cosine powers can be approximated by their 1$^{\text{st}}$-order Taylor approximation around the average normalized time-constant $\bar{b}$:

$$P_{\cos}(b_m) = P_{\cos}(\bar{b}) + \Delta b_m \frac{\partial P_{\cos}}{\partial b}\left(\bar{b}\right) \tag{3.142}$$

$$P_{\sin}(b_m) = P_{\sin}(\bar{b}) + \Delta b_m \frac{\partial P_{\sin}}{\partial b}\left(\bar{b}\right) \tag{3.143}$$

where $P_{\cos}(\bar{b})$ (respectively $P_{\sin}(\bar{b})$) is the power that would have the cosine-filtered (respectively sine-filtered) sub-ADC output signal if the S&H's normalized time-constant was $\bar{b}$:

$$P_{\cos}(\bar{b}) = \int_{-\pi}^{\pi} \frac{\breve{R}_{XX}(\omega)}{1 + (\omega\bar{b})^2} \cos(M\omega)^2 d\omega \tag{3.144}$$

$$P_{\sin}(\bar{b}) = \int_{-\pi}^{\pi} \frac{\breve{R}_{XX}(\omega)}{1 + (\omega\bar{b})^2} \sin(M\omega)^2 d\omega \tag{3.145}$$

As explained in the introduction of this section, making the bandwidth mismatch estimation immune to static gain mismatches is achieved by taking the ratio $\{\gamma_m, m \in M\}$ between the sine power $\{P_{\sin}^m, m \in \mathcal{M}\}$ and the cosine power $\{P_{\cos}^m, m \in \mathcal{M}\}$

$$\gamma_m = \frac{P_{\cos}(\bar{b}) + \Delta b_m \frac{\partial P_{\cos}}{\partial b}\left(\bar{b}\right)}{P_{\sin}(\bar{b}) + \Delta b_m \frac{\partial P_{\sin}}{\partial b}\left(\bar{b}\right)} \tag{3.146}$$

If the cosine and sine powers are non zero, rearranging Equation 3.146 leads to a bandwidth mismatch estimation formula that provides a relation between the normalized

time-constant offsets $\left\{\beta_m = \frac{\Delta b_m}{\bar{b}}, m \in \mathcal{M}\right\}$ and the ratios $\{\gamma_m, m \in \mathcal{M}\}$:

$$\beta_m = \frac{P_{\cos}(\bar{b}) - \gamma_m P_{\sin}(\bar{b})}{\gamma_m \bar{b} \frac{\partial P_{\sin}}{\partial b}\left(\bar{b}\right) - \bar{b} \frac{\partial P_{\cos}}{\partial b}\left(\bar{b}\right)} \tag{3.147}$$

The partial derivatives of the cosine power (respectively sine power) are obtained by differentiating Equation 3.140 (respectively Equation 3.141), and by swapping the derivative and the integral operators:

$$\bar{b}\frac{\partial P_{\cos}}{\partial b}\left(\bar{b}\right) = -2\int_{-\pi}^{\pi} \frac{\breve{R}_{XX}(\omega)\cos(M\omega)^2(\omega\bar{b})^2}{\left(1+(\omega\bar{b})^2\right)^2} d\omega \tag{3.148}$$

$$\bar{b}\frac{\partial P_{\sin}}{\partial b}\left(\bar{b}\right) = -2\int_{-\pi}^{\pi} \frac{\breve{R}_{XX}(\omega)\sin(M\omega)^2(\omega\bar{b})^2}{\left(1+(\omega\bar{b})^2\right)^2} d\omega \tag{3.149}$$

A closer look at these expressions indicates that the cosine (respectively sine) power derivative are equal to the power of the output signal of the bandwidth mismatch correction filter F_b, itself filtered by a cosine filter (respectively sine filter)

$$\bar{b}\frac{\partial P_{\cos}}{\partial \bar{b}}\left(\bar{b}\right) = -2\int_{-\pi}^{\pi} \breve{R}_{YY}(\omega)\cos(M\omega)^2 \left|F_b\left(e^{j\omega}\right)\right|^2 d\omega \tag{3.150}$$

$$\bar{b}\frac{\partial P_{\sin}}{\partial \bar{b}}\left(\bar{b}\right) = -2\int_{-\pi}^{\pi} \breve{R}_{YY}(\omega)\sin(M\omega)^2 \left|F_b\left(e^{j\omega}\right)\right|^2 d\omega \tag{3.151}$$

As we will see later, this equality makes it easy to calculate the value of the partial derivatives.

3.7.3.4 Theoretical explanation – non-stationary process

The analysis in the previous section remains valid when the input signal is a realization of a non-stationary random process. The reasons are similar to those mentioned for the gain and skew mismatch estimation algorithms. When the TIADC input signal is non-stationary, the variances of the cosine filtered and sine filtered sub-ADC output signals depend on the time (variable k):

$$P_{\cos}(b_m) = \mathrm{E}\left(\frac{1}{2}\left[\tilde{Y}^m_{k+1} + \tilde{Y}^m_{k-1}\right]^2\right) = r^m_{YY}[kM+m,0] + r^m_{YY}[kM+m,1] \tag{3.152}$$

$$P_{\sin}(b_m) = \mathrm{E}\left(\frac{1}{2}\left[\tilde{Y}^m_{k+1} - \tilde{Y}^m_{k-1}\right]^2\right) = r^m_{YY}[kM+m,0] - r^m_{YY}[kM+m,1] \tag{3.153}$$

where $r^m_{YY}[n,l]$ is the autocovariance function associated to sub-ADC m. As a consequence, the ratios $\{\gamma_m[k], m \in \mathcal{M}\}$ are also time-dependent. This time-dependence is not an issue for the estimation as long as the non-stationary components of $\{\gamma_m[k], m \in \mathcal{M}\}$

do not contain spurious tones at frequencies multiple of F_s/M. It means that the autocovariance function $r_{YY}(t,\tau)$ of the low-pass filtered input signal must not contain frequency components (with respect to the time variable t) that are multiple of F_s/M. This condition is true if the autocovariance function $r_{XX}(t,\tau)$ of the input signal before low-pass filtering also verifies the property. Indeed, the low-pass filtering operation does not add new frequency components to the signal, just attenuate those which already exist.

3.7.3.5 Circuit implementation

The bandwidth mismatch estimation formula in Equation 3.147 contains quantities that are replaced by their estimates in the circuit implementation of the algorithm.

Each power ratio $\{\gamma_m, m \in \mathcal{M}\}$ is replaced by its estimate $\{\hat{\gamma}_m, m \in M\}$ that is a ratio between a cosine power estimate $\left\{\hat{P}_{\cos}(b_m), m \in \mathcal{M}\right\}$ and a sine power estimate $\left\{\hat{P}_{\sin}(b_m), m \in \mathcal{M}\right\}$.

$$\hat{\gamma}_m = \frac{\hat{P}_{\cos}(b_m)}{\hat{P}_{\sin}(b_m)} \tag{3.154}$$

Each sine (respectively cosine) power estimate is obtained by calculating the mean square of each sine-filtered (respectively cosine-filtered) sub-ADC's output signal:

$$\hat{P}_{\cos}(b_m) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{1}{2}(\tilde{y}_m[k+1] + \tilde{y}_m[k-1]^2)\right]^2 \tag{3.155}$$

$$\hat{P}_{\sin}(b_m) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{1}{2}(\tilde{y}_m[k+1] - \tilde{y}_m[k-1]^2)\right]^2 \tag{3.156}$$

We do not know the *ideal* low-pass filtered signal because the sub-ADCs are corrupted by bandwidth mismatch. Therefore, calculating the ideal cosine filtered power estimate $\hat{P}_{\cos}(\bar{b})$ (respectively ideal sine filtered power estimate $\hat{P}_{\sin}(\bar{b})$) directly is not possible. However, if the number of sub-ADCs is large enough, one can consider that the time constants have a normal distribution around $\bar{b}$ such that the ideal cosine (respectively sine) filtered power estimate can be obtained by averaging the cosine filtered power estimates $\left\{\hat{P}_{\cos}(b_m)\right\}$ (respectively sine filtered power estimates $\left\{\hat{P}_{\sin}(b_m)\right\}$) across the sub-ADCs:

$$\hat{P}_{\cos}(\bar{b}) = \frac{1}{M}\sum_{k=0}^{M-1}\hat{P}_{\cos}(b_m) \tag{3.157}$$

$$\hat{P}_{\sin}(\bar{b}) = \frac{1}{M}\sum_{k=0}^{M-1}\hat{P}_{\sin}(b_m) \tag{3.158}$$

The sine and cosine power derivative estimates are obtained by replacing the signal term

by the output signal of the bandwidth correction filter in the above equation:

$$\bar{b}\frac{\partial \hat{P}_{\cos}}{\partial b}(\bar{b}) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{1}{2}(\tilde{y}'[k+1] + \tilde{y}'[k-1])\right]^2 \tag{3.159}$$

$$\bar{b}\frac{\partial \hat{P}_{\sin}}{\partial b}(\bar{b}) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{1}{2}(\tilde{y}'[k+1] - \tilde{y}'[k-1])\right]^2 \tag{3.160}$$

Replacing all the theoretical by their estimates yields the following bandwidth mismatch estimation formula:

$$\hat{\beta}_m = \frac{\hat{P}_{\cos}(\bar{b}) - \hat{\gamma}_m \hat{P}_{\sin}(\bar{b})}{\hat{\gamma}_m \bar{b}\frac{\partial \hat{P}_{\sin}}{\partial b}\left(\bar{b}\right) - \bar{b}\frac{\partial \hat{P}_{\cos}}{\partial b}\left(\bar{b}\right)} \tag{3.161}$$

In term of hardware cost, the sine and cosine filtered power estimations each require one simple 2-tap filter per sub-ADC, two multipliers to square the output of the filters and one adder to perform the average. The same amount of hardware is required for the calculation of the cosine and sine power derivative estimates. The division in Equation 3.154 and Equation 3.161 can be processed at low speed with one multiplier.

3.7.4 Accuracy analysis

This section provides an analysis of the bandwidth mismatch estimation accuracy. The bandwidth mismatch estimation algorithm was run on a narrow band signal (normalized bandwidth of 0.014) at different carrier frequencies, and the RMS level of the residual mismatches was measured and averaged over 10 Monte-Carlo simulations.

The residual mismatches are defined in term of the RMS error $\sigma_{\tilde{\beta}}$ between the normalized time-constant offset estimates and their true value.

$$\sigma_{\tilde{\beta}} = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1}\left(\hat{\beta}_m - \beta_m\right)^2} \tag{3.162}$$

This residual mismatch level is in turn expressed as a function of the SNDR with a sine input using (Equation 2.81). Figure 3.35 shows the variation of the SNDR as a function of the average length N.

The plot in Figure 3.34 corresponds to the case where the RMS level of the bandwidth mismatch was relatively high ($\sigma_\beta = 5\%$), and we see that the estimation accuracy does not improve anymore above a certain average length. This limitation arises when the estimation error becomes limited by the second-order terms. This phenomenon is further emphasized when looking at Figure 3.35a where the initial mismatch level is smaller ($\sigma_\beta = 1\%$). In this case, the attainable SNDR is much higher, and it is reached for a

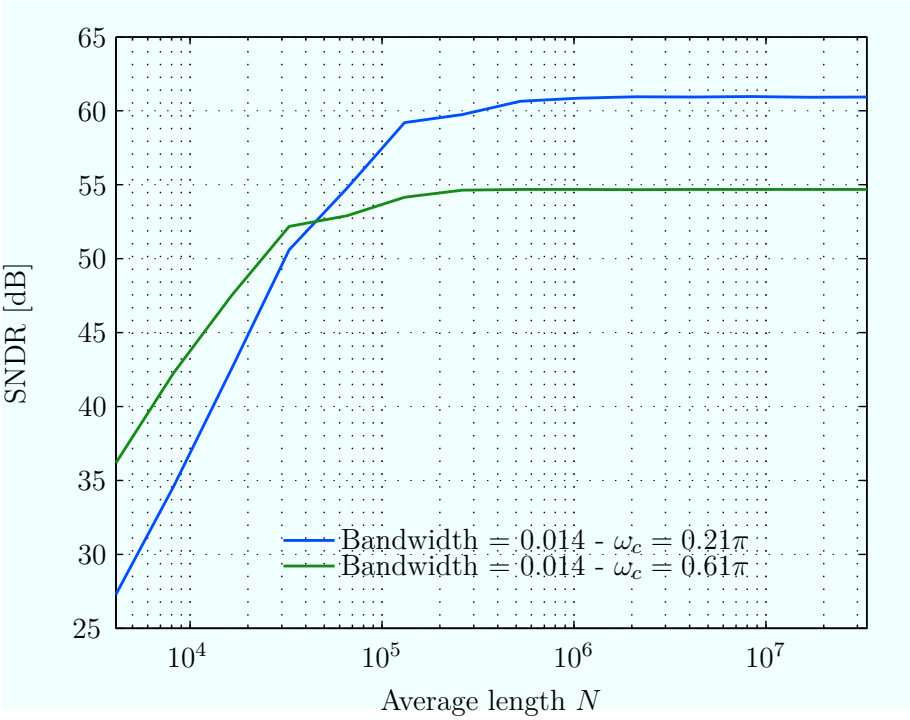
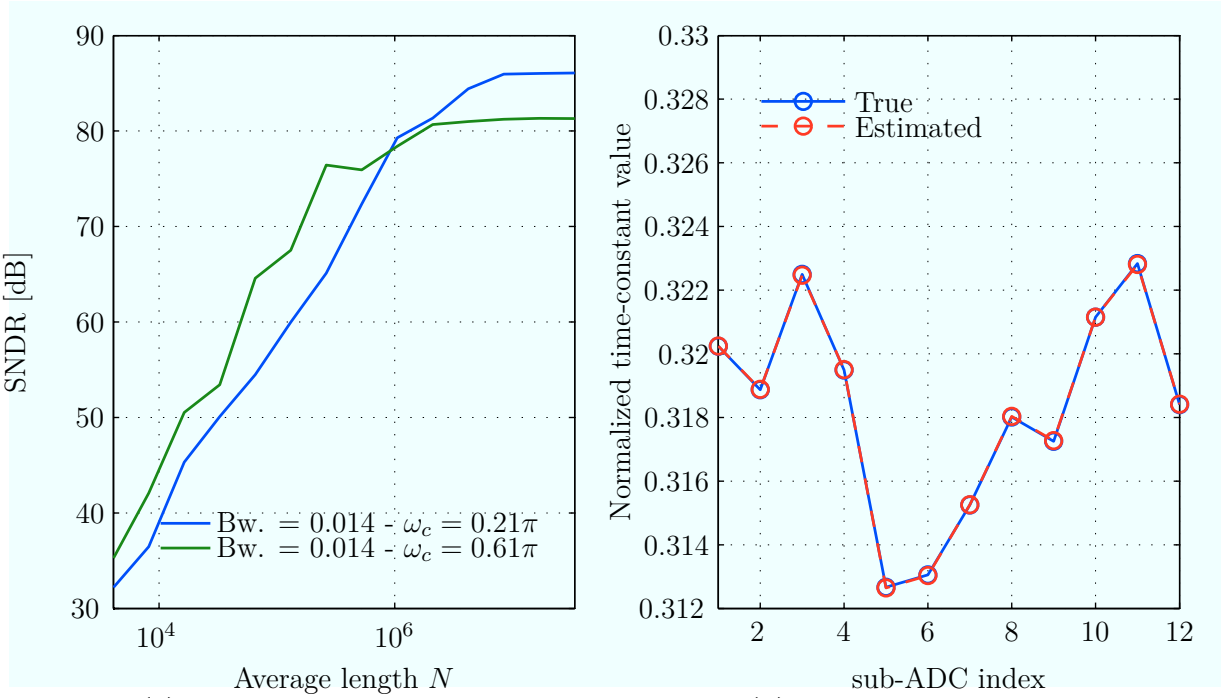


Figure 3.34 – Bandwidth mismatch estimation accuracy as a function of average length, and for different modulated signals ($M = 12$, $\sigma_\beta = 5\%$, sine input frequency $\omega_0 = 0.8\pi$ residual mismatches averaged over 10 Monte-Carlo simulations, no quantization)

bigger value of N. The plot in Figure 3.35b gives an example of the estimated time-constant offsets values when the bandwidth mismatch estimation has converged.

3.7.5 Conclusion

This section addressed the problem of first-order bandwidth mismatch calibration. In the proposed technique, both the bandwidth mismatch estimation and the bandwidth mismatch correction take advantage of the fact that the time-constant offsets are small as compared to the nominal time-constant of the S&H low pass filter. In this linear regime, the bandwidth mismatch correction can be done using a FIR filter with fixed coefficients, and the bandwidth mismatches are estimated by linearizing the mean-square of a sine-filtered version and a cosine-filtered version of each sub-ADC's output signal. Importantly, the bandwidth mismatch estimation is insensitive to static gain mismatches, and converges for non-stationary signals with only small restrictions.



(a) SNDR vs. average length

(b) sub-ADC time-constants

Figure 3.35 – Bandwidth mismatch estimation accuracy as a function of average length, and for different modulated signals ($M = 12$, $\sigma_\beta = 1\%$, sine input frequency $\omega_0 = 0.8\pi$, residual mismatches averaged over 10 Monte-Carlo simulations, no quantization)

3.8 Conclusion

The problem of TIADC blind background mismatch calibration has been a widely studied topic in the literature, and numerous solutions have been proposed to alleviate the effects of offset, gain, timing and bandwidth mismatches. This chapter adds two major contributions to the state of the art.

First, making the assumption that the mismatches are small enables to linearize the mismatch error models, and devise mismatch estimation algorithms that converge in a single iteration. This approach greatly reduces the estimation time as compared to adaptive calibration techniques. To the best of my knowledge, state of the art publications on TIADC digital mismatch calibration all use adaptive architectures with feedback loops. Adaptive techniques potentially have instability issues when the input signal is not known. The algorithms proposed in this chapter are guaranteed not to stay in a non-converging state for an undetermined amount of time in case a bad input signal situation happens (absence of signal for example).

Second, the input signal conditions on which blind calibration relies were analyzed in the context of wideband communication systems. The proposed estimation techniques converge even if the input signal is non-stationary as long as certain spectral properties of its autocovariance function are satisfied. Specifically, it is required that the non-stationary part of the input signal autocovariance does not have spurious tones at frequencies multiple of F_s/M. These findings widen the generally admitted but restricting assumption that the input signal must be WSS, which is important because WSS processes often do not accurately describe communication signals. We showed that pulse-shaped modulated signals exhibits properties that make them suitable for blind mismatch estimation, and further work needs to be done to study a wider class of inputs such as OFDM signals.

One of the goals of this chapter was to emphasize the structural similarities between the different calibration stages, particularly between skew mismatch calibration and bandwidth mismatch calibration. It would be interesting to see other types of mismatches (n-th order bandwidth mismatches for example) treated using the same type of framework.

Chapter 4

Digital mismatch calibration circuit implementation

4.1 Introduction

The previous chapter gave the theoretical foundations of the mismatch algorithms as well as analysis and simulations that proved their performance. This chapter presents a circuit implementation that embeds offset, gain and skew mismatch calibration [13]. Bandwidth mismatch calibration was not required in this implementation because the circuit to calibrate was not foreseen to exhibit significant bandwidth mismatch impairments.

