PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

A BLIND CALIBRATION SCHEME FOR SWITCHED-CAPACITOR PIPELINE ANALOG-TO-DIGITAL CONVERTERS

JUAN ANDRÉS BOZZO JIMÉNEZ

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ANGEL ABUSLEME

Santiago de Chile, March 2020

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

A BLIND CALIBRATION SCHEME FOR SWITCHED-CAPACITOR PIPELINE ANALOG-TO-DIGITAL CONVERTERS

JUAN ANDRÉS BOZZO JIMÉNEZ

Members of the Committee:
ANGEL ABUSLEME
CHRISTIAN OBERLI
MARCELO GUARINI
LUIS FERNANDO ALARCÓN

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, March 2020

Dedicated to my parents, María Cristina and Mauricio.

There are only two lasting bequests we can hope to give our children.

One of these is roots, the other, wings.

JOHANN WOLFGANG VON GOETHE

ACKNOLEGMENTS

I thank my thesis advisor Angel Abusleme from the Department of Electrical Engineering of Pontifica Universidad Católica de Chile. He guided me through the development of this work even if my specific topic sometimes put him in a uncomfortable place outside of his field of expertise. He also was available for giving advice or meeting outside of working hours when something urgent had to be resolved. And had infinite patience for listening to my convoluted presentations when things failed with no clear reason.

I would also like to thank professor José Silva Martínez from the Department of Electrical and Computing Engineering of Texas A&M University (TAMU), who brought about the initial idea of the calibration algorithm and tutored me during my stay at TAMU. He helped me at the inception of the algorithm, even when faced with a tight schedule.

My thanks to Enrique Álvarez, currently working on his PhD. at the University of California, San Diego. After having the willingess to read a 15-page long document, he steered this project into a working algorithm with an indication of just two sentences. I was not able to figure out the issue by myself at that time.

Finally, I would like to thank my family and friends, which had more faith in me than myself and are probably happier at the fact that I'm finally graduating.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

RESUMEN

Este trabajo presenta un nuevo esquema de calibración para conversores de datos análogo-a-digital (ADC). La calibración en *foreground* está diseñada para auto-calibrar un ADC *pipeline* de capacitores conmutados. La calibración estima los parámetros eléctricos del ADC; los capacitores de las etapas *pipeline*, referencias de voltaje y pérdida de carga debido a ganancia finita de los amplificadores. La estimación es usada para compensar digitalmente errores en la conversión durante la operación normal del ADC. Para realizar la estimación de parámetros, el algoritmo de calibración se basa en la instanciación de conversores $\Delta\Sigma$ en la interfaz entre etapas *pipeline* reorganizando los componentes eléctricos de las etapas *pipeline*. Diferentes configuraciones pueden ser probadas usando señales de entrada auto-generadas, lo que permite al algoritmo inferir los parámetros eléctricos subyacentes.

La calibración es realizada por la misma circuitería que opera durante el funcionamiento normal del conversor y no requiere de un circuito o voltage que actúe como *tierra real*. Sin embargo depende de los voltajes the *threshold* de los sub-ADCs de las etapas *pipeline*, haciéndolo inadecuado para ciruitos donde el error de los sub-ADCs es dominante. El comportamiento de un número de ADCs de 10 bits con una *ENOB* de 6.3 bits fué simulado y una mejora en resolución entre 2.5 bits para el mejor caso y 1 bit para el peor fué observada.

Palabras Claves: Modulación delta-sigma, conversión análoga-digital, medición de variables eléctricas, estimación de parámetros, optimización

ABSTRACT

In this work, a novel analog-to-digital converter (ADC) calibration scheme is presented. The foreground calibration is designed to self-calibrate a switched-capacitor pipeline ADC. The calibration estimates the electrical parameters of the ADCs stages capacitors, voltage references and charge loss due to finite amplifier gain. The estimation is used to compensate digitally for errors in conversion. To perform the parameter estimation, the calibration algorithm relies on $\Delta\Sigma$ ADCs that can be instanciated in the interface between pipeline stages by rearreanging the pipeline electrical components. Different arrangements can be tested using self-generated input signals, which enables the algorithm to infer the underlying electrical parameters.