The ADC in this implementation is a 1.62 GS/s TIADC comprised of 12 SAR sub-ADCs. After mismatch calibration, it achieves mismatch spurs below -70 dBFS up to 750 MHz of input frequency.

4.2 Overall architecture

The overall architecture of the circuit is shown in Figure 4.1. It consists of a TIADC, the "analog" front-end, followed by the digital mismatch calibration. The mismatch calibration takes the 9-bit output samples of the TIADC and outputs 12-bit samples. The calibration runs in the background and there is no feedback loop either toward the analog-front-end or inside the digital unit itself.

The chip, which photo is shown in Figure 4.2, was designed in ST 40 nm CMOS technology, and contains two copies of the same TIADC and two copies of the digital mismatch calibration unit. It has a total area of 1.66 mm^2 with 0.70 mm^2 (42%) dedicated to the digital mismatch calibration and 0.96 mm^2 (58%) to the analog front-end.

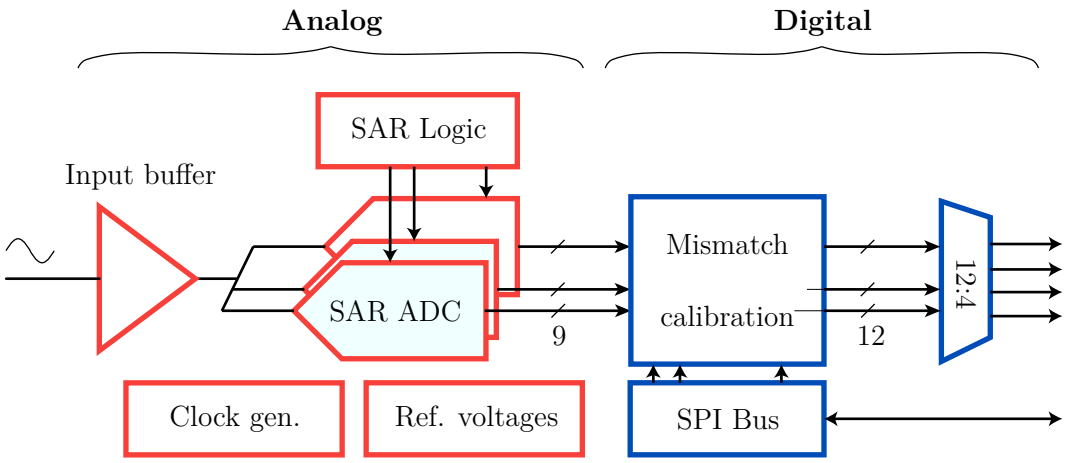


Figure 4.1 – Overall architecture of the circuit

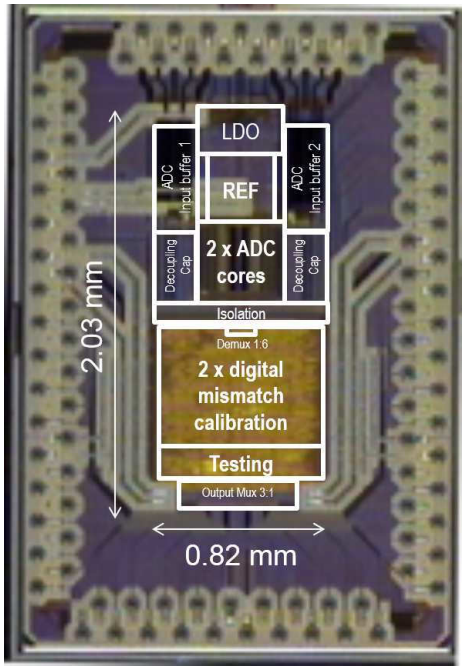


Figure 4.2 – Die photo of the chip comprised of 2 copies of the 12-way TIADC and 2 copies of the digital mismatch calibration unit

4.2.1 Analog front-end

The TIADC to calibrate is comprised of 12 SAR sub-ADCs, each having a sampling frequency of 135 MS/s. All the sub-ADCs are driven by a 1.7 V input buffer that delivers a differential input signal with a maximum amplitude of 1 V peak-to-peak. The input buffer is a simple follower that isolates the signal source from the kickback noise generated by the TIADC operation.

The sub-ADCs are radix-2 SAR ADCs (Figure 4.3), meaning that, during the quantization, the input signal is successively compared to voltage levels that are ratios of two of a reference voltage.

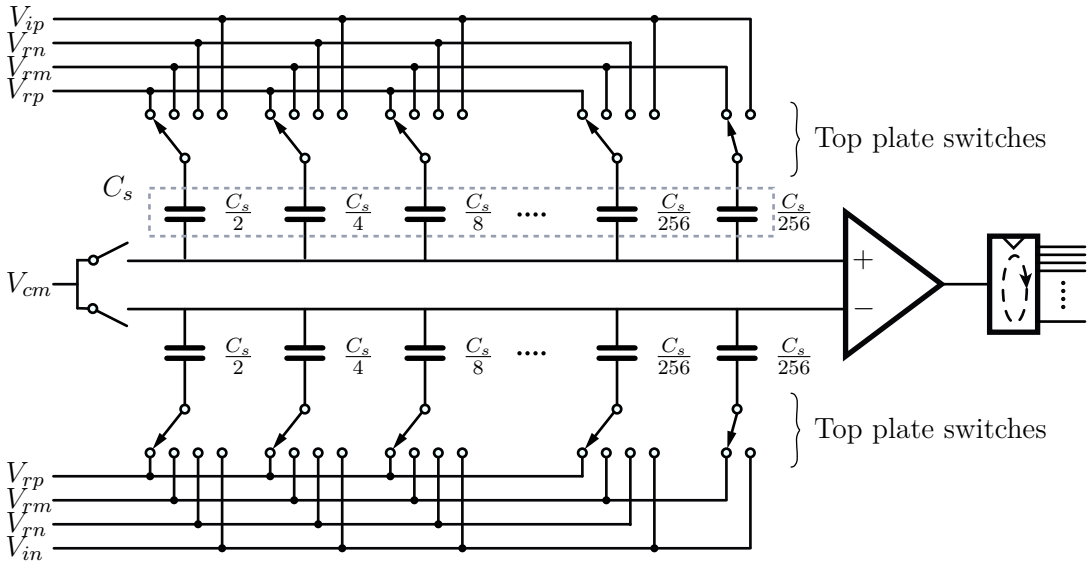


Figure 4.3 – Sub-ADC architecture

The input signal is sampled individually by each of the SAR ADCs, and the signal is held on a sampling capacitor, comprised of a bank of binary weighted capacitors. This capacitor array has a custom interdigitated structure as shown in Figure 4.4. The top plates of the capacitors are metal fingers (two layers) that are inserted into a comb structure that represents the common bottom plate of the capacitors. The (lateral) capacitance of each capacitor is determined by its number of fingers. The specificity of this design is that the unit capacitor $C_s/512$ is made of a quarter finger and the capacitor $C_s/256$ is made of a half finger. This custom layout structure ensures good matching between the capacitors (and therefore a good ADC linearity), and the small size of the overall structure helps reduce the wiring to the capacitors.

The signal is sampled using bottom plate sampling in order to reduce signal-dependent charge injection on the sampling capacitor. During the tracking phase, the input signal is tracked on the top plate of the sampling capacitor (top plate switch closed) and the

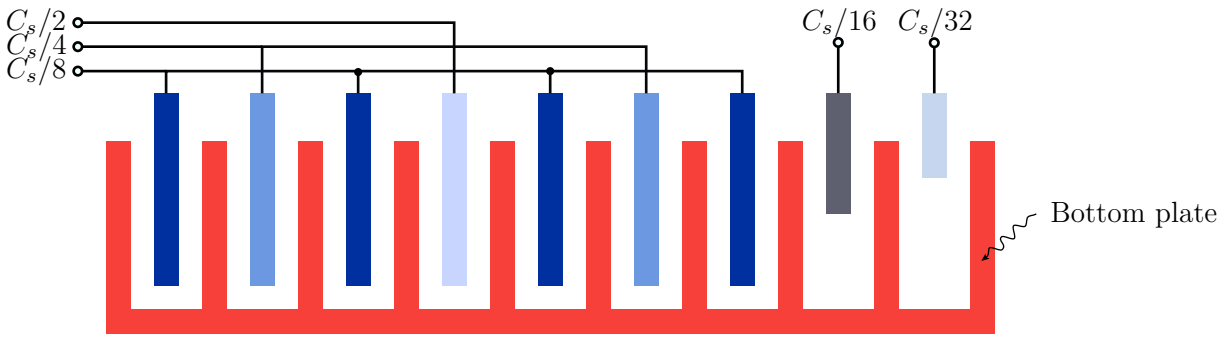


Figure 4.4 – Custom layout of the binary weighted capacitor array

bottom plate is connected to a fixed-voltage $V_{cm} = 700$ mV (bottom plate switch closed). The sampling is done in two phases (Figure 4.5). First, the bottom plate switch is opened,

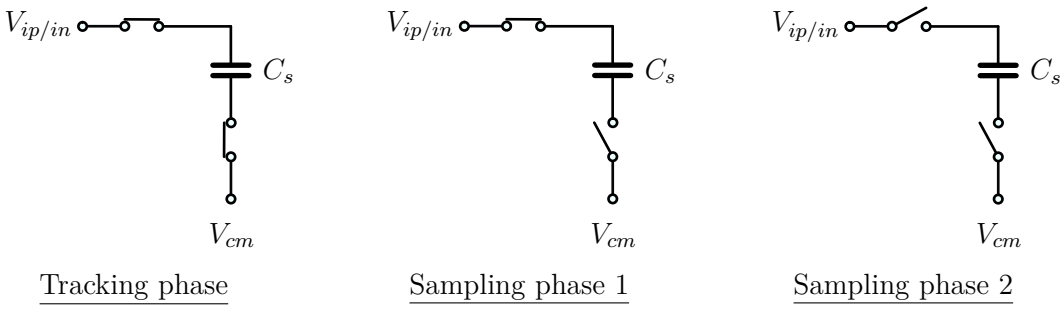


Figure 4.5 – Bottom plate sampling principle

which prevents the charge stored on the sampling capacitor to change. Second, the top plate switch is opened in order to disconnect the input signal from the top plates of the capacitors. When the top plate switch is opened, no charge injection can occur because the bottom plates of the capacitors are in a high impedance state. The only time when charge injection happens is when the bottom plate switch is opened. However, the charge that is injected at that moment is not signal dependent, thereby only yielding a fixed offset that gets cancelled out in differential operation.

The top plate switches are transmission gates made of back-to-back PMOS and NMOS transistors to improves linearity. Indeed, as shown in Chapter 2, the resistance of a MOS transistor depends on its gate-to-source voltage. That is why switches made of MOS transistors have a signal-dependent on-resistance. Using a transmission gate reduces this effect because the resistance of the PMOS, and the resistance of the NMOS vary in opposite direction when the input signal amplitude varies.

The quantization process is done using the Merge Capacitive Switching (MCS) scheme [15]. The signal is successively compared to different voltage levels generated by the capacitive DAC. In the MCS scheme, the top plates of the capacitors are initially all connected to a middle voltage ($V_{rm} = 250$ mV). During this initial state, the differential voltage at the

input of the comparator is $V_i = V_{ip} - V_{in}$ and the signal is therefore compared to 0 V. Depending on the value of the MSB (1 if $V_i > 0$ and 0 if $V_i < 0$), the bottom plates of the two MSB capacitors are either connected to a positive reference voltage $V_{rp} = 750$ mV, or a negative reference voltage $V_{rn} = 250$ mV, such that the differential signal at the input of the comparator is either $V_i + (V_{rp} - V_{rn})/2$ or $V_i - (V_{rp} - V_{rn})/2$. This operation is managed by the SAR logic and is repeated until each bit is resolved. The fully differential comparator is comprised of a two-stage pre-amplifier followed by a latch comparator.

Of course, the design of the TIADC was done carefully in order to minimize the amount of mismatches but no additional analog circuitry was used for calibration. The mismatches are compensated by the digital mismatch calibration unit described in the next section.

4.2.2 Mismatch calibration

The digital mismatch calibration unit takes the 12 streams of 9-bit samples coming from the sub-ADCs in order reduce the mismatches. It is driven by a 135 MHz digital clock and has it own 1.1 V supply.

The samples are passed through three calibration stages as shown in Figure 4.6. First,

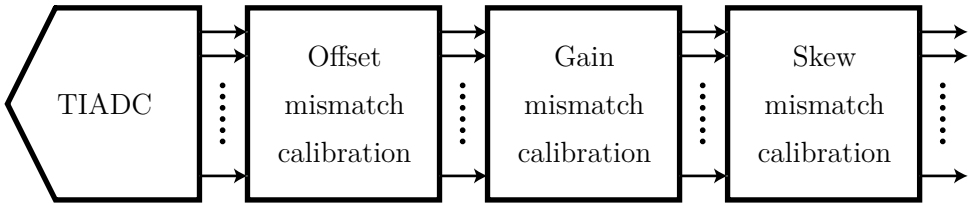


Figure 4.6 – Mismatch calibration structure

offset mismatch calibration is performed. In order to reduce the post-calibration residual offsets, the samples at the output of the offset calibration stage are quantized to 12 bits. Digital dithering as described in Chapter 3 is also performed, which enables to whiten the residual offset mismatch noise.

The 12 streams of 12-bit samples coming out of the offset mismatch calibration block are then transmitted to the gain mismatch calibration unit described in 3.

After gain mismatch calibration, the samples are passed through the skew mismatch calibration stage before being multiplexed at the output of the chip in 4 streams of 12-bit samples. The derivative of the signal is calculated using a FIR filter of 32 taps that has an accurate frequency response up to 700 MHz. The choice to set the cut-off frequency of the derivative filter to 700 MHz was governed by the product specifications that insured that the input signal had no frequency content above this frequency. The calibration blocks are designed such that the average length during the estimation phase can be programmed

to up to 2^{30} samples per sub-ADC. This average length represents an estimation time of about 8 seconds. In practice however, the estimation length is set to 2^{25} samples per sub-ADC, which corresponds to an estimation time of 250 ms .

One of the greatest advantages of digital mismatch calibration as compared to mixed mismatch calibration is the ability to use an automated digital design flow. In order to further increase this advantage, High Level Synthesis (HLS) was used to generate the calibration unit. The goal was to develop a flow that was easily reusable and verifiable for any TIADC architecture.

The design steps that we followed are illustrated in Figure 4.7. First, the mismatch

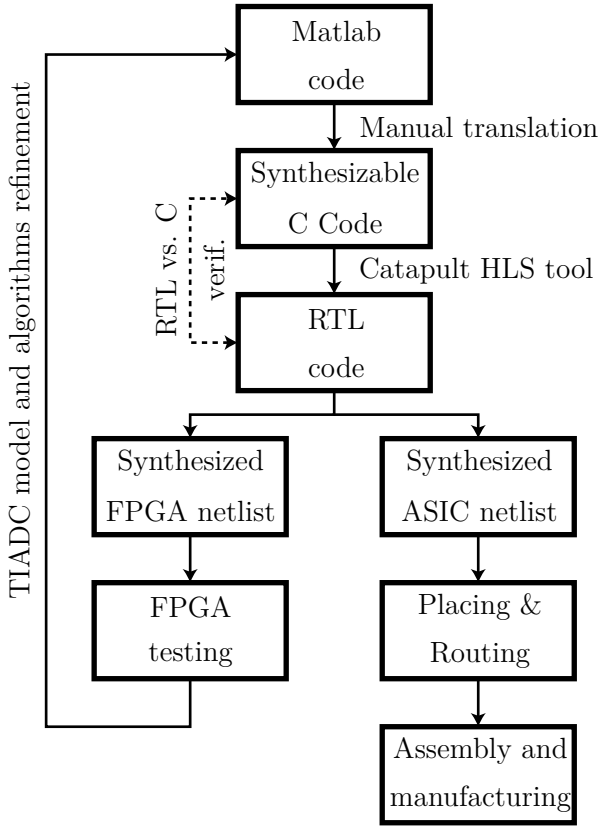


Figure 4.7 – Mismatch calibration digital flow

calibration algorithms were entirely simulated in Matlab in order to optimize the parameters associated to each block (average length, number of filter taps, quantization). Doing so enables to benefits from the simulation speed of Matlab, as well as its powerful visualization tools.

Then, the calibration algorithms were implemented in fixed-point synthesizable C code and executed on CPU-based machines in order to verify that the performance was similar to the one obtained in Matlab. The C code was written with a high degree of parametrization (number of ADCs, level of mismatches, level of quantization), which

makes it easy to adapt to any TIADC architecture. The power of HLS tools, such as Catapult, lies in their ability to generate optimized Register Transfer Level (RTL) code that can be targeted either to FPGAs or to ASICs.

We used that versatility by first implementing the calibration unit in a FPGA that we connected to an existing TIADC to verify its functionality. This validation phase was critical to refine the initial Matlab model, and the calibration algorithms. For example, the FPGA implementation helped us tune the bit-width of the different signals inside the calibration blocks.

The last development stage was to synthesize the RTL code for the ASIC. We followed the classical steps of digital design (synthesis, placing and routing, verification).

4.3 Measurements

The section provides measurement results of the chip. The focus is put on the mismatch calibration performance but the usual performance indicators are also provided (SNDR and SFDR vs. frequency, Figure of Merit (FOM), linearity), and the overall TIADC performance is compared to the state of the art.

4.3.1 Measurement test bench

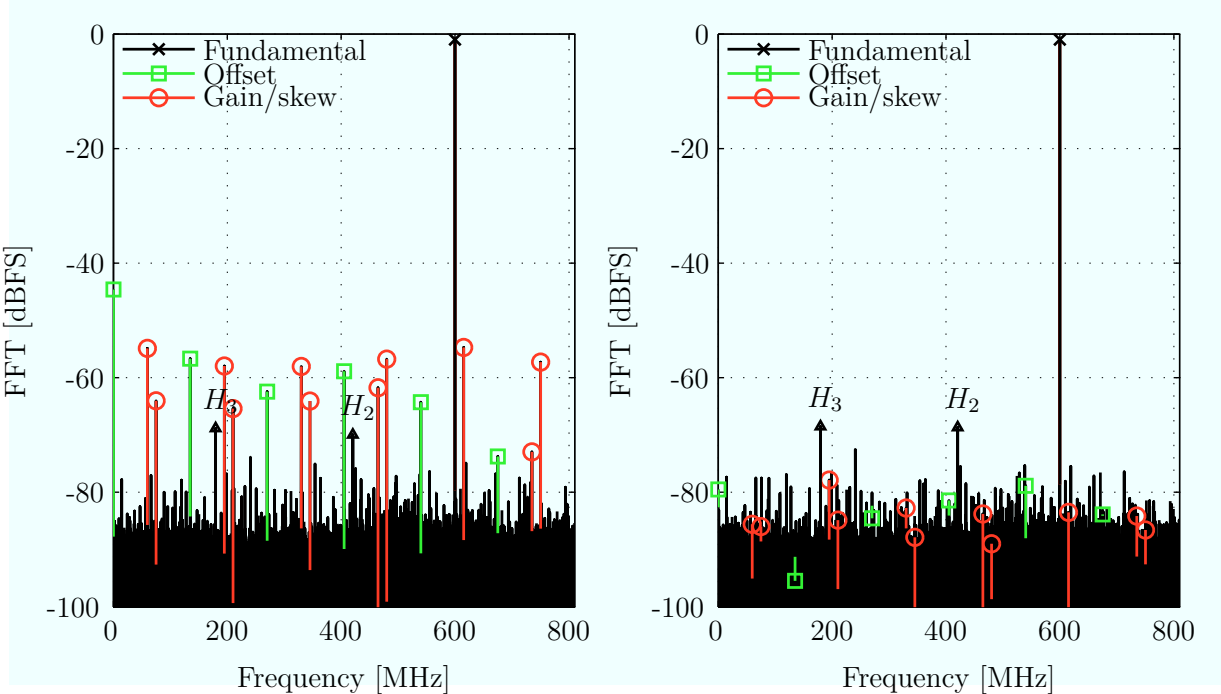
The chip is mounted on a PCB and supplied by 1.1 V and 1.7 V supplies. The 1.62 GHz clock is generated by a sine wave generator and converted into a square waveform before being provided single-ended at the input of the chip. Single or multi-tone signals are generated by a vector signal generator, and more complicated waveforms such as modulated signals are generated by an Arbitrary Waveform Generator (AWG). Single-tone signals are also low-pass or band-pass filtered in order to suppress the harmonics coming from the generator non-linearities. A high bandwidth splitter generates the 0° and 180° signal phases that are provided to the chip.

The digital outputs of the TIADC (12×12 bits divided into 4 groups) are buffered and acquired at a rate of 400 MHz by the Logic Analyzer. The digital samples are then transmitted and processed on a PC. The PC also controls the signal generators (frequency, amplitude) such that the whole testing can be performed through this unique interface.

The calibration can be by-passed by an on-board switch (hard by-pass), or finely controlled through a SPI interface also controlled from the PC. The SPI control enables to turn on or off the different calibration blocks and modify parameters such as the average length of each estimation. The values of the estimated mismatches is stored in internal registers and can also be accessed through the SPI interface.

4.3.2 Mismatch calibration performance

Figure 4.8 illustrates the performance of the mismatch calibration by showing the TIADC output spectrum before (Figure 4.8a) and after calibration (Figure 4.8b) with a sine input signal 1 dB below full scale (-1 dBFS). Before mismatch calibration, the mismatch tones are as high as -55 dBFS, whereas after calibration they are reduced to a level around -80 dBFS that is not distinguishable from the noise floor. Only harmonics due the non-linearity of the TIADC remain.



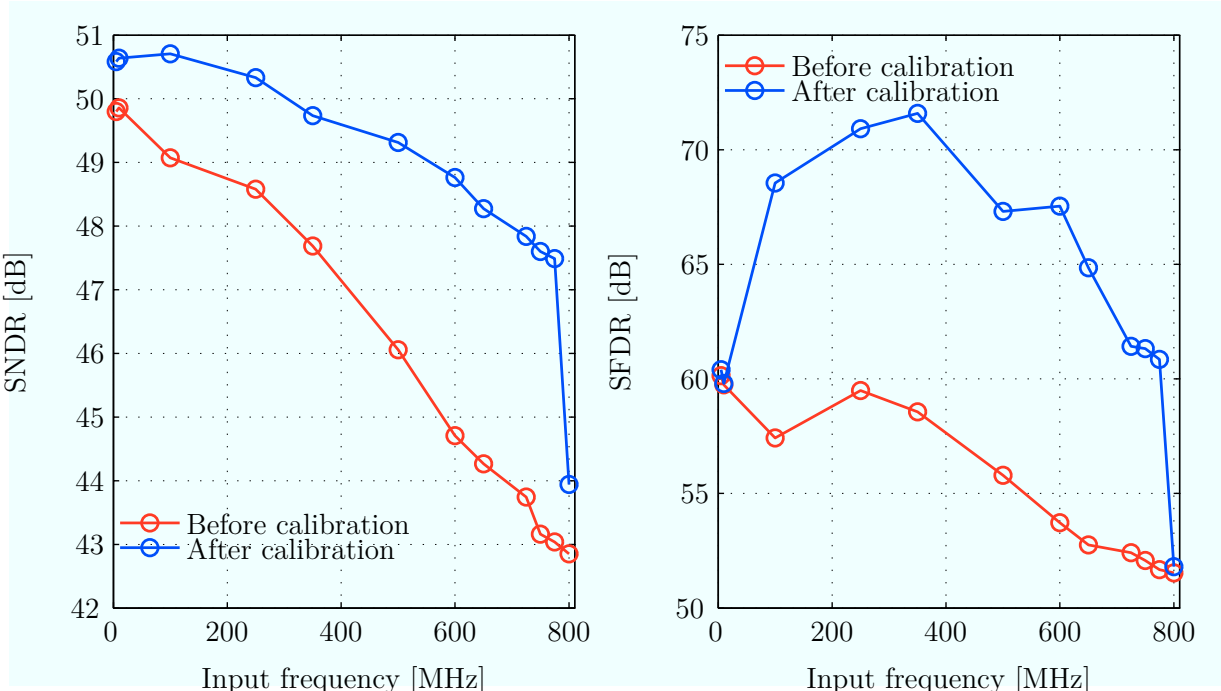
(a) Before calibration **(b)** After calibration

Figure 4.8 – Output spectrum before and after calibration with a -1 dBFS 600 MHz sine input ($N_{\text{FFT}} = 12 \times 4096$)

The plots in Figure 4.9 are obtained by sweeping the input signal frequency and by measuring the SNDR and the SFDR at each frequency. Before calibration, the SNDR decreases with increasing input frequency because the skew mismatch effects degrade the signal more at high frequency. After calibration, the SNDR still decreases with increasing input frequency but it stays above 47 dB almost up to the Nyquist frequency. This increase in SNDR represents almost 1 additional effective bit of resolution as compared to the non-calibrated case. A performance drop can be seen closer to the Nyquist frequency because of the derivative filter frequency response starts to roll down above 700 MHz.

Before calibration, the SFDR also decreases with increasing input frequency and is mainly affected by the mismatch tones. This effect is shown in Figure 4.10a and in Figure 4.10b that represent the level of the offset and skew/gain mismatch tones as a

function of the input signal frequency. After calibration, the SFDR stays above 60 dB up to 750 MHz, with a peak at 72 dB close to 400 MHz. However, the SFDR plot does not completely represent the performance of the calibration because the calculation of the SFDR also takes into account the harmonics of the signal. These harmonics are mostly caused by the non-linearity of the input buffer that drives the sub-ADCs. The level of the mismatch tones after calibration is a more relevant indicator to assess the calibration performance. The plot in Figure 4.10a shows that the offset mismatch tones stay at a level below -75 dBFS across the entire spectrum, whereas the plot in Figure 4.10b) indicates that the skew and gain mismatch tones are below -70 dBFS up to 750 MHz of input frequency.

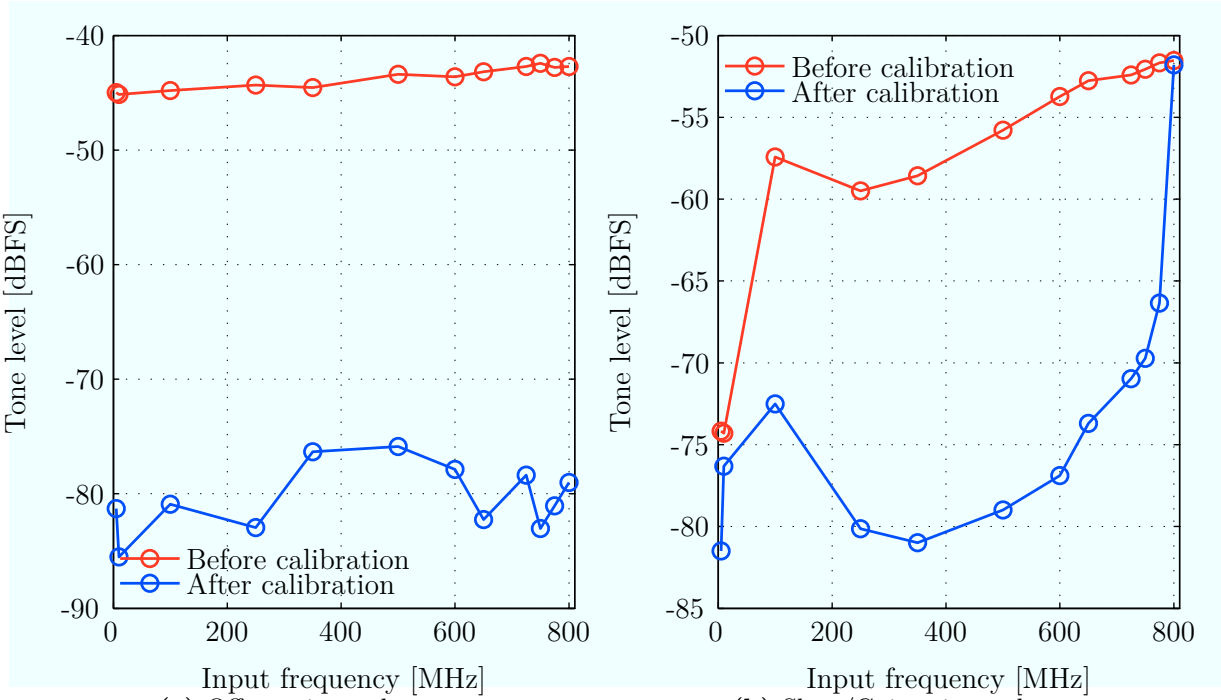


(a) SNDR **(b)** SFDR

Figure 4.9 – Measurement of SNDR and SFDR as a function of input frequency with and without calibration

4.3.3 Mismatch characterization

Another advantage of non-adaptive mismatch estimation is that it provides precise values for the offset, gain and timing offset estimated for each sub-ADC. This is very useful has a feedback to the analog design team. By measuring the signature of the mismatches from chip to chip, one can look for deterministic mismatch patterns caused, for example, by a layout mistake.



(a) Offset mismatch tones **(b)** Skew/Gain mismatch tones

Figure 4.10 – Measurement of highest mismatch tones as a function of the input frequency

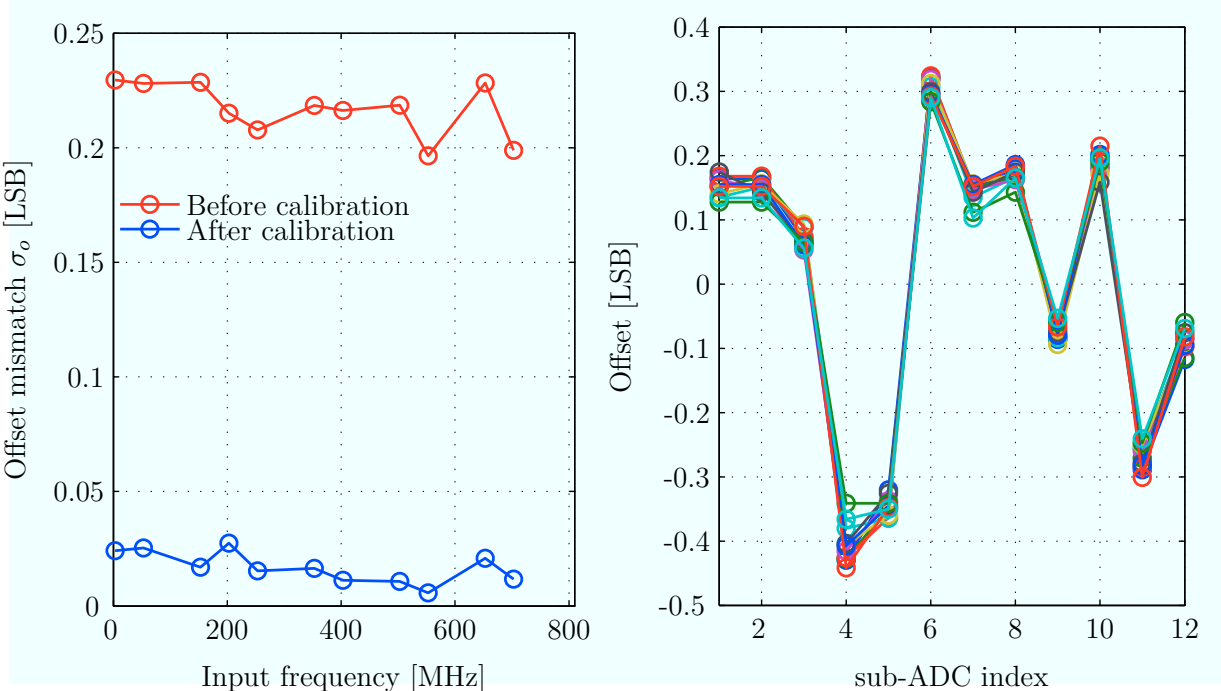
It also allows to look at the mismatch variations as a function of the input frequency and across different temperatures.

4.3.3.1 Mismatch variations vs. input frequency

Figures 4.11a, 4.12a and 4.13a show the mismatch levels before and after calibration as functions of the input frequency.

The offset mismatches are measured in terms of the RMS offset mismatch level σ_o. The RMS offset mismatch level is determined by inverting the "offset mismatch SNDR" expression given in Chapter 2. This is done by only considering the offset mismatch tones in the calculation of the SNDR. Theoretically, the offset mismatches should not vary with the input frequency but the plot in Figure 4.11a shows slights variations around a RMS level of 0.23 LSB. This might be due to measurement imprecisions (because the input signal is a sinusoid for example), or to analog effects such as signal feedthrough in the sampling switch. After calibration, the offset mismatches are kept under a RMS level of 0.03 LSB [1]. The distribution of the estimated offsets across the sub-ADCs is shown in Figure 4.11b.

1. The residual offset mismatches might even be lower because after offset calibration most of the offset mismatch tones are not distinguishable from the noise floor

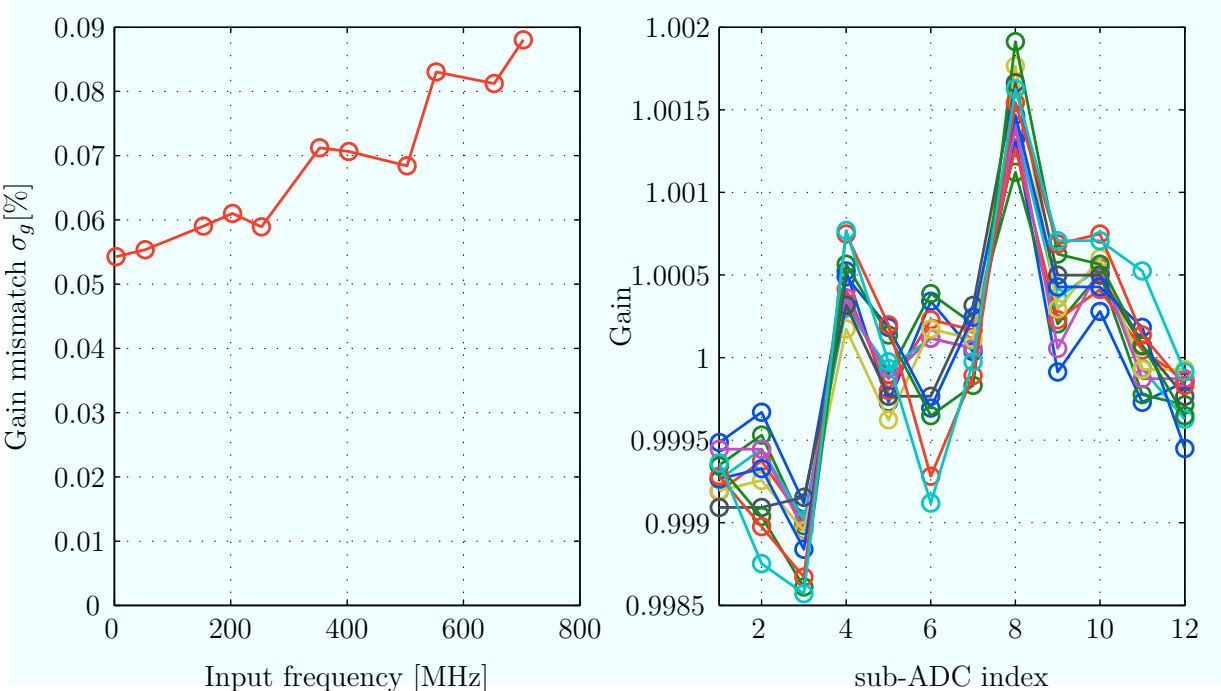


(a) RMS offset mismatch **(b)** sub-ADC offset distribution

Figure 4.11 – Offset mismatch measurements as a function of the input frequency

Since the gain mismatch tones are not distinguishable from the skew mismatch tones, the gain mismatch level was measured just by looking at the estimated gains provided by the calibration. The gain mismatch RMS value σ_g, defined in 2, is used to measure the level of the gain mismatches. The plot in Figure 4.12a shows that the gain mismatch level is low (below 0.1%) for this particular circuit implementation. However, the gain mismatch level increases at high input frequency most likely due to small bandwidth mismatches between the sub-ADC samplers. The plots in Figure 4.12b shows the distribution of the gains across the sub-ADCs for different input frequencies. As compared to the offset distribution, the estimated gain distribution is more "noisy". Again, this inconsistency most likely stem from frequency dependent gain mismatches that get added to the static gain mismatches. Given the small gain mismatch level, it is also possible that, the gain estimation algorithm provides slightly imprecise gain estimates.