The calibration is performed by the same circuitry that operates during normal conversion and does not require an external circuit or voltage to act as *true ground*. But relies on the sub-ADC threshold voltages of the pipeline stages, making it unsuitable for circuits where error is dominated by the sub-ADC. The behavior of a number of 10-bit ADCs with a mean uncalibrated ENOB of 6.3 bits was simulated, and a resolution improvement between 2.5 bits, for the best case, and 1 bits for the worst, was observed.

Keywords: Delta-sigma modulation, Analog-digital conversion, Electric variables measurement, Parameter estimation, Optimization

1. INTRODUCTION

Analog to digital converters (ADCs) are ubiquitous. Even a fully digital device such a cell-phone or computer requires ADCs to read analog, physical, signals in order to be processed by the digital system. As ADCs performances have increased over the past years, design constraints have tightened. Also process, voltage and temperature (PVT) errors have become an important limiting factor in ADCs figures of merit.

1.1 Motivation

In order to compensate for these errors, calibration techniques have been developed (Gines, Peralias, & Rueda, 2009). If the ADC operation has to be stopped for the calibration algorithm to run, an *offline* calibration is being used. Otherwise the calibration algorithm is said to be *online*. To perform the calibration process, usually some parameter or system has to be assumed as true ground, such as a known signal in correlation-based techniques or a high precision converter for calibrations based on channel error identification. This work will test the hypothesis if it is possible to calibrate an ADC without having to assume the value or any external of internal parameter or system. A novel offline algorithm is proposed and analyzed that attempts to perform an ADC calibration following these requirements.

A review of basic ADC and calibration concepts will be presented in the remainder of this chapter. The next chapter describes the ADC architecture required to run the proposed algorithm and the algorithm itself is presented. The following chapter explores the algorithm limitations and implementation strategies. After that chapter, simulation results are reviewed and analyzed. Finally, conclusions are drawn based on these results and future work is introduced.

1.2 ADC concepts

The objective of this section is to become familiar with the basic concepts involving the characterization and operation of an ADC. In this stage, it is best to interpret an ADC as an ideal function that maps an input signal from a continuous input domain into a discrete output domain. The full scale range (FSR) is given by the maximum and minimum input values that the converter is designed to convert from the input domain. Operating a converter outside of this range is known as overload and the output signal usually clips to the a minimum or maximum. This concept can be also applied for sub-systems inside the converter that receive the input in a range of voltages, currents or charges.

1.2.1 Error vs noise

In this work the word error will be used to refer to deterministic sources of distortions during the processing of the input signal. Since these errors are deterministic, they could be predicted if they are measured and modeled and the operation conditions are stable. In the other hand, noise will be used to refer to random processes which behavior can not be predicted and at most be modeled using statistical tools. Examples of this kind of processes are thermal noise, KT noise, shot noise.

1.2.2 ADCs as functions

An ideal ADC with infinite resolution should behave as the identity function with a constant scale factor, it would take the value from the input domain and transform it into the same value but in the output domain. In figure 1.1, an example of an ideal mapping is shown as a plot. On the mapping between these two domains, information can be lost or distorted.

The simplest form of error is a linear distortion. Since the line equation only has two parameters ($y = ax + b$) the linear distortion can be characterized with these two parameters: gain error and offset error. To characterize them, a line is fitted to the function plot minimizing mean square