Based on these observations, the effect of gain mismatches on the skew mismatch spurious tones can be considered negligible, especially at high input frequency where the skew mismatches deteriorate more the signal. Therefore, the skew mismatch level can be measured by inverting the "skew mismatch SNDR" equation derived in Chapter 2. The noise level is calculated by adding together each skew mismatch tone's power whereas the signal power is obtained by measuring the power of the fundamental. This calculation leads to the plot in Figure 4.13a that represents the RMS skew mismatch level $\sigma_{\delta t} = \sigma_r \times T_s$



(a) RMS gain mismatch **(b)** sub-ADC gain distribution

Figure 4.12 – Gain mismatch measurements as a function of the input frequency

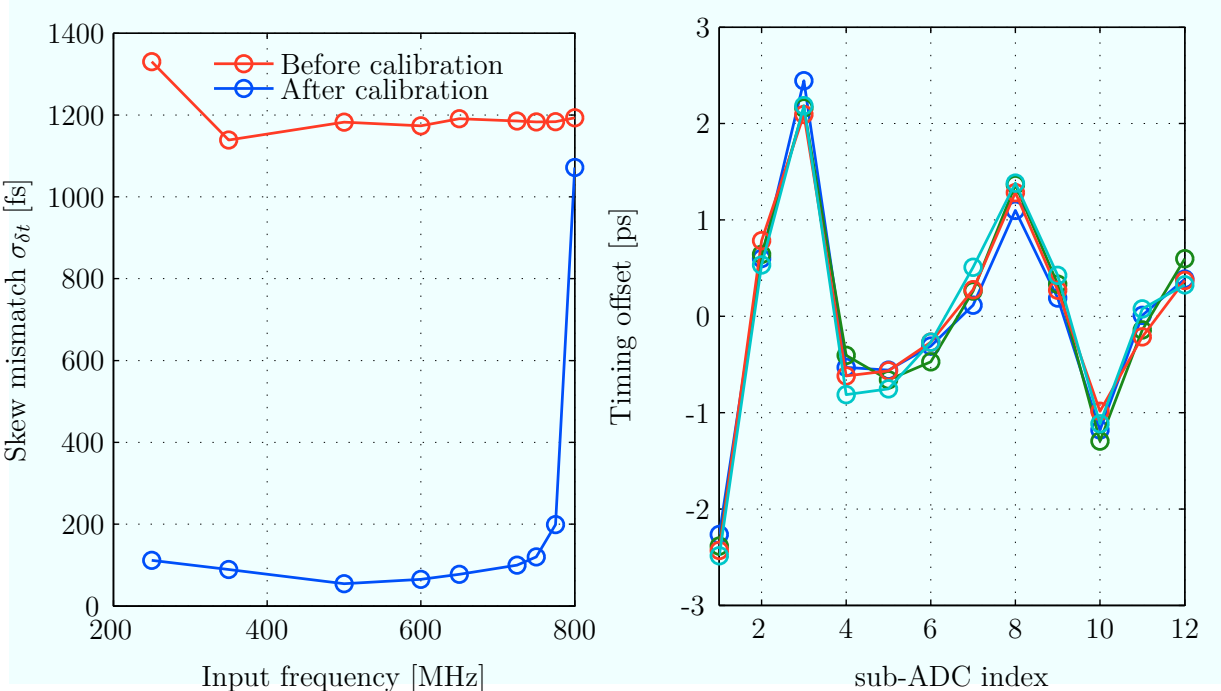
as a function of the input signal frequency. The plot only shows the skew mismatch level above 200 MHz in order to make sure that the gain mismatches have a negligible contribution to the level of the mismatch tones. Before calibration, the skew mismatch level is around 1.2 ps whereas after calibration it stays below 100 fs up to an input signal frequency of 700 MHz. Past 700 MHz, the skew mismatch level after calibration increases because the signal is reaching the cut-off frequency of the derivative filter. The plot in Figure 4.13b shows the timing offset distribution across the sub-ADCs for different input frequencies [2].

4.3.3.2 Mismatch variations vs. temperature

One of the main advantage of background mismatch calibration is its ability to track mismatch changes caused by temperature variations. In order to quantify this effect, the chip was tested in a temperature-controlled environment, and we measured the mismatch levels while stressing the chip with different temperatures. The results of these measurements are displayed in Figures 4.14, 4.15 and 4.16.

As shown in Figure 4.14a, the offset mismatch level barely varies with temperature, spanning a RMS level range from 0.222 LSB at 50°C to 0.24 LSB at 50°C. Therefore, if

2. The data points close to the Nyquist frequency have been removed because the estimated values make little sense in that frequency region



(a) RMS skew mismatch

(b) sub-ADC timing offset distribution

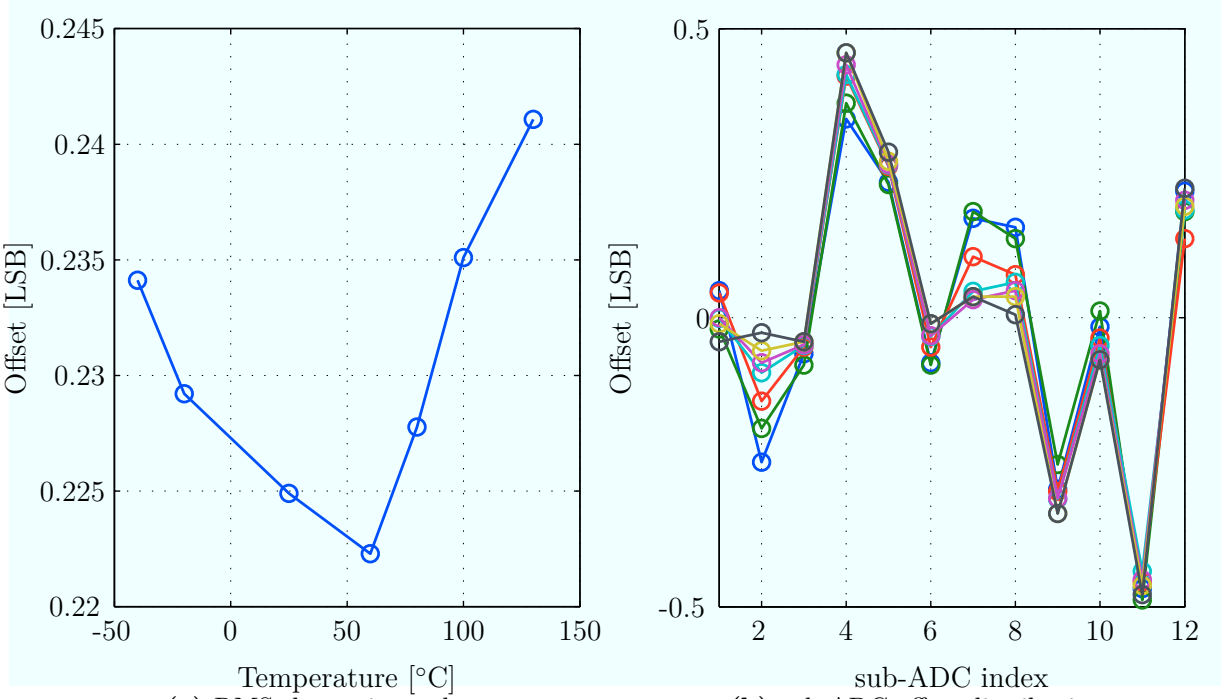
Figure 4.13 – Skew mismatch measurements as a function of the input frequency

the offset mismatch calibration was not running continuously, the RMS level difference would be at most 0.02 LSB from one extreme temperature case to another. This does not seem sufficient to justify the use of background calibration because the error is of the same order as the residual offset mismatches after calibration. However, a close look at Figure 4.14b shows that the offset distribution across the sub-ADCs changes slightly with temperature, even though the RMS value is not affected much. Therefore, estimating the offsets of the sub-ADCs at one temperature, and using the same set of estimated value to correct the TIADC at another temperature would increase the level of the offset mismatch tones.

Figure 4.15a shows that the effects of temperature on the gain mismatches is a little more clear. The plot shows an increase in gain mismatch level with temperature but the increase is very small relatively to the original gain mismatch level. In this chip, the gain mismatch level was very small to begin with so the temperature effects are almost completely negligible. The distribution of the gains across the different sub-ADCs also remains relatively constant across temperature as shown in Figure 4.15b.

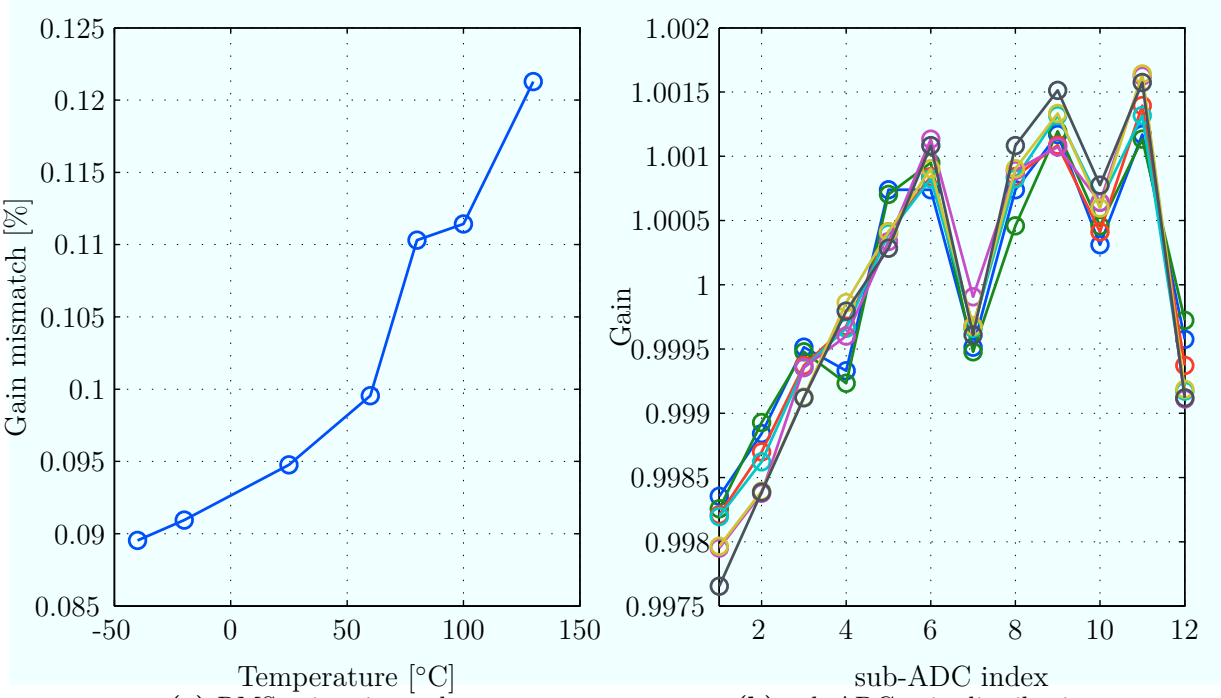
The variation of the skew mismatch level as a function of the temperature is shown in Figure 4.16a. The skew mismatch level first decreases with temperature, with a minimum around 25°C, before increasing again [3]. As illustrated in Figure 4.16b, the distribution of

3. It is not clear whether the data point at 100°C is a measurement error or a real phenomenon



(a) RMS skew mismatch **(b)** sub-ADC offset distribution

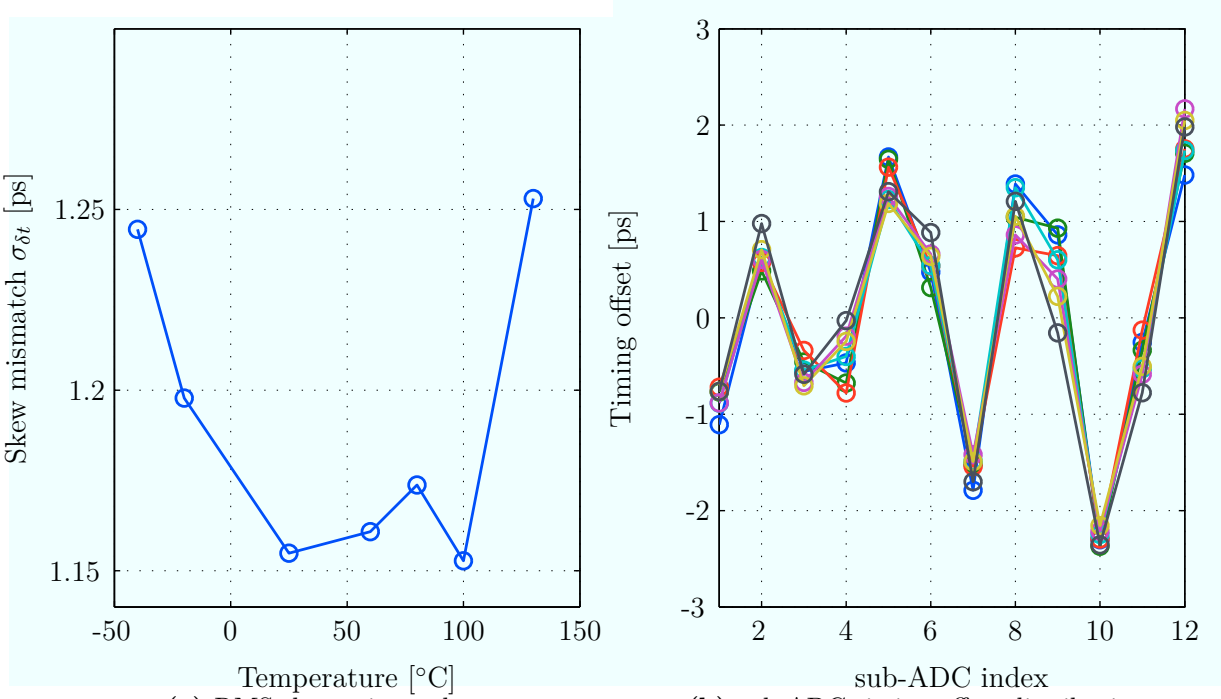
Figure 4.14 – Offset mismatch measurements as a function of temperature



(a) RMS gain mismatch **(b)** sub-ADC gain distribution

Figure 4.15 – Gain mismatch measurements as a function of temperature

the timing offsets across the sub-ADCs does not change much with temperature, However, some sub-ADC timing offsets vary significantly from one temperature to another. For example, the estimated timing offset of sub-ADC 9 changes by about 1 ps across the temperature range. This type of local variation reinforces the need for background mismatch calibration to track the mismatch variations due to temperature.



(a) RMS skew mismatch **(b)** sub-ADC timing offset distribution

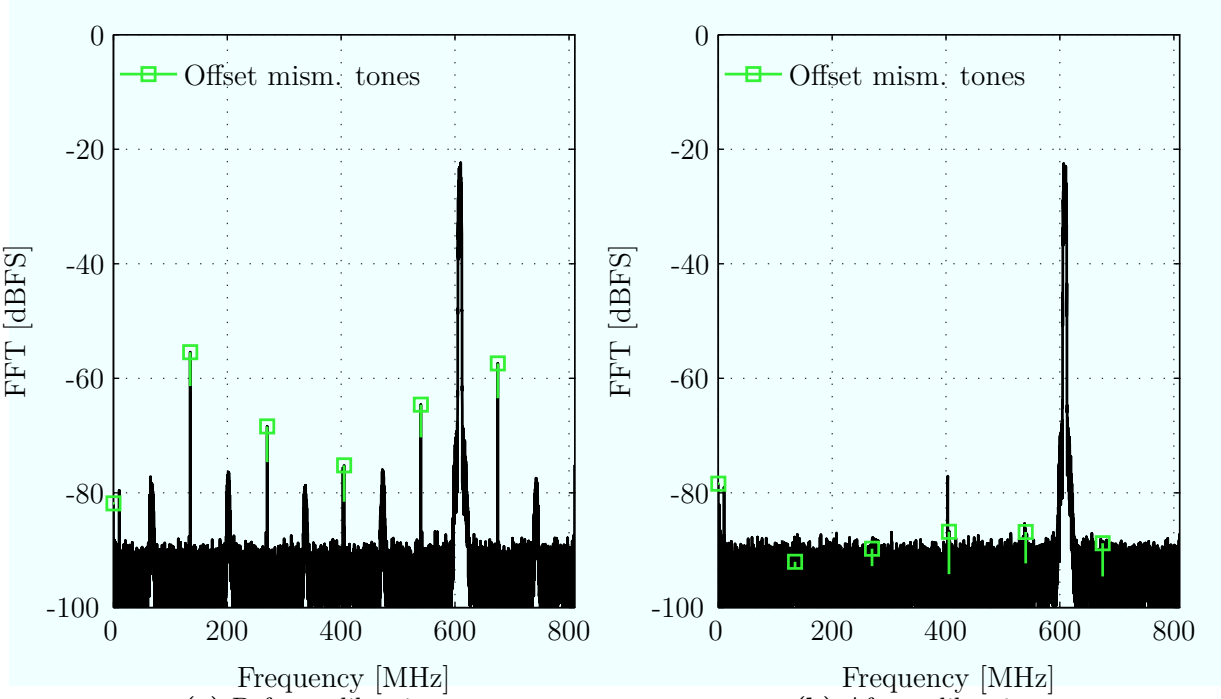
Figure 4.16 – Skew mismatch measurements as a function of temperature

4.3.4 Performance with non sinusoid inputs

It is a common practice to characterize an ADC using a sine input because it helps comparing the performance of different ADCs. When it comes to evaluating mismatch calibration performance however, a sine input is not the most relevant type of signal because it does not have the statistical properties of a real telecommunication signals.

Chapter 2 gives a proof that the calibration algorithms perform well for real telecommunication signals; this is emphasized in Figure 4.17. The figure shows the TIADC output spectrum before and after calibration when the input is a QAM modulated signal with a 1-MHz bandwidth. Now, the skew and gain mismatch frequency artifacts are aliases versions of the input signal and not spurious tones. Before calibration, the mismatch artifacts are above the noise level. After calibration, both the offset mismatch spurious tones and the skew and gain mismatch artifacts are not distinguishable from the noise

floor. Some remaining artifacts, very likely caused by imperfections of the equipment generating the input signal, can be observed around DC and 400 MHz.



(a) Before calibration **(b)** After calibration

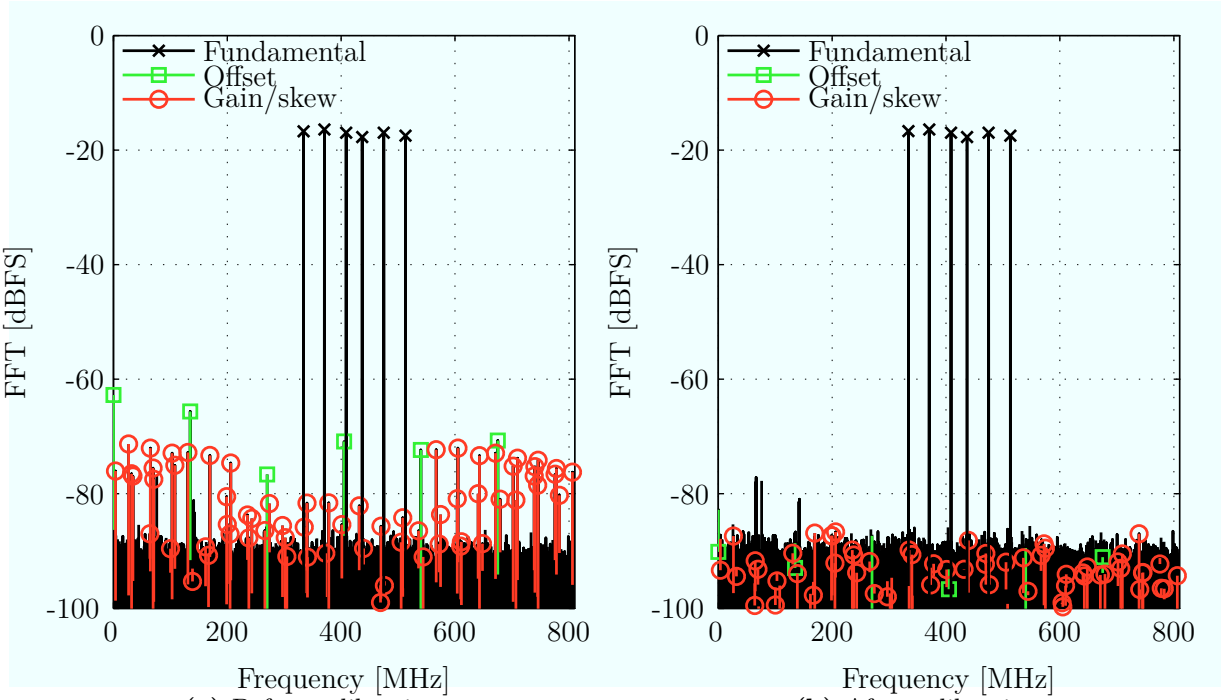
Figure 4.17 – Output spectrum before and after calibration with a QAM modulated input ($N_{\text{FFT}} = 12 \times 4096$)

Results of calibration with a multi-tone signal are displayed in Figure 4.18. They show that the mismatch calibration is efficient for this type of input signal as well. Before calibration (Figure 4.18a), the offset mismatch spurious tones can be seen at fixed frequencies independent of the input signal, whereas the gain and skew mismatch tones are now in greater number because of the multi-tone nature of the input signal. After calibration (Figure 4.18b), the mismatch artifacts are not distinguishable from the noise floor. Some imperfections remain below 200 MHz but they are due to the imperfections of the signal generator that was used for this experiment.