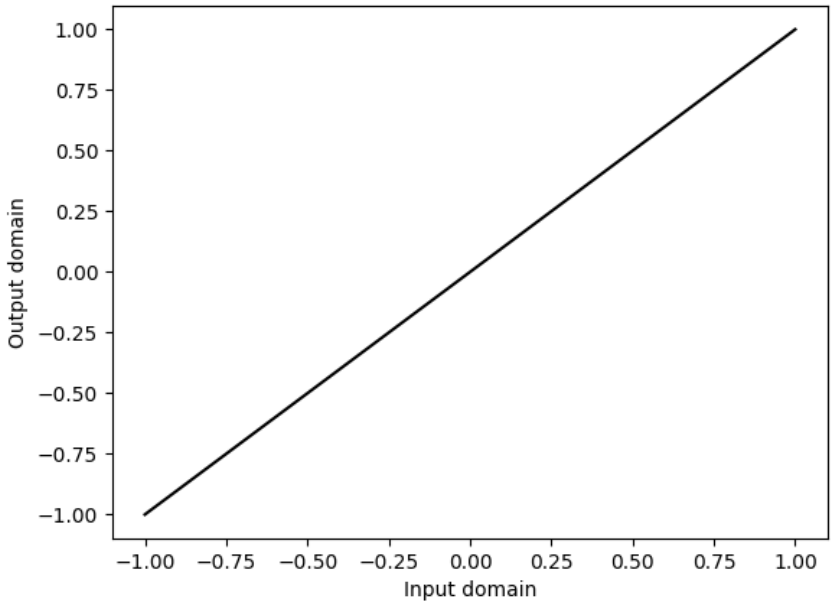


FIGURE 1.1. An example of an ideal mapping.

error. The gain error is the factor that cancels any deviation from the ideal gain. The offset error is the value that must be added to the fit to match the ideal static transfer curve after the gain error has been compensated for. This method of obtaining the linear error is the independently-based method. Another method exists that uses the extreme values, known as the terminal based method ("IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters", 2011).

Non-linear errors are measured after compensating for linear errors. The plot of the mapping encodes such errors but it is usually difficult to interpret them directly from the curve. To better visualize non-linear errors it is preferred to use the Differential non-lineality (DNL) and Integral non-lineality (INL). These measurements are performed after the input signal is quantized, so they will be described later in this chapter.

To characterize the signal distortion, a generic Signal to Noise Ratio (SNR) can be defined. This ratio compares the power of the input signal with the power of the difference between the

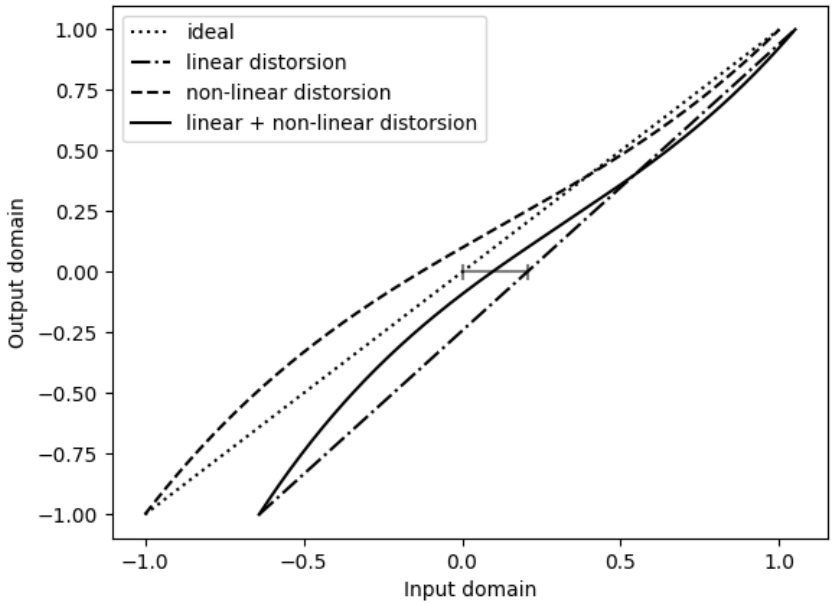


FIGURE 1.2. An example of a distorted mapping with the linear and non-linear components added separately. Offset error is shown as an horizontal bar.

ideal output an the real output generated by the function. To measure the power of a signal; first the constant component is removed, then the squared value of signal is integrated or summed for discrete signals and finally it is divided by the support length of the signal, usually time. To measure constant signals, an alternative method is used, since all constant signals would have zero power. The constant value is squared. With the previous description, the formula for the power of a signal is:

$$P\{f\} = \frac{\int_{t=a}^{b} (f(t) - \mu(f))^2 \, dt}{b-a} \qquad \text{For continous signals}$$

$$P\{f\} = \frac{\sum_{t=a}^{b} (f[t] - \mu(f))^2}{b-a} \qquad \text{For discrete signals}$$

with $\mu(f)$ the mean of the function in the $[a..b]$ range. Then a generic SNR would be expressed as follows:

$$SNR = \frac{P\{in\}}{P\{in - out\}}$$

1.2.3 Real ADCs distortions

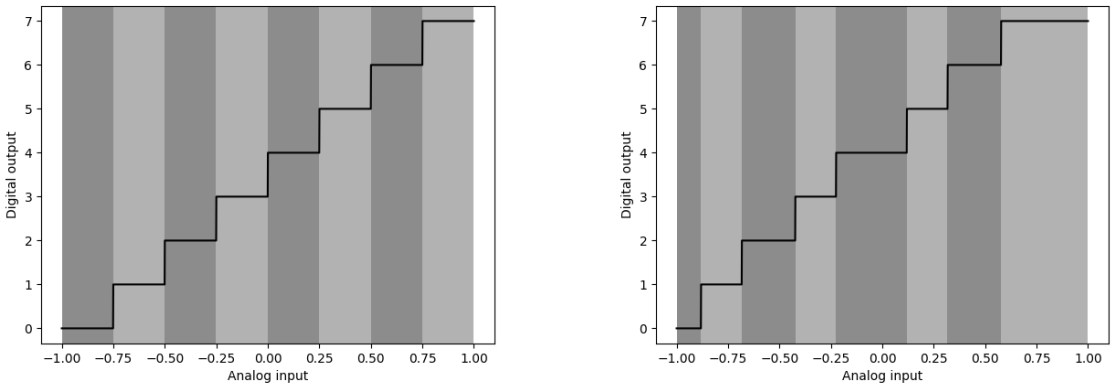
The previous concepts, which are generic mathematical descriptions, are not applied directly to ADC characterization. But they can be translated into the most used concepts. The input domain of the ADC is an analog voltage, current or charge and the output domain are the binary words that fit in the ADCs resolution. Furthermore, the plot of the mapping between the two domains is referred to as the static transfer curve. The static in the name points to the fact that the curve is obtained by obtaining the output value on the asymptotic behavior of the ADC for a sweep of input values. It can be conceptualized as capturing the output of the ADC for an infinitely slow ramp at the input signal.

To fully understand how to characterize the signal-to-noise ratio, it is necessary to introduce quantization, which arises naturally from the fact that an ADC is mapping from a continuous to a discrete domain.

1.2.4 Quantization

Quantization is the process of mapping an input signal which domain is a large set, and mapping it into a smaller output set. In the context of ADCs, the large set is a continuous voltage, current or charge and it is mapped into a discrete binary set. Each word in this binary set is called a code and each code corresponds to a range of values in the original signal domain. This range of values is called a code-bin and the input values at the boundaries of these code-bins are called transition levels. All input values that belong to a code-bin are mapped to the code associated with the code bin. When a code has to be mapped to a single value of the input domain, that value is called the quantized value. Usually the center of the code-bin is assigned as the quantized value. In figure 1.3 the code-bins of an ideal and a distorted quantizer of 3-bit are displayed in

different colors. An ideal and distorted 3-bit quantizer static transfer curve is shown on figure 1.4.



(A) Static transfer curve of an ideal quantizer. (B) Static transfer curve of an distorted quantizer.

FIGURE 1.3. Two 3 bits quantizer code-bins, show in alternating colors.

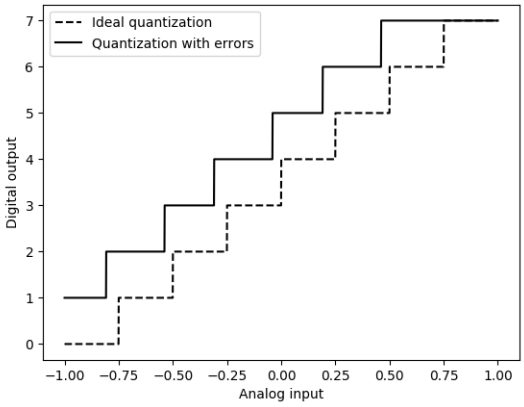


FIGURE 1.4. The static transfer curve of an ideal 3-bit ADC, compared with a distorted 3-bit ADC.

A distortion known as quantization error appears when the input signal is compared to the quantized signal. This error can be characterized by the Signal-to-Quantization Noise Ratio

(SQNR) figure of merit, which is computed as:

$$SNQR = \frac{P\{in\}}{P\{in - quantized\ in\}}$$

The quantization process creates steps in the static transfer curve. For an ideal quantizer, these steps have the same length for all codes with the exception of the extremes, which have infinite length. This length is known as Least Significant Bit (LSB), which is the minimum variation in the input signal that it is still detected by the converter.