4.3.5 Power consumption breakdown

The pie chart in Figure 4.19 illustrates the power consumed by each part of the TIADC. The overall power consumption is 93.5 mW.

The left half of the pie chart represents the power consumed by the analog front-end (43.5 mW), where most of the power is consumed by the ADC core and the input driver. The digital mismatch calibration power consumption (49.5 mW) is represented on the



(a) Before calibration **(b)** After calibration

Figure 4.18 – Output spectrum before and after calibration with a multitone input ($N_{\mathrm{FFT}} = 12 \times 4096$)

right half of the pie-chart. A large share of the mismatch calibration power comes from the skew mismatch calibration block, mostly because of the 32-tap FIR derivative filter used for the timing correction. It is important to note that the skew estimation power is calculated assuming that the estimation is performed continuously. However, in a real setting, it is likely that the average power consumption would be much less because the estimation block would be turned on only periodically, to track temperature effects for example. The power consumption overhead due to the calibration is not negligible but the next section shows that the overall TIADC energy efficiency is similar or better state of the art high-speed time-interleaved ADCs.

4.4 Comparison with state of the art

Table 4.1 summarizes the performance achieved by our chip, and compares it with previously published TIADCs in the same sampling frequency range.

The key performance of this chip is the mismatch noise reduction, especially skew mismatch noise reduction. The mismatch tones are kept at a level below -70 dBFS across an Effective Resolution Bandwidth (ERBW) representing 90% of the first Nyquist zone. This is better by 10 dB than the best published TIADC [11] with respect to this particular

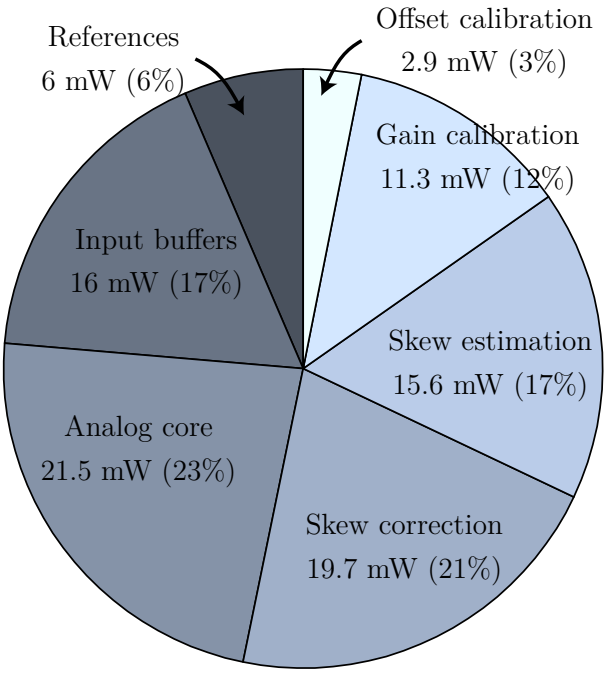


Figure 4.19 – Circuit power breakdown

metric.

This improvement comes at the expense of a slightly lower energy efficiency with a Walden Figure of Merit of 283 fJ/conv. This is worse than the TIADCs published in [11] and [17] [4].

The TIADC achieves a linearity performance similar to the state of the art with a Total Harmonic Distortion (THD) of -58 dB. The SNDR is at least 48 dB over the ERBW, which is also similar to the other circuits in the comparison table.

4.5 Conclusion

After the theoretical proofs and explanations of Chapter 3, this chapter demonstrates the functionally of the mismatch calibration algorithms in a real circuit implementation. The background digital mismatch calibration helps reduce the mismatch noise, especially the timing mismatch noise, to a level lower than state-of-the-art TIADCs. We also show that the background operation is important to track mismatch variations across temperature. This achievement comes at the expense of a higher power consumption (and chip area), which ends up slightly degrading the FOM of the TIADC.

It is also important put this work in perspective with the industrial challenges associated to the design of high-speed data converters. It is indeed extremely difficult to

4. The TIADCs presented in [11] and [17] exhibit lower power consumption because they do not have on-chip reference voltages nor input buffer, which consequently reduces their FOM

	ISSCC 2013 [12]	JSSC 2011 [10]	VLSI 2012 [11]	ISSCC 2014 [17]	**ISSCC 2014 [13]**
Technology CMOS	65 nm	65 nm	65 nm	65 nm	**40 nm**
Sampling rate [GS/s]	3.6	2.6	2.8	1.0	**1.6**
Offset mismatch cal.	yes	yes	yes	yes	**yes**
Gain mismatch cal.	yes	yes	no	no	**yes**
Skew mismatch cal.	no	no	yes	yes	**yes**
Highest mismatch tone [dbFS]	50	55	60	60	**70**
SFDR [dBFS]	50	55	55	60	**62**
THD [dB]	-55	-58	-55	–	**-58**
SNDR [dB]	47	49	48	51.4	**48**
Power [mW]	795	480	44.6	19.8	**93**
Walden FOM [fJ/conv]	1207	801	76	62.3	**283**
Area [mm^2]	7.4	5.1	0.63	0.78	**0.83**

Table 4.1 – Comparison with state of the art

design high-speed TIADCs with a low level of mismatches. Blind background mismatch calibration offers a flexible way to drastically reduce the mismatch effects without modifying the analog design. High Level Synthesis, which offers the possibility to synthesize the calibration blocks in a short time for any TIADC architecture, explains part of this flexibility. A flexible and easily-reusable design is very important in industry but hard to quantize with a hard metric. It is nonetheless important to keep this aspect in mind when comparing background digital calibration to other techniques.

At the end, the calibration techniques presented here are particularly relevant when:

— the mismatches must be reduced to a very low level

— the success of the product requires a quick ADC design cycle

— the power consumption is important but is not the main variable

Chapter 5

Conclusion

The aim of this last chapter is to highlight and summarize the main findings of this work, and to share some perspectives about TIADC mismatch calibration.

5.1 Summary

As explained in Chapter 1, the problem of mismatch calibration in TIADCs arises from the ever increasing demand for high speed ADCs in the communication industry. Indeed, the time-interleaved architecture has become the dominant ADC architecture because its enables both a high sampling rate and a good resolution, for a moderate hardware cost.

The mismatches between the interleaved converters are the main limitation of this architecture. Mismatches are characterized by the fact each sub-ADC has different characteristics (for example offset, gain, sampling instant or bandwidth). Chapter 2 briefly lists some of the physical sources for these mismatches, and offers a characterization of their effects on a TIADC performance, for example in terms of SNDR and SFDR.

An overview of state-of-the-art publications in the domain in Chapter 1 shows that reducing the effects of the mismatches, by means of calibration or other techniques, has been a widely investigated topic both in industry and academia. Most circuit realizations that include mismatch calibration perform some part of the calibration in the analog domain, which makes them *mixed* solutions. Mixed calibration techniques work well but they require the analog front-end of the TIADC to undergo special modifications.

The concept of *fully digital* background mismatch calibration has also been extensively investigated in the literature. From an industrial point of view, it is a very seducing concept because digital designs have a lot of advantages over analog designs when it comes to circuit implementation. Digital designs scale better with technology, they are easily adaptable to any TIADC architecture, and they benefit from fast CAD and simulation

tools. Unfortunately, published digital mismatch calibration techniques are rarely proven beyond numerical simulations. This limitation makes their performance difficult to evaluate. From a more conceptual point of view, the existing calibration techniques are often proven by making assumptions about the input signal that are generally too optimistic in real communication systems.

The goal of this work was to go beyond state-of-the-art research in the area of digital mismatch calibration. The offset and gain mismatch calibration algorithms are inspired from prior art but their convergence is proven for non-stationary signals, which had not been studied in the past. This proof is an important achievement because non-stationary random processes generally describe communication signals more accurately than WSS random processes. The skew mismatch and bandwidth mismatch calibration algorithms achieve a direct estimation of the timing offsets, and the time constant offsets of each sub-ADC. This is a significant difference as compared to the adaptive solutions proposed in the past. Indeed, adaptive solutions need several (potentially long) iterations to converge to the correct estimates, whereas a direct estimation provides the mismatch parameter estimates after a single iteration. The convergence of both the bandwidth mismatch calibration technique and the skew mismatch calibration technique was also proven for non-stationary input signals with certain spectral conditions on their autocovariance function.

If theoretical proofs are an important and required step when it comes to calibration algorithms, they rarely take into account all the non-idealities that can occur in a real circuit. A circuit implementation can reveal limitations that are not always foreseen when working at a theoretical or numerical simulation level. For example, a circuit implementation enables to accurately measure the power consumption of a calibration algorithm, and compare it to the power consumption of the ADC itself. This is hard (if not impossible) if working with numerical simulations only. For that reason, the offset, gain and skew mismatch calibration algorithms described in Chapter 3 were implemented in a 1.6 GS/s TIADC manufactured in ST 40 nm CMOS technology. The measurements of the chip, described in Chapter 4, validate the behavior expected by the theory. With the help of the digital calibration unit, the TIADC achieves the lowest mismatch level among published circuits in the same sampling frequency range. This is to the expense of a power and area overhead that could potentially be reduced in future implementations. Finally, the circuit implementation allows to analyze the mismatch variation across temperatures, which shows the necessity of background calibration.

If summarized, here are the main breakthroughs of this work:

— Analysis and proof of calibration algorithms in the context of non-stationary signals

— Development of a direct method for estimating the timing offsets and time-constant offsets of the sub-ADCs
— First integrated circuit realization of a TIADC embedding a fully digital offset, gain and skew mismatch calibration unit

5.2 Perspectives and future work

This work used the small-mismatch approximation to linearize the mismatch models and devise mismatch estimation and correction methods. This strategy works well in the case of 9-bit 1.6 GS/s TIADC with RMS skew mismatch not higher than 0.2% of the sampling period T_s but it is likely that TIADC performances will keep being pushed forward both in term of sampling speed and resolution. At some point, the second-order effects may start to become non negligible, which will require to modifications of the timing skew calibration.

Adding more complex filters to recover the correct sampling times is an option but, as seen in Chapter 4, the simple derivative-based correction in this work already consumes a significant share of the total power of the circuit. A better solution might be to use a mixed mismatch calibration technique but with a different strategy that what was published so far. We saw that digital mismatch calibration is efficient in reducing the timing mismatches to a very small residual level (of the order of 100 fs RMS), assuming that the original skew mismatch level was small enough. Reaching these levels only by trimming analog elements will probably be difficult, so this role should be left to digital circuits. However, one could think about using the current skew mismatch estimation algorithm to reduce very coarsely the mismatches in the analog domain such that they fall within the acceptable digital calibration range for fine tuning.

The bandwidth mismatch calibration technique will probably be subject to the same type of decision in a more distant future. For now, bandwidth mismatches have not been performance-limiting enough that they require calibration. However, with increasing sampling speeds, it is likely that signal degradation due to bandwidth mismatches will become noticeable in the performance of the circuit. This will be an opportunity for a on-chip validation of the proposed bandwidth mismatch calibration method.

Last but not least, we began to extend the digital mismatch calibration algorithms to-non stationary input random processes because those processes describe communication signals more accurately. It would be interesting to analyze more deeply the statistical characteristics of different classes of real-world wide band communication signals (such as modern CATV signals or satellite signals) to fully understand how they behave in term of their autocovariance function. This analysis would enable to build a very strong

framework to analyze the convergence of blind mismatch calibration techniques in real applications.

Bibliography

[1] *Data Over Cable Service Interface Specifications, DOCSIS 3.0, Physical Layer Specification*, Cable Labs Std., Rev. CM-SP-PHYv3.0-I04-070518, May 2007. 2, 49, 50, 51, 144

[2] *Data-Over-Cable Service Interface Specifications, DOCSIS 3.1, Physical Layer Specification*, Cable Labs Std., Rev. CM-SP-PHYv3.1-I01-131029, October 2013. 2, 49, 50, 144

[3] B. Murmann, "ADC Performance Survey 1997-2014." [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html 3, 5

[4] W. Black and D. Hodges, "Time interleaved converter arrays," *Solid-State Circuits, IEEE Journal of*, vol. 15, no. 6, pp. 1022 – 1029, dec 1980. 3, 6, 12, 144, 146

[5] S. Limotyrakis, S. Kulchycki, D. Su, and B. Wooley, "A 150-MS/s 8-b 71-mW CMOS time-interleaved ADC," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 5, pp. 1057 – 1067, may 2005. 4, 145

[6] S. Gupta, M. Inerfield, and J. Wang, "A 1-GS/s 11-bit ADC With 55-dB SNDR, 250-mW Power Realized by a High Bandwidth Scalable Time-Interleaved Architecture," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 12, pp. 2650 –2657, dec. 2006. 4, 145

[7] C.-C. Hsu, C.-C. Huang, Y.-H. Lin, C.-C. Lee, Z. Soe, T. Aytur, and R.-H. Yan, "A 7b 1.1GS/s Reconfigurable Time-Interleaved ADC in 90nm CMOS," in *VLSI Circuits, 2007 IEEE Symposium on*, june 2007, pp. 66 –67. 4, 145

[8] W.-H. Tu and T.-H. Kang, "A 1.2V 30mW 8b 800MS/s time-interleaved ADC in 65nm CMOS," in *VLSI Circuits, 2008 IEEE Symposium on*, june 2008, pp. 72 –73. 4, 145

[9] J. Wu, C.-Y. Chen, T. Li, L. He, W. Liu, W.-T. Shih, S. Tsai, B. Chen, C.-S. Huang, B.-J. Hung, H. Hung, S. Jaffe, L. Tan, and H. Vu, "A 240-mW 2.1-GS/s 52-dB SNDR Pipeline ADC Using MDAC Equalization," *Solid-State Circuits, IEEE Journal of*, vol. PP, no. 99, pp. 1–11, 2013. 4, 145

[10] K. Doris, E. Janssen, C. Nani, A. Zanikopoulos, and G. van der Weide, "A 480 mW 2.6 GS/s 10b Time-Interleaved ADC With 48.5 dB SNDR up to Nyquist in 65 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 12, pp. 2821 –2833, dec. 2011. 4, 129, 145, 160

[11] D. Stepanovic and B. Nikolic, "A 2.8GS/s 44.6mW time-interleaved ADC achieving 50.9dB SNDR and 3dB effective resolution bandwidth of 1.5GHz in 65nm CMOS," in *VLSI Circuits (VLSIC), 2012 Symposium on*, june 2012, pp. 84 –85. 4, 8, 9, 127, 128, 129, 145, 147, 148, 160

[12] E. Janssen, K. Doris, A. Zanikopoulos, A. Murroni, G. v. d. Weide, Y. Lin, L. Alvado, F. Darthenay, and Y. Fregeais, "An 11b 3.6GS/s time-interleaved SAR ADC in 65nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, 2013, pp. 464–465. 4, 129, 145, 160

[13] N. Le Dortz, J.-P. Blanc, T. Simon, S. Verhaeren, E. Rouat, P. Urard, S. Le Tual, D. Goguet, C. Lelandais-Perrault, and P. Benabes, "A 1.62GS/s time-interleaved SAR ADC with digital background mismatch calibration achieving interleaving spurs below 70dBFS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 386–388. 4, 111, 129, 145, 160

[14] V.-C. Chen and L. Pileggi, "A 69.5mW 20GS/s 6b time-interleaved ADC with embedded time-to-digital calibration in 32nm CMOS SOI," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 380–381. 4, 9, 145

[15] S. Le Tual, P. Singh, C. Curis, and P. Dautriche, "22.3 A 20GHz-BW 6b 10GS/s 32mW time-interleaved SAR ADC with Master T amp;H in 28nm UTBB FDSOI technology," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 382–383. 4, 20, 114, 145

[16] L. Kull, T. Toifl, M. Schmatz, P. Francese, C. Menolfi, M. Braendli, M. Kossel, T. Morf, T. Andersen, and Y. Leblebici, "A 90GS/s 8b 667mW 64-interleaved SAR ADC in 32nm digital SOI CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 378–379. 4, 7, 145, 146

[17] S. Lee, A. Chandrakasan, and H.-S. Lee, "A 1GS/s 10b 18.9mW time-interleaved SAR ADC with background timing-skew calibration," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 384–385. 5, 9, 128, 129, 160

[18] C. Vogel, D. Draxelmayr, and F. Kuttner, "Compensation of timing mismatches in time-interleaved analog-to-digital converters through transfer characteristics tuning," in *Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on*, vol. 1, july 2004, pp. I – 341–4 vol.1. 7, 9, 146, 147

[19] Y. Jenq, "Digital spectra of nonuniformly sampled signals: theories and applications-measuring clock/aperture jitter of an A/D system," *Instrumentation and Measurement, IEEE Transactions on*, vol. 39, no. 6, pp. 969 –971, dec 1990. 7, 12, 146, 150

[20] M. El-Chammas and B. Murmann, "A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 4, pp. 838 –847, april 2011. 7, 8, 146, 147

[21] D. Fu, K. Dyer, S. Lewis, and P. Hurst, "A digital background calibration technique for time-interleaved analog-to-digital converters," *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 12, pp. 1904 –1911, dec 1998. 7, 146

[22] J.-E. Eklund and F. Gustafsson, "Digital offset compensation of time-interleaved ADC using random chopper sampling," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 3, 2000, pp. 447 –450 vol.3. 7, 146

[23] S. Mendel and C. Vogel, "On the Compensation of Magnitude Response Mismatches in M-channel Time-interleaved ADCs," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, may 2007, pp. 3375 –3378. 7, 146

[24] P. Satarzadeh, B. Levy, and P. Hurst, "Adaptive Semiblind Calibration of Bandwidth Mismatch for Two-Channel Time-Interleaved ADCs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 9, pp. 2075 –2088, sept. 2009. 7, 146

[25] L. Wu and J. Black, W.C., "A low-jitter skew-calibrated multi-phase clock generator for time-interleaved applications," in *Solid-State Circuits Conference, 2001. Digest of Technical Papers. ISSCC. 2001 IEEE International*, Feb 2001, pp. 396–397. 8

[26] J. Elbornsson, F. Gustafsson, 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, no. 1, pp. 151 – 158, jan. 2004. 8, 147

[27] D. Stepanovic and B. Nikolic, "A 2.8 GS/s 44.6 mW Time-Interleaved ADC Achieving 50.9 dB SNDR and 3 dB Effective Resolution Bandwidth of 1.5 GHz in 65 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 4, pp. 971–982, 2013. 8, 9, 147, 148

[28] I.-N. Ku, Z. Xu, Y.-C. Kuan, Y.-H. Wang, and M.-C. Chang, "A 40-mW 7-bit 2.2-GS/s Time-Interleaved Subranging CMOS ADC for Low-Power Gigabit Wireless Communications," *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 8, pp. 1854 –1865, aug. 2012. 8, 147

[29] J. Atherton and H. Simmonds, "An offset reduction technique for use with CMOS integrated comparators and amplifiers," *Solid-State Circuits, IEEE Journal of*, vol. 27, no. 8, pp. 1168–1175, Aug 1992. 8

[30] K.-L. Wong and C.-K. Yang, "Offset compensation in comparators with minimum input-referred supply noise," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 5, pp. 837–840, May 2004. 8, 19

[31] V. Divi and G. Wornell, "Blind Calibration of Timing Skew in Time-Interleaved Analog-to-Digital Converters," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 3, pp. 509 –522, june 2009. 8, 10, 147, 148

[32] S. Jamal, D. Fu, N.-J. Chang, P. Hurst, and S. Lewis, "A 10-b 120-Msample/s time-interleaved analog-to-digital converter with digital background calibration," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 12, pp. 1618 – 1627, dec 2002. 8

[33] J. Elbornsson and J.-E. Eklund, "Blind estimation of timing errors in interleaved AD converters," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 6, 2001, pp. 3913 – 3916 vol.6. 8, 147

[34] J. Elbornsson, K. Folkesson, and J.-E. Eklund, "Measurement verification of estimation method for time errors in a time-interleaved A/D converter system," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 3, 2002, pp. III–129 – III–132 vol.3. 8, 147

[35] M. Seo, M. Rodwell, and U. Madhow, "A Low Computation Adaptive Blind Mismatch Correction for Time-Interleaved ADCs," in *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on*, vol. 1, aug. 2006, pp. 292 –296. 8, 9, 147, 148

[36] H. Jin, E. Lee, and M. Hassoun, "Time-interleaved A/D converter with channel randomization," in *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, vol. 1, jun 1997, pp. 425 –428 vol.1. 9, 147

[37] K. El-Sankary, A. Assi, and M. Sawan, "New sampling method to improve the SFDR of time-interleaved ADCs," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 1, may 2003, pp. I–833 – I–836 vol.1. 9, 147

[38] R. Payne and M. Corsi, "Track and hold architecture with tunable bandwidth," US Patent 8 248 282 B2, 2012. 9, 147

[39] C. Luo, L. Zhu, and J. McClellan, "Coordinated blind calibration for time interleaved ADCS," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 3890–3894. 9, 148

[40] M. Seo, M. Rodwell, and U. Madhow, "Blind Correction of Gain and Timing Mismatches for a Two-Channel Time-Interleaved Analog-to-Digital Converter," in *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, 28 - november 1, 2005, pp. 1121 – 1125. 9, 148

[41] ——, "Blind correction of gain and timing mismatches for a two-channel time-interleaved analog-to-digital converter: experimental verification," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, may 2006, pp. 4 pp. –3397. 9, 148

[42] M. Seo and M. Rodwell, "Generalized Blind Mismatch Correction for a Two-Channel Time-Interleaved ADC: Analytic Approach," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, may 2007, pp. 109 –112. 10, 148

[43] M. Seo, M. Rodwell, and U. Madhow, "Generalized Blind Mismatch Correction for Two-Channel Time-Interleaved A-to-D Converters," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, april 2007, pp. III–1505 –III–1508. 10

[44] ——, "Comprehensive digital correction of mismatch errors for a 400-msamples/s 80-dB SFDR time-interleaved analog-to-digital converter," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 53, no. 3, pp. 1072 – 1082, march 2005. 10, 148

[45] J. Matsuno, T. Yamaji, M. Furuta, and T. Itakura, "All-digital background calibration for time-interleaved ADC using pseudo aliasing signal," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, may 2012, pp. 1050 –1053. 10, 148

[46] ——, "All-Digital Background Calibration Technique for Time-Interleaved ADC Using Pseudo Aliasing Signal," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. PP, no. 99, pp. 1–9, 2013. 10, 148

[47] S. Jamal, D. Fu, M. Singh, P. Hurst, and S. Lewis, "Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 1, pp. 130–139, Jan 2004. 10, 148

[48] H. Le Duc, C. Jabbour, P. Desgreys, O. Jamin, and V. T. Nguyen, "A fully digital background calibration of timing skew in undersampling TI-ADC," in *New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International*, June 2014, pp. 53–56. 10, 148

[49] C. Vogel, "A Frequency Domain Method for Blind Identification of Timing Mismatches in Time-Interleaved ADCs," in *Norchip Conference, 2006. 24th*, nov. 2006, pp. 45 –48. 10, 148

[50] S. Saleem and C. Vogel, "LMS-based identification and compensation of timing mismatches in a two-channel time-interleaved analog-to-digital converter," in *Norchip, 2007*, nov. 2007, pp. 1 –4. 10, 148

[51] C. Vogel, S. Saleem, and S. Mendel, "Adaptive blind compensation of gain and timing mismatches in M-channel time-interleaved ADCs," in *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, 31 2008-sept. 3 2008, pp. 49 –52. 10, 148

[52] S. Saleem and C. Vogel, "On blind identification of gain and timing mismatches in time-interleaved analog-to-digital converters," in *Proceedings of 33rd International Conference on Telecommunications and Signal Processing (TSP 2010)*. Citeseer, 2010, pp. 151–155. 10, 148

[53] ——, "Adaptive compensation of frequency response mismatches in high-resolution time-interleaved ADCs using a low-resolution ADC and a time-varying filter," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 30 2010-june 2 2010, pp. 561 –564. 10, 148

[54] ——, "Adaptive Blind Background Calibration of Polynomial-Represented Frequency Response Mismatches in a Two-Channel Time-Interleaved ADC," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 6, pp. 1300 –1310, june 2011. 10, 148

[55] Y. Eldar and A. Oppenheim, "Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples," *Signal Processing, IEEE Transactions on*, vol. 48, no. 10, pp. 2864 –2875, oct 2000. 10, 148

[56] H. Jin and E. Lee, "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, no. 7, pp. 603 –613, jul 2000. 10, 148

[57] H. Johansson and P. Lowenborg, "Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters," *Signal Processing, IEEE Transactions on*, vol. 50, no. 11, pp. 2757 – 2767, nov 2002. 10, 148

[58] S. Huang and B. Levy, "Blind Calibration of Timing Offsets for Four-Channel Time-Interleaved ADCs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, no. 4, pp. 863 –876, april 2007. 10, 148

[59] A. Petraglia and S. Mitra, "Analysis of mismatch effects among A/D converters in a time-interleaved waveform digitizer," *Instrumentation and Measurement, IEEE Transactions on*, vol. 40, no. 5, pp. 831 –835, oct 1991. 12, 150

[60] B. Yu and J. Black, W.C., "Error analysis for time-interleaved analog channels," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 1, may 2001, pp. 468 –471 vol. 1. 12, 150

[61] Y.-C. Jenq, "Digital spectra of nonuniformly sampled signals: fundamentals and high-speed waveform digitizers," *Instrumentation and Measurement, IEEE Transactions on*, vol. 37, no. 2, pp. 245 –251, jun 1988. 12, 150

[62] ——, "Digital-to-analog (D/A) converters with nonuniformly sampled signals," *Instrumentation and Measurement, IEEE Transactions on*, vol. 45, no. 1, pp. 56 –59, feb 1996. 12, 150

[63] S.-W. Sin, S.-P. U, R. Martins, and J. Franca, "Timing-mismatch analysis in high-speed analog front-end with nonuniformly holding output," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 1, may 2003, pp. I–129 – I–132 vol.1. 12, 150

[64] C. Vogel, "Comprehensive, error analysis of combined channel mismatch effects in time-interleaved ADCs," in *Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Proceedings of the 20th IEEE*, vol. 1, may 2003, pp. 733 – 738. 12, 150

[65] ——, "The impact of combined channel mismatch effects in time-interleaved ADCs," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, no. 1, pp. 415 – 427, feb. 2005. 12, 150

[66] C. Vogel and G. Kubin, "Time-interleaved ADCs in the context of hybrid filter banks," in *Proceedings International Symposium on Signals, Systems, and Electronics (ISSSE)*, 2004, pp. 214–217. 12, 150

[67] ——, "Analysis and compensation of nonlinearity mismatches in time-interleaved ADC arrays," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 1, may 2004, pp. I – 593–6 Vol.1. 12

[68] ——, "Modeling of time-interleaved ADCs with nonlinear hybrid filter banks," *AEU-International Journal of Electronics and Communications*, vol. 59, no. 5, pp. 288–296, 2005. 12, 150

[69] G. Leger, E. Peralias, A. Rueda, and J. Huertas, "Impact of random channel mismatch on the SNR and SFDR of time-interleaved ADCs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 1, pp. 140 – 150, jan. 2004. 12, 150

[70] M. El-Chammas and B. Murmann, "General analysis on the impact of phase-skew in time-interleaved ADCs," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, may 2008, pp. 17 –20. 13, 150

[71] M. Pelgrom, A. C. J. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *Solid-State Circuits, IEEE Journal of*, vol. 24, no. 5, pp. 1433–1439, Oct 1989. 16, 17, 149

[72] O. Weber, O. Faynot, F. Andrieu, C. Buj-Dufournet, F. Allain, P. Scheiblin, J. Foucher, N. Daval, D. Lafond, L. Tosti, L. Brevard, O. Rozeau, C. Fenouillet-Beranger, M. Marin, F. Boeuf, D. Delprat, K. Bourdelle, B. Nguyen, and S. Deleonibus, "High immunity to threshold voltage variability in undoped ultra-thin FDSOI MOSFETs and its physical understanding," in *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, Dec 2008, pp. 1–4. 17

[73] A. Asenov, "Simulation of Statistical Variability in Nano MOSFETs," in *VLSI Technology, 2007 IEEE Symposium on*, June 2007, pp. 86–87. 17

[74] P. Oldiges, Q. Lin, K. Petrillo, M. Sanchez, M. Ieong, and M. Hargrove, "Modeling line edge roughness effects in sub 100 nanometer gate length devices," in *Simulation of Semiconductor Processes and Devices, 2000. SISPAD 2000. 2000 International Conference on*, 2000, pp. 131–134. 17

[75] J.-S. Lim, S. Thompson, and J. Fossum, "Comparison of threshold-voltage shifts for uniaxial and biaxial tensile-stressed n-MOSFETs," *Electron Device Letters, IEEE*, vol. 25, no. 11, pp. 731–733, Nov 2004. 17

[76] A. Asenov, S. Kaya, and J. Davies, "Intrinsic threshold voltage fluctuations in decanano MOSFETs due to local oxide thickness variations," *Electron Devices, IEEE Transactions on*, vol. 49, no. 1, pp. 112–119, Jan 2002. 17

[77] J.-B. Shyu, G. Temes, and K. Yao, "Random errors in MOS capacitors," *Solid-State Circuits, IEEE Journal of*, vol. 17, no. 6, pp. 1070–1076, Dec 1982. 18

[78] V. Tripathi and B. Murmann, "Mismatch Characterization of Small Metal Fringe Capacitors," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 8, pp. 2236–2242, Aug 2014. 19

[79] R. Aparicio and A. Hajimiri, "Capacity limits and matching properties of integrated capacitors," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 3, pp. 384–393, Mar 2002. 19

[80] W. Gardner, *Introduction to random processes: with applications to signals and systems*. Macmillan Pub. Co., 1986. 50

Résumé en français

Introduction

Problématique générale

Dans les systèmes de communications modernes, l'information transmise entre un émetteur et un récepteur est véhiculée par des signaux *analogiques*. Ces signaux analogiques peuvent prendre la forme de courants électriques transmis par des câbles en cuivre ou encore d'impulsions lumineuses transmises par des fibres optiques. Ils peuvent aussi, dans le cas des communications sans fil, être des ondes électromagnétiques qui se déplacent dans le vide. Cependant l'information qu'ils contiennent doit généralement être traitée par des circuits *numériques*. En effet, les circuits numériques comportent de nombreux avantages par rapport aux circuit analogiques. Ils permettent notamment d'effectuer des traitements mathématiques complexes comme de la correction d'erreur. De plus, concevoir des circuits numériques est plus aisé car les outils de génération de circuit sont largement automatisés. Les circuits numériques sont aussi plus facilement portables d'un noeud technologique à un autre, et plus robustes au bruit. Enfin, de part la nature discrète des signaux, les simulations numériques sont beaucoup plus rapides que les simulations analogiques.

La conversion du domaine analogique au domaine numérique est faite par des Convertisseurs Analogique-Numérique (CAN ou ADC pour *Analog-to-Digital converter*). Ceux-ci sont souvent considérés comme un goulot d'étranglement dans les systèmes de communication modernes. Le but d'un ADC est d'échantillonner le signal analogique continu à des intervalles de temps réguliers et de quantifier sa valeur à chaque instant en utilisant un nombre finis de bits.

Du fait des avantages des circuits numériques, la tendance est la réduction des composants analogiques dans les systèmes de communication. Ceux-ci sont remplacés par des équivalents numériques ayant une fonction similaire. On peut notamment citer l'exemple de l'échantillonnage direct en radio-fréquence (*RF direct sampling*). Il s'agit d'effectuer la conversion analogique numérique dans le domaine RF, c'est-à-dire sans ramener le signal

au préalable dans une bande fréquence intermédiaire (IF pour *Intermediate Frequency*) par l'intermédiaire de mixeurs analogiques.

Cela implique de déplacer l'ADC plus en amont dans la chaîne de réception. Cependant, cela impose des contraintes de performances élevées sur l'ADC. Pour illustrer ce problème, on peut citer le cas de la transmission TV par câble où le signal est transmis sur un câble coaxial sur une bande de fréquence allant de 54 MHz à 1002 MHz. Dans le standard le plus commun aujourd'hui (DOCSIS 3.0 [1]), cette bande est occupée par des canaux de 6 MHz de large, représentant chacun une chaîne de télévision. Ces canaux utilisent généralement des modulations de type QAM 256 mais il est prévu que la taille des constellations soient augmentée dans les versions future du standard (DOCSIS 3.1 [2]). D'autant plus que certains fournisseurs cherchent à utiliser le câble pour transmettre d'autres types de services multimédia (jeux, vidéos en streaming, TV HD) sous le standard MoCA (Multimedia over Coax Alliance). Ce standard utilise une bande qui peut s'étaler de 500 MHz à 1650 MHz. En résumé, le signal occupe potentiellement une bande fréquentielle allant de 54 MHz jusqu'à 1650 MHz et est encodé en utilisant des techniques de modulations complexes.

Pour parvenir à effectuer de l'échantillonnage direct en RF sur ce type de signal, la fréquence d'échantillonnage de l'ADC doit être au moins deux fois supérieure à la largeur de bande du signal, soit par exemple 3.2 GE/s. Sa résolution doit aussi être importante, de l'ordre de 10 bits effectifs.

Convertisseurs analogique-numérique à entrelacement temporel

Il existe de nombreuses architectures d'ADCs, parmi lesquelles figurent notamment les ADCs *flash*, les ADCs *pipeline*, les ADCs à approximations successives (SAR ADC pour *Successive Approximation Register* ADC) ou encore les ADCs à entrelacement temporel (TIADC pour *Time-Interleaved* ADC). Les ADCs *pipeline* ont longtemps été très utilisés pour atteindre une fréquence d'échantillonnage élevée (> 100 MHz) avec un bon rapport signal sur bruit (SNDR pour *Signal to Noise and Distorsion Ratio*). Il sont maintenant en compétition avec les TIADCs.

Inventés dans les années 1980 [4], le TIADC est, aujourd'hui, le seul type de convertisseur qui permet d'atteindre de des fréquences d'échantillonnage de l'ordre du GE/s avec un SNDR supérieur à 50 dB. En pratique, il est exagéré de séparer les TIADCs des autres familles de convertisseurs analogique-numérique puisque un TIADC est composé de plusieurs sous-ADCs appartenant à l'une des autres familles (par exemple des ADCs SAR). L'idée de l'architecture à entrelacement temporel est d'augmenter la fréquence d'échantillonnage en mettant en parallèle plusieurs sous-convertisseurs fonctionnant à une vitesse plus faible. Les sous-ADCs échantillonnent le signal d'entrée tour à tour et

leurs échantillons de sortie sont recombinés pour constituer le signal discrétisé complet. Un autre avantage des TIADCs est de présenter une efficacité énergétique très bonne car chacun des sous-convertisseurs peut-être conçu de sorte à fonctionner dans sa zone de fonctionnement optimale (en terme de fréquence d'échantillonnage).

Comme mentionné précédemment, un TIADC est constitué de plusieurs sous-convertisseurs, eux-mêmes appartenant à une catégorie d'ADCs. Parmi les type de convertisseurs fréquemment utilisés comme sous-ADCs de base, les ADCs *pipeline* [5, 6, 7, 8, 9] sont souvent privilégiés mais ils sont de plus en plus fréquemment délaissés au profit des ADCs SAR [10, 11, 12, 13]. Les TIADCs à base d'ADCs SAR sont ceux qui permettent d'atteindre les fréquences d'échantillonnage les plus élevées [14, 15] avec en particulier une fréquence d'échantillonnage record de 90 GE/s [16]. Ce choix est dû au fait que les performance des ADCs *pipeline* tendent à empirer dans les nouveaux noeuds technologiques, du fait des plus faibles gains des transistors. D'un autre côté, les ADCs SAR ne contiennent pas de circuits dynamiques (amplificateurs) et bénéficient davantage de l'évolution des technologies CMOS grâce à leur fonctionnement très similaire à un circuit numérique.

Désappariements dans les TIADCs

Bien que les TIADCs semblent être la solution idéale pour augmenter la fréquence d'échantillonnage des convertisseurs, en mettant en parallèle de nombreux sous-ADCs, leurs performances sont limitées par les désappariements entre les sous-convertisseurs. En effet, lors de la fabrication du circuit, il est impossible de garantir que tous les sous-ADCs auront les mêmes caractéristiques, notamment à cause des variations aléatoires dans le processus de fabrication.

La conséquence de ces désappariements est une dégradation du signal numérique de sortie, ce qui se traduit par un bruit en sortie du TIADC. Les désappariements peuvent être classés en différentes catégories en fonction de leurs effets. Les catégories les plus communes, celles qui en pratique limitent les performances du TIADC, sont :

- les désappariements d'offset (*offset mismatches*) lorsque les sous-ADCs ont des offsets différents
- les désappariements de gain (*gain mismatches*) lorsque les sous-ADCs ont des gains différents
- les désappariements d'instant d'échantillonnage (*timing skew mismatches*) lorsque les sous-ADCs échantillonnent le signal avec un décalage par rapport à l'instant d'échantillonnage idéal
- les désappariements de bande passante (*bandwidth mismatches*) lorsque les filtres d'entrée des sous-ADCs ont des fréquences de coupure différentes

Les désappariements d'offset se traduisent, dans le spectre de sortie, par des raies à des

fréquences multiples de fréquence d'échantillonnage des sous-ADCs. Les désappariements de gain, d'instant d'échantillonnage et de bande passante créent des réplications du signal d'entrée autour des fréquences multiples de la fréquence d'échantillonnage des sous-ADCs. Les sources et les effets des désappariements sont traités en détails dans le Chapitre 2.

La réduction du bruit dû aux désappariements est apparu comme un challenge dès l'invention du premier TIADC [4]. Bien que de nombreuses solutions de calibration des désappariements aient été proposées, le challenge existe toujours aujourd'hui.

État de l'art de la calibration des désappariements

Différentes types de techniques de calibrations ont été inventées pour réduire les effets des désappariements dans les TIADCs depuis l'invention du premier TIADC.

Calibration hors-ligne

Le principe des techniques de calibrations hors-ligne est de faire la calibration lorsque que le TIADC est dans un mode "spécial" de calibration [18, 19]. Durant ces phase, dite hors-ligne, le signal d'entrée peut être remplacé par un signal connu, par exemple un signal sinusoïdal de fréquence et d'amplitude connues. Dans le circuit présenté dans [16], les instants d'échantillonnage des sous-ADCs sont ajustés dans le domaine analogique, avec un signal sinusoïdal connu à l'entrée du TIADC. Dans [20], les offsets des sous-ADCs sont calibrés hors-ligne en ajustant certaines caractéristiques des comparateurs constituant les sous-ADCs. Les techniques de calibration hors-ligne ne sont cependant pas adaptées à des applications dans lesquelles le convertisseur ne peut pas être arrêté, ce qui est généralement le cas dans une chaîne de réception. En effet, les variations de température et les effets du vieillissement du circuit peuvent nécessiter de calibrer le convertisseur très fréquemment.

Calibration en ligne

L'alternative aux techniques de calibration hors-ligne est la calibration en ligne qui consiste à effectuer la calibration pendant le fonctionnement normal du TIADC, sans l'arrêter.

Certaines de ces techniques nécessitent cependant de modifier le signal d'entrée dans le domaine analogique. C'est le cas des techniques présentées dans [21] et [22] pour lesquelles un séquence aléatoire est ajoutée ou multipliée au signal d'entrée avant la conversion. Dans [23] et [24], c'est un signal sinusoïdal qui est rajouté au signal d'entrée pour la calibration des désappariements de bande passante.

Les techniques de calibration dites aveugles, en revanche, ne nécessitent pas de modifier le signal d'entrée, ce qui réduit le risque d'introduire d'autres perturbations. Bien sûr, développer des techniques de calibration aveugles est, en général, plus difficile car peu d'informations sur le signal d'entrée sont disponibles. Parmi les techniques de calibration aveugles, on peut distinguer celles qui sont mixtes, c'est-à-dire en partie faites à la fois dans le domaine numérique et dans le domaine analogique, et celles qui sont entièrement numériques.

Calibration mixte

La plupart des techniques de calibration mixtes sont séparées en une partie numérique qui quantifie les désappariements et une partie analogique qui utile les informations provenant de la partie numérique afin de réduire les désappariements.

Ainsi, une manière classique d'estimer les désappariements d'offsets est de mesurer les différences entre les moyennes des signaux de sortie de chaque sous-ADCs [26, 27]. Cette information peut ensuite servir à ajuster les caractéristiques des comparateurs présents dans les sous-ADCs [28].

De façon similaire, les désappariements de gain peuvent être mesurés en calculant les rapports entre les puissances moyennes de sortie des sous-ADCs [26]. Un autre approche est d'essayer de minimiser le bruit dû aux désappariements de gains dans le domaine fréquentiel [31].

Les techniques de calibration des désappariements d'instants d'échantillonnage sont généralement plus complexes. L'estimation des désappariements peut se faire dans le domaine temporel en minimisant des un fonction de coût basée sur des corrélations des signaux de sortie des sous-ADCs [33, 34, 35]. Les instants d'échantillonnage des sous-ADCs peuvent ensuite être ajustés à l'aide de délais ajustables dans le domaine analogique. Certaines techniques requièrent des sous-ADCs redondants [11, 27, 20], ce qui complique la conception analogique. Enfin, d'autres approches [36, 37, 18] proposent de réorganiser aléatoirement les instants d'échantillonnage des sous-ADCs de façon étaler le bruit sur toute la bande fréquentielle.

Il existe peu de techniques de calibration mixtes des désappariements de bande passante mais l'on peut citer le brevet [38] dans lequel la bande passante de chaque sous-ADC est ajustée en faisant varier le condensateur de *bootstrap* dans l'échantillonneur.

Calibration numérique

Les techniques de calibration entièrement numériques des désappariements présentent certains avantages comparées aux techniques de calibration mixtes. En effet, elles ne né-

cessitent pas de modifier la partie analogique du TIADC et elles s'adaptent à n'importe quelle architecture de TIADC. De plus, elles sont facilement portables d'un noeud technologique à un autre, et bénéficie des gains de performances associés à la réduction de la taille des transistors. Cependant, certaines fonctions qui peuvent être faites dans le domaine analogique à un coût modéré sont beaucoup plus coûteuses à effectuer dans le domaine numérique.

La correction des désappariements d'offset et de gain dans le domaine numérique est assez aisée. La correction des désappariements d'offset peut se faire en soustrayant du signal de sortie de chaque sous-ADC l'offset estimé correspondant [11, 39, 27]. De la même manière, la correction des désappariements de gain peut se faire en multipliant le signal de sortie de chaque sous-ADC par l'inverse du gain estimé correspondant [35, 39]. L'estimation des désappariements d'offset et de gain peut être faite de la même façon que pour les calibrations mixtes.

Les solutions purement numériques de calibration des désappariements d'instant d'échantillonnage et de bande passante ont été très étudiés de manière théorique mais peu de circuit intégrés ont fait la preuve de leur bon fonctionnement. Les techniques présentées dans [40, 44, 41, 42, 42] corrigent le signal de sortie du TIADC de telle sorte à ce qu'il retrouve sa propriété de stationnarité au sens large. Dans [45, 46, 47, 48], les signaux de sortie des sous-ADCs subissent une transformation de Hadamard, ce qui permet de séparer le signal désiré du bruit dû aux désappariements. Dans [49, 50, 51, 52, 31, 53, 54], les effets des désappariements de bande passante et d'instant d'échantillonnage sont compensés en minimisant le bruit qu'ils génèrent dans une bande fréquentielle qui ne contient pas le signal utile. La correction des désappariements se fait ensuite au moyen de filtres numériques, coûteux dans la plupart des cas, du fait de leurs coefficients variables. Les travaux présentés dans [55, 56, 57, 58] se concentrent uniquement sur la correction des désappariements.

Objectifs de la thèse

Le but de ce travail est de démontrer la faisabilité d'une méthode numérique et aveugle de calibration des désappariements d'offset, de gain, d'instant d'échantillonnage et de bande passante, implémentable dans un circuit intégré et adaptée aux signaux de communication réels.

Les solutions de d'état de l'art sur ce sujet sont majoritairement théoriques. Même si des simulations numériques prouvent leur efficacité, la plupart des solutions existantes ajouteraient une surconsommation non négligeable si elles étaient réalisées dans un circuit. C'est notamment le cas pour les techniques de calibration des désappariements de bande

passante et des désappariements d'instants d'échantillonnage qui nécessitent souvent des filtres de correction complexes. D'autre part, les solutions proposées dans l'état de l'art sont toutes itératives et nécessiteraient un temps de convergence long si elles étaient appliquées à un signal réel de communication. Enfin, la plupart des travaux partent du principe que le signal d'entrée du TIADC est un signal stationnaire au sens large (SSL) Ce n'est pas le cas, en pratique, pour des signaux réels de télécommunication. Ceux-ci sont en général cyclostationnaires au sens large (CSSL) au mieux, voire non stationnaires.

Les travaux présentés dans cette thèse ont pour but d'apporter une réponse à ces limitations. Le Chapitre 2 est principalement une reformulation de nombreux travaux précédents. Il explique les sources des désappariements dans un circuit et analyse leurs effets sur le signal de sortie. En particulier, il donne des indicateurs de performances du TIADC (SNDR et SFDR) en fonction du niveau des désappariements

Le Chapitre 3 constitue le coeur de cette thèse car il décrit des techniques de calibration des désappariements entièrement numériques et non itératives. Il analyse aussi les suppositions théoriques permettant le bon fonctionnement de ces techniques de calibration, en particulier lorsque le signal est une réalisation d'un signal non stationnaire.

Le Chapitre 4 présente le circuit intégré d'un TIADC cadencé à 1.62 GE/s et intégrant les calibrations des désappariements d'offset, de gain et d'instant d'échantillonnage présentées dans le Chapitre 3. Les mesures de ce circuit démontre la viabilité des calibrations entièrement numérique, tout en mettant en évidence certaines de leur limitations.

Enfin, le Chapitre 5 résume les résultats des chapitres précédents et donne des perspectives concernant le futur des méthodes de calibration des désappariements dans les TIADCs.

Sources et effets des désappariements dans les TIADCs

Sources des désappariements

Les désappariements dans les TIADCs, bien qu'ils puissent être dûs à des erreurs de design, sont généralement intrinsèques à la technologie employée pour créer le circuit. En effet, il est impossible de concevoir deux circuits dont les composants présentent des caractéristiques physiques égales. Du fait de phénomènes aléatoire, des composants d'un même circuit supposés être identiques peuvent ainsi présenter des différences de comportements. Ces désappariements, sont d'autant plus important que la taille des composantes est faible, comme l'affirme la loi de Pelgröm [71].

Parmi les composants de base des circuits intégrés en technologie CMOS on trouve notamment les transistors et les condensateurs, et ce sont eux qui sont à la source d'une

grande partie des désappariements dans les TIADCs. Ainsi deux transistors d'un même circuit peuvent ainsi présenter des désappariements de tension de seuil, ce qui affecte nombre de leurs caractéristiques telles que leur résistance à l'état fermé ou leur vitesse de fermeture. Ces désappariements de tension de seuil sont un des contributeurs majeurs en ce qui concerne les désappariements d'offset, de gain, de décalage de temps de bande passante. De la même façon, deux condensateurs conçus avec des paramètres identiques auront des capacités différentes. Étant donné que les circuits d'échantillonnages sont généralement modélisables par un circuit RC, on comprend bien que ces désappariements de condensateurs peuvent être une source importante des désappariements des bande passante dans les TIADCs.

Effets des désappariements

Les effets des Désappariements sur les performances des TIADCs ont fait l'objet de nombreuses études [59, 60, 61, 19, 62, 63, 64, 65, 66, 68, 69, 70] et sont donc un sujet bien connu. Certains des résultats de ces études sont redémontrés dans ce mémoire afin de donner une lecteur une bonne compréhension des effets des désappariements.

Ainsi, il est redémontré que les désappariements d'offset créent, dans le spectre du signal de sortie, des raies à des fréquence multiples de F_s/M, où F_s est la fréquence d'échantillonnage du TIADC et M est le nombre de sous-ADCs entrelacés. Cela conduit à une dégradation du SNDR (calculé sur un signal sinusoïdal) en sortie du TIADC, et on peut ainsi démontrer que le SNDR en présence de désappariements d'offset est de la forme :

$$\mathrm{SNDR} = 20\log_{10}\left(\frac{2^{B-1}}{\sqrt{2}\sigma_o^{\mathrm{LSB}}}\right)$$

où B est le nombre de bits du TIADC et σ_o^{LSB} est l'écart-type de la distribution des offsets des sous-ADCs, exprimée en LSB. Il est aussi intéressant de constater que la hauteur de chaque raie d'offset est liée à la magnitude des coefficients de la série de Fourier associée aux valeurs des offsets des sous-ADCs.

Les effets des désappariements de gain sont différents. Cette fois, ce sont des réplications du signal autour de fréquences multiples de F_s/M qui viennent dégrader le signal de sortie du TIADC. Ainsi, si le signal d'entrée est large bande, le bruit dû au désappariements de gain sera lui aussi large bande. Comme pour le cas des désappariements d'offset, on peut montrer que la puissance de chaque réplica est liée à la magnitude des coefficients de la série de Fourier associées aux valeurs de gains des sous-ADCs. La valeur du SNDR

en présence de désappariements de gains peut être exprimée de la façon suivante :

$$\text{SNDR} = 10\log_{10}\left(\frac{\bar{g}^2}{\sigma_g^2}\right) = -20\log_{10}\left(\bar{\sigma}_g\right)$$

où $\bar{g}$ est la moyenne des gains des sous-ADCs et σ_g est leur écart-type. Ainsi, on remarque que, bien que les effets des désappariements de gain soient liés au signal d'entrée, le SNDR ne dépend ni de la fréquence, ni de l'amplitude du signal.

De façon similaire, les désappariements d'instants d'échantillonnages génèrent, dans le spectre du signal de sortie, des réplications sur spectre du signal d'entrée autour des fréquences multiples de F_s/M. Cette fois, le SNDR dépends à la fois du niveau des désappariements mais aussi de la fréquence du signal d'entrée. Lorsque les décalages de temps sont petits par rapport à la période d'échantillonnage, il peut s'exprimer :

$$\text{SNDR} = -20\log_{10}\left(\omega_0\sigma_r\right)$$

où w_0 est pulsation du signal d'entrée et σ_r l'écart type des décalages d'instant d'échantillonnage des sous-ADCs. On voit logiquement que la dégradation de SNDR est plus importante lorsque la fréquence d'entrée du signal est élevée. En effet, un signal variant rapidement est plus affecté par une erreur d'échantillonnage qu'un signal variant lentement.

Les désappariements de bande passante, liés à des différences de constante de temps entre les filtres d'entrée du premier ordre des sous-ADCs, occasionnent le même type de bruit que les désappariements de gain et d'instant d'échantillonnage. En effet, ils créent une combinaison d'un désappariement de gain et d'un désappariement de phase, dépendant de la fréquence du signal d'entrée. Lorsque les décalages de constante de temps sont petits par rapport à la valeur de la constante de temps nominale, le SNDR peut s'exprimer :

$$\text{SNDR} = 20\log_{10}\left(\frac{\sqrt{1+(\omega_0\bar{b})^2}}{\omega_0\bar{b}\sigma_\beta}\right)$$

où ω_0 est la pulsation du signal d'entrée, $\bar{b}$ est la moyenne des constantes de temps et σ_β leur écart-type.

Quantifier les dégradations du signal dues aux désappariements est primordial car cela permet de concevoir des méthodes de calibrations adaptées aux performances à atteindre. Comme démontré dans le Chapitre 3, cela permet aussi d'analyser les performances des méthodes de calibration selon les différents paramètres.

Calibration numérique des désappariements

Les résultats présentés dans Chapitre 3 sur les techniques de calibration s'ajoutent aux travaux précédents sur le sujet tout en s'en inspirant.

Conditions sur le signal d'entrée

Dans les précédents travaux sur la calibration aveugle des désappariements, le signal est généralement supposé stationnaire au sens large (SSL), ce qui n'est en pratique pas vérifié par les signaux de communication, généralement constitués de plusieurs canaux de faible largeur de bande. Une autre condition généralement admise est que le spectre du signal d'entrée ne doit pas contenir de raies à des fréquences multiples de F_s/M.

L'étude présentée dans ce mémoire généralise ces hypothèses aux cas de signaux issues de processus aléatoires non stationnaires, plus réalistes dans le contexte des systèmes de télécommunications large bande. L'espérance et l'autocovariance d'un processus stationnaire varient dans le temps et peuvent donc être représentées par leur transformée de Fourier. La condition pour les que les techniques de calibration proposées dans ce mémoire fonctionnent est que les transformées de Fourier de l'espérance et de l'autocovariance ne contiennent pas de raie à des fréquences multiples de F_s/M. Il est démontré que cela est le cas lorsque le signal d'entrée est une somme de canaux ayant subis une mise en forme d'impulsion, et modulés à différentes fréquences porteuses.

Structure de calibration

L'idée du système de calibration proposé est d'enchaîner des blocs de calibration chacun dédié à un type de désappariement. L'ordre des blocs de calibration ne peut cependant pas être choisi de façon arbitraire. La calibration des désappariements d'offset doit avoir lieu en premier car la présence de désappariements d'offset perturberait la détection des autres types de désappariement. La calibration des désappariements de bande passante suit le bloc calibration des désappariements d'offset. En effet, les désappariements de bande passante occasionnent des désappariements de gain et de phase qui dépendent de la fréquence du signal d'entrée. Ces erreurs de désappariement doivent être corrigées pour ne pas perturber la détection des désappariements de gain statique et celle des désappariements d'instant d'échantillonnage. L'ordre des deux blocs de calibration restant est aussi important. La calibration des désappariements de gain n'étant pas perturbée par des décalage d'instant d'échantillonnage, c'est celle qui doit être effectuée en premier. Le dernier bloc de calibration est donc celui de la calibration des désappariements d'instant d'échantillonnage.

Calibration des désappariements d'offset

La calibration des désappariements d'offset se fait en deux temps. Dans un premier temps, l'offset de chaque sous-ADC est estimé. Une fois les offsets des sous-ADCs connus, le signal est corrigé.

L'estimation des offsets se fait en calculant la moyenne du signal de sortie de chaque sous-ADC. Si le signal d'entrée est supposé de moyenne nulle, alors la moyenne du signal de sortie de chaque sous-ADC devrait aussi être nulle. En présence de désappariements d'offset, la moyenne du signal de sortie de chaque sous-ADC converge vers la valeur de l'offset correspondant. Suivant le type du signal d'entrée et les performances voulues, le temps de moyennage peut varier. Pour un signal de télécommunication large bande typique, la longueur de moyennage peut atteindre 50 millions d'échantillons par sous-ADC.

Une fois l'offset de chaque sous-ADC connu, le signal est corrigé en soustrayant l'offset estimé au signal de sortie de l'ADC correspondant. Afin d'éviter les désappariements résiduels dus à la quantification de l'offset estimé, il peut être nécessaire de coder le signal de sortie sur un plus grand nombre de bits. Une séquence aléatoire peut aussi être rajoutée au moment de la correction pour étaler le bruit résiduel sur toute la bande fréquentielle.

Calibration des désappariements de gain

Similairement à la calibration des désappariements d'offsets, la calibration des désappariements de gains se fait en deux temps. Tout l'abord on estime le gain de chaque sous-ADC relativement à un sous-ADC de référence puis ensuite le signal de sortie de chaque sous-ADC est corrigé.

L'estimation du gain relatif de chaque sous-ADC par rapport au sous-ADC de référence se fait en calculant la puissance moyenne en sortie de chaque sous-ADC. Lorsqu'il n'y a pas de désappariements de gain, la puissance moyenne de sortie de chaque sous-ADC est la même. En présence de désappariements de gain, les puissances moyennes des sous-ADCs sont différentes et le ratio entre la puissance du signal de sortie d'un sous-ADC et la puissance de sortie du sous-ADC de référence donne le gain relatif du sous-ADC. Le calcul de la puissance moyenne de sortie de chaque sous-ADC se fait en moyennant les carrés des échantillons de sortie du sous-ADC. Pour simplifier l'implémentation, on peut préférer calculer la moyenne des valeurs absolues des échantillons. Comme pour l'estimation des désappariements d'offset, la longueur de moyennage doit prendre en compte les spécifications à atteindre et le type de signal d'entrée. Les simulations montrent que dans des cas typiques, 1 million d'échantillons par sous-ADC peuvent être nécessaire.

Une fois les gain relatif de chaque sous-ADC connu, la correction est simple et peut coûteuse puisqu'elle se fait en divisant le signal de sortie de chaque sous-ADC par le gain correspondant.

Calibration des désappariements d'instant d'échantillonnage

La calibration des désappariements d'instant d'échantillonnage est plus complexe, et donc plus coûteuse que les calibration, relativement simples, des désappariements de gain et d'offset. La technique proposée dans ce mémoire utilise le fait que les décalages d'instant d'échantillonnage sont petits comparés à la période d'échantillonnage globale du TIADC. Cela permet de linéariser les modèles en faisant des approximations de Taylor au premier ordre des équations.

Ainsi, lorsque les décalages d'instant d'échantillonnage sont petits, l'erreur occasionnée par un décalage de temps sur un sous-ADC est proportionnelle au décalage de temps et à la dérivée du signal, les termes du second ordre étant négligeables. Dès lors que l'on connaît le décalage de temps de chaque sous-ADC et la dérivée du signal, on peut calculer l'erreur d'échantillonnage et la soustraire au signal de sortie du TIADC. En pratique, il est impossible de connaître la dérivée du signal avec exactitude car seul le signal échantillonné est connu. Cependant, il est possible d'utiliser un filtre à réponse impulsionnelle finie (FIR pour *Finite Impulse Response*) pour estimer la dérivée du signal. Suivant le nombre de coefficients du filtre, l'estimation de la dérivée du signal est plus ou moins en précise. En particulier, augmenter le nombre de coefficients du filtre permet d'augmenter la fréquence jusqu'à laquelle le filtre est précis. Malheureusement, augmenter le nombre de coefficients augmente la complexité du filtre et donc son coût en composants lorsqu'implémenté dans un circuit. Pour cette raison, le nombre de coefficients est une compromis entre coût d'implémentation et précision de correction. En pratique, un filtre avec 50 coefficients permet d'estimer la dérivée du signal avec une erreur en magnitude inférieure à 0.1% jusqu'à 90% de la fréquence de Nyquist.

L'estimation des décalage de temps utilise le fait que, lorsqu'il n'y a pas de désappariements d'instant d'échantillonnage, le signal de sortie de chaque sous-ADC est "orthogonal" au signal de sortie du TIADC filtré par un filtre FIR dont les coefficients sont impairs. Ici l'orthogonalité signifie que la moyenne du produit du signal de sortie de chaque sous-ADC avec le signal de sortie du TIADC filtré tends vers 0. On peut montrer qu'en présence de disparités d'instant d'échantillonnage, l'orthogonalité n'est plus vérifiée. La moyenne du produit signal sous-ADC/signal filtré n'est pas égale à 0 mais dépends des décalages de temps des sous-ADCs adjacents au sous-ADC considéré. Cette dépendance est linéaire lorsque les décalages d'instant d'échantillonnage sont petits par rapport à la période d'échantillonnage. On obtient donc un système linéaire de M équations mettant

en relation les produits signal sous-ADC/signal filtré et les décalages de temps des sous-ADCs. Ce système d'équations peut être réécrit de sorte à exprimer le décalage de temps de chaque sous-ADC par rapport à celui d'un sous-ADC de référence, ce qui permet de l'inverser. On obtient ainsi les valeurs des décalages de temps en fonction des valeurs des produits signal sous-ADC/signal filtré. Comme dans les calibrations précédemment mentionnées, la longueur de moyennage est importante puisqu'elle détermine la précision de l'estimation. Ainsi pour des cas typiques d'application, le nombre d'échantillons par sous-ADC requis pour atteindre des performances correctes est de l'ordre de quelques millions.

Bien sûr, que ce soit pour la correction ou pour l'estimation, les approximations du premier ordre qui sont faites limitent la performance. En pratique on constate que les décalages de temps relatifs à la période d'échantillonnage augmentent lorsque la fréquence d'échantillonnage du TIADC augmente. Or, plus le TIADC est rapide, moins sa résolution est importante et moins la calibration des désappariements d'instant d'échantillonnage a besoin d'être précise. De ce fait, une calibration au premier ordre est souvent adaptée aux besoins de performance des TIADCs modernes.

Calibration des désappariements de bande passante

La technique calibration des désappariements de bande passante proposée dans ce mémoire est adaptée au cas où les filtres d'entrée des sous-ADC peuvent être assimilés à des filtres passe-bas du premier ordre. Comme pour la calibration des désappariements d'instant d'échantillonnage, elle tient compte du fait que les décalages de constante de temps des filtres sont petits par rapport à la constante de temps nominale.

La correction des désappariements de bande passante est relativement similaire à celle des désappariements d'instant d'échantillonnage. Ainsi, lorsque les décalages de constante de temps sont petits, l'erreur d'échantillonnage apparaissant à la sortie d'un sous-ADC est proportionnel au décalage de constante de temps et à la "dérivée passe-bas" du signal. La dérivée passe-bas du signal est ainsi dénommée car elle s'obtient en filtrant le signal de sortie du TIADC par un filtre FIR dont la réponse fréquentielle est le produit de la réponse fréquentielle d'un filtre passe-bas et de la réponse fréquentielle d'un filtre différenciateur. Lorsque l'on connaît la "dérivée passe-bas" du signal et les décalage de constante de temps des sous-ADCs, on peut donc calculer les erreurs d'échantillonnage associées et les soustraire au signal de sortie du TIADC. Comme dans le cas, du filtre différenciateur, le nombre de coefficients du filtre "dérivée passe-bas" détermine la fréquence maximale jusqu'à laquelle le filtre est précis. Il doit donc être déterminé en fonction des caractéristiques du signal d'entrée et des performances à atteindre.

Étant donné, que les désappariements de bande passante occasionnent des désappa-

riements de gain et de phase, détecter l'un ou l'autre de ces effets permet de détecter les désappariements de bande passante. L'approche choisie dans ce travail est détecter les désappariements de bande passante en mesurant les désappariements de gain qu'ils occasionnent. Cela peut être fait de manière assez similaire à l'estimation des désappariements de gain statique en faisant des mesures de puissance moyenne en sortie de chaque sous-ADC Cependant, il est important que la calibration soit immune aux désappariements de gain statique. Afin de supprimer cette dépendance, la solution proposée consiste à mesurer la puissance moyenne de deux versions filtrées du signal de sortie de chaque sous-ADC. Étant donné que les désappariements de bande passante occasionnent des désappariements de gain dépendant de la fréquence, les puissances correspondant à ces deux versions filtrées contiennent de l'information sur les désappariements de gain correspondant à deux régions fréquentielles différentes. En revanche, les gains statiques affectent ces mesures de puissances de façon exactement similaire, et en prenant le ratio des puissances des signaux filtrés, on peut éliminer la contribution des désappariements de gain statique. En pratique chaque sous-ADC est filtré par des filtres de type "sinus" et "cosinus", dénotés ainsi du fait de la forme de leur réponse fréquentielle. Une fois que les ratios de puissance sont calculés pour chaque sous-ADC, l'erreur de décalage de constante de temps relative à la constante de temps nominale peut être estimée de manière directe, en faisant l'approximation que que les décalages de constante de temps sont petits. La condition pour que cette technique de calibration soit fonctionnelle est bien sûr que le signal soit suffisamment large bande pour que les puissances des signaux filtrés contiennent suffisamment d'information sur les désappariements de gain dépendant de la fréquence. Comme pour l'estimation des autres types de désappariements, la précision de estimation des désappariements de bande passante dépend du nombre d'échantillons pris en compte dans le calcul de la puissance moyenne. En pratique, on constate que plusieurs millions d'échantillons par sous-ADC sont nécessaires pour atteindre des performances suffisantes.

Implémentation d'un circuit intégré de test

Les techniques de calibration décrites précédemment ont été implémentées sur un circuit de test afin de valider leur principe théorique. Un TIADC de 1.6 GS/s comprenant 12 sous-ADCs a permis de valider les calibrations numérique des désappariements d'offset, de gain et d'instant d'échantillonnage. Le système de calibration permet notamment une réduction des raies de désappariement à un niveau inférieur à -70 dB par rapport à un signal sinusoïdal pleine échelle pour des fréquence d'entrée jusqu'à 750 MHz.

Partie analogique

Le TIADC est constitué de 12 SAR ADC fonctionnant à 135 ME/s. L'architecture SAR (*Successive Approximation Register*) a été choisie pour son efficacité énergétique particulièrement bonne dans cette zone de fréquence. Bien qu'aucune technique de réduction des désappariements n'ait été mise en place du point de vue analogique, chacun des sous-ADC a été conçu de façon minutieuse.

Ainsi le condensateur d'échantillonnage, constitué de sous-condensateurs de capacité un multiple d'une capacité unitaire, a été dessiné spécifiquement. Il prend la forme d'un peigne métallique dans lequel sont insérés des doigts, le nombre de doigts déterminant la valeur de chaque condensateur. Cela permet notamment d'augmenter la compacité tout en maintenant un bon niveau d'appariements entre les condensateurs constituant un même sous-ADC. L'échantillonnage du signal se fait en utilisant la technique dite de "bottom plate sampling". La quantification à proprement parlée est effectuée par approximations successives en utilisant la technique dite MCS (pour *Merge Capacitive Switching* an anglais). Elle consiste à utiliser une tension de référence milieu, une tension de référence haute et une tension de référence basse, qui peuvent être connectées aux sous-condensateurs, afin de comparer le signal échantillonné à différents niveaux par dichotomie.

Enfin un suiveur en entrée du TIADC permet de transmettre différentiellement le signal d'entrée aux différents sous-ADCs, tout isolant les sous-ADCs du circuit de génération du signal.

Calibration numérique

Le signal numérisé est ensuite transmis aux différents blocs de calibration numérique. La calibration des désappariement d'offset est effectuée en premier, puis elle est suivie par la calibration des désappariements de gain, et enfin la calibration des désappariements d'instant d'échantillonnage est effectuée.

Un des principaux avantages des techniques de calibration numériques, comparées aux techniques de calibration mixtes, est leur rapidité de mise en place. Cela est critique dans un environnement où le temps de mise sur le marché est un aspect essentiel de la compétitivité des entreprises. Ainsi le processus de conception utilise la synthèse au niveau (HLS pour *High Level Synthesis* en anglais). Cela permet, à partir d'un programme écrit en langage C, de générer du code RTL synthétisable. La simulation du code se fait avec des simulateurs C classique bien plus rapide que les simulateurs de code RTL. De plus, le langage C permet de décrire les algorithmes avec un grand niveau d'abstraction, ce qui permet d'adapter les blocs de calibration facilement à n'importe quelle architecture

de TIADC.

Les autres avantages d'effectuer la calibration dans le domaine numérique tiennent au fait des avantages de la conception numérique par rapport à la conception analogique de manière générale. Ainsi, la conception numérique bénéficie d'outils de conception presqu'entièrement automatisés qui permettent de générer rapidement l'architecture physique du circuit à partir d'une description relativement haut niveau (langage RTL). De plus, l'évolution d'un noeud technologique à un autre est relativement simple car le code ne nécessite pas d'être changé, ce qui rend les blocs de calibration très portables.

La calibration des désappariements de bande passante n'est pas intégrée dans le circuit car les désappariements de bande passante ne sont pas limitant dans ce circuit. Elle fera cependant l'objet de validation sur circuit dans un futur relativement proche.

Performances du circuit en fonction de la fréquence

Les performances des algorithmes de calibration, après avoir été validés par des simulations Matlab, on été vérifiés à travers des mesures de performance du circuit de test implémentant le TIADC.

Le test de mesure le plus classique pour un ADC est d'observer le spectre de sortie lorsque le signal d'entrée est un signal sinusoïdal pleine échelle. En faisant varier la fréquence du signal d'entrée, on peut mesurer des indicateurs de performance tels que le SNDR, la SFDR, la hauteur des raies harmoniques, et surtout le niveau des raies de désappariement.

Afin d'évaluer les performance du bloc de calibration numérique, les spectres du signal de sortie avant et après calibration de désappariement on été mesurés. Ces mesures montrent une forte réduction des raies de désappariement sur une bande fréquence allant jusqu'à 750 MHz. Par exemple, un gain de 5 dB de SNDR est obtenu pour une fréquence d'entrée de 750 MHz. De la même façon, les raies de désappariements après calibration restent inférieures à -70 dBFS jusqu'à 750 MHz de fréquence d'entrée. Au dessus de 750 MHz, on observe une baisse de performance de la calibration du fait du choix de conception effectué pour le filtre différenciateur inclus dans le bloc de calibration des désappariements d'instant d'échantillonnage. En effet, le nombre du coefficients du filtre a été choisi de telle sorte que sa fréquence limite soit autour de 750 MHz, ce afin de limiter sa complexité.

Au final, la SFDR du TIADC est limitée par les raies harmoniques dues au nonlinéarités du suivre d'entrée, et non par les raies dues aux désappariements.

Bien que les ADC soient caractérisés avec des signaux sinusoïdaux, ces types de signaux reflètent mal la réalité des signaux de communication. Les résultats de test du circuit avec des signaux modulés démontrent l'efficacité de la calibration en réduisant le

niveau de bruit de façon importante.

Évolution des niveaux de désappariement avec la température

Afin de justifier l'emploi d'une technique de calibration fonctionnant en arrière-plan (*background calibration* en anglais), des mesures du circuit à différentes températures ont été effectuées. En effet, le niveau des désappariements change avec la température et il est important de savoir si ces changements justifient une calibration fonctionnant en continu.

L'avantage des algorithmes de calibration proposés est qu'ils fonctionnent tous de manière directe et permettent d'obtenir la valeur estimée des désappariements relatifs de chaque sous-ADC. Connaissant la hauteur des différentes raies, on peut aussi déduire l'écart-type de chaque type de désappariement, en inversant les équations données dans le Chapitre 2. En utilisant ces deux résultats, on peut donc mesurer le niveau global des désappariements et aussi avoir une idée du désappariement relatif d'un sous-ADC par rapport à une autre.

Les résultats des mesures montrent que le niveau des désappariements est affecté par la température. D'une part, l'écart-type des désappariements parmi les sous-ADC varie mais les variations ne sont pas nécessairement homogène parmi les sous-ADC. En pratique, on remarque on la variation d'écart-type seule n'est pas suffisante pour justifier l'utilisation d'une calibration en continu. En revanche, les variations locales d'un sous-ADC à l'autre peuvent être parfois plus importantes et les désappariements nécessitent d'être ajustés en continu.

Comparaison à l'état de l'art

Le circuit décrit dans ce manuscrit a été réalisé en technologie CMOS 40nm de chez STMicroelectronics et sa surface est de 0.83 mm^2. Environ 40 % de cette surface est dédiée au bloc de calibration numérique alors que les 60 % restant sont dédiés à la partie analogique.

Le circuit consomme une puissance de 283 mW sous une alimentation de 1.1 V. Environ 45% de cette puissance est allouée à la conversion analogique-numérique en tant que telle (partie analogique)et 55 % sont utilisés par la calibration numérique. Une grande partie de cette énergie est consommée par le filtre numérique permettant de calculer la dérivée du signal.

Il s'agit d'une surcharge de consommation importante mais qui peut se relativiser lorsque l'on compare les performances du circuit à d'autres circuits qui ont des caractéristiques similaires. Le tableau ci-dessous compare le circuit proposé à d'autres TIADC de l'état de l'art. Malgré la consommation de la calibration, l'efficacité énergétique (FOM

	ISSCC 2013 [12]	JSSC 2011 [10]	VLSI 2012 [11]	ISSCC 2014 [17]	**ISSCC 2014 [13]**
Technologie CMOS	65 nm	65 nm	65 nm	65 nm	**40 nm**
Fréquence d'échantillonnage [GE/s]	3.6	2.6	2.8	1.0	**1.6**
Cal. offset	oui	oui	oui	oui	**oui**
Cal. gain	oui	oui	non	non	**oui**
Cal. instant d'échantillonnage	non	non	oui	oui	**oui**
Plus haute raie de désappariement	50	55	60	60	**70**
SFDR [dBFS]	50	55	55	60	**62**
THD [dB]	-55	-58	-55	–	**-58**
SNDR [dB]	47	49	48	51.4	**48**
Puissance [mW]	795	480	44.6	19.8	**93**
Walden FOM [fJ/conv]	1207	801	76	62.3	**283**
Surface [mm^2]	7.4	5.1	0.63	0.78	**0.83**

Table 1 – Comparaison avec l'état de l'art

pour *Figure of Merit* en anglais) reste acceptable, et en dessous de la plupart des autres circuits. Après calibration, la hauteur des raies de désappariement est plus faible de 10 dB par rapport à la meilleure de performances publiées jusqu'à maintenant.

Conclusion et perspectives

Le travail présenté dans ce mémoire utilise de façon extensive le fait que les désappariements sont petits. Cela permet de notamment d'écrire des systèmes d'équations linéaires aisément solvable de façon directe, sans nécessiter d'utiliser de techniques adaptatives, ou fonctionnant par approximations successives. Alors que cette technique a ici été utilisée dans le cas des désappariements d'instant d'échantillonnage et de bande passante, on peut très bien imaginer appliquer le même genre de raisonnement pour d'autres types de désappariements, par exemple des désappariements de bande passante qui ne sont pas du premier ordre.

Il faut cependant garder à l'esprit, qu'avec les performances des TIADCs s'améliorant sans cesse (fréquence d'échantillonnage et résolution), il est possible que les effets du second ordre ne soient plus négligeables dans le futur. Cela nécessitera vraisemblablement d'étendre les méthodes proposées dans ce mémoire.

On peut par exemple imaginer utiliser des filtres de correction plus complexes mais cela rajouterait une pénalité de consommation non négligeable. Une autre approche pourrait être de s'aider d'une correction de type mixte basique. Par exemple, les travaux présentés ici montrent que la calibration numérique permet de réduire les niveaux de désappariements d'instant d'échantillonnage à des niveaux très faibles, lorsque les désappariements

initiaux étaient petits. Il est fort possible qu'atteindre ce niveau de précision en ajustant des éléments analogiques sera très difficile. Une meilleure stratégie pourrait être d'effectuer une correction analogique grossière des décalages d'instant d'échantillonnage. Cela permettrait de réduire le niveau des désappariements à un seuil que la calibration numérique est en mesure de gérer très précisément à un coût acceptable.

La calibration des désappariements de bande passante sera probablement sujette au même type de décision dans un future distant. Pour l'instant, les désappariements de bande passante ne limitent pas encore les performances des TIADCs actuels, mais avec l'augmentation des vitesses d'échantillonnage, il est fort à parier que ce ne sera plus le cas d'ici à quelques années. Cela sera bien sûr une bonne opportunité de tester une implémentation sur circuit des algorithmes de calibration des désappariements de bande passante proposés dans ce mémoire.

Enfin, la partie théorique sous-jacente à ce travail a permis de définir les conditions de fonctionnement des algorithmes de calibration, en montrant qu'ils fonctionnaient pour des signaux non stationnaires, à certaines conditions sur leur fonction d'autocovariance. Bien que certaines réponses soient apportées ici, il serait très intéressant de s'intéresser de plus prêts aux caractéristiques statistiques de signaux de communication numérique large bande réels (comme les signaux de TV par câble ou les signaux de communication satellitaires). A terme, cela permettrait d'avoir un cadre théorique bien défini pour la validation des algorithmes de calibration.

cfchapterVarious mathematical demonstrationsappendix/appendixAappendixA.tex