If the input signal is random enough, the input signal can be assumed to fill uniformly the code-bin of each code. Then the quantization error can be treated as a uniform random distribution instead of a deterministic value. When the quantization error is treated as a random variable it is referred to as quantization noise. The variance of a uniform distribution is $\frac{LSB^2}{12}$. Then, as an example, for a B-bit quantizer and a sinusoidal input with variance of $2^{2\mathrm{B}-3}LSB^2$ the SQNR, which is the ratio between these variances interpreted as power, is $1.5 \times 2^{\mathrm{B}} = 6.02\mathrm{B} + 1.76dB$.

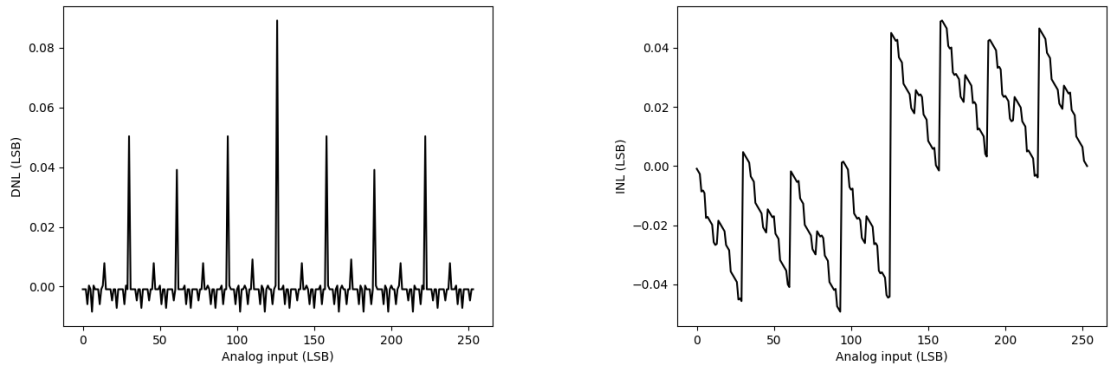
Variations of the SNR are defined to refer to specific types of errors, such as the Signal to Noise and Distorsion Ratio (SNDR) where the output signal includes the errors of static distortions and dynamic noise. This ratio is used to compute the Effective Number Of Bits (ENOB), this figure captures the number of meaningful bits on the output of the ADC. If an ADC code has 16-bits, but the static transfer curve of that ADC mirrors the static transfer curve of an 8-bit ADC, then only the first 8 most significant bits of the 16-bit code will have useful information. The ENOB is computed from the SNDR as $ENOB = \frac{SNDR - 1.76dB}{6.02dB}$.

After quantization, the DNL and INL can be computed. The DNL measures deviations in the step width associated with each code and is compared with the LSB, or average step size for a real ADC: $DNL(k) = \frac{step_k - \mu(step_k)}{\mu(step_k)}$. A positive/negative value for a specific code mean that the code-bin size is bigger/smaller than the ideal code-bin size. A value of -1 for a code in the DNL

means that code is missing and value of 0 means the code-bin has the ideal size. The DNL is undefined for the extreme steps since they have infinite length. The DNL of an 8-bit is depicted in figure 1.5A.

While the DNL measures local distorsions in the step size, the INL measures the global deviation from a specific code with respect to the ideal transfer curve: $INL(k) = \sum_{i=1}^{k-1} DNL(i)$. A value of zero means that the ideal transfer curve and the real transfer curve match at the beginning of that code-bin. A positive value means the real transfer curve is assigning greater codes to that input than the ideal transfer curve, and the opposite for negative values.

The DNL and INL of an 8-bit quantizer is depicted in figure 1.5B. From the DNL it can be seen that in this example the ADC has longer steps in fractions similar to $\frac{k}{2^n}$, the simpler the fraction, the greater the step deviation. And from the INL it can be seen that the first half of the static transfer curve is over an ideal static transfer curve and the second half is below an ideal static transfer curve.



(A) DNL (B) INL

FIGURE 1.5. Example of non-linearities of an 8-bit ADC.

1.2.5 Sampling

Digital systems are not only discrete in states or values but also in time, since they are usually clocked. In an analogous manner in which quantization discretizes values, sampling is the process of discretizing the time domain of a input signal. This is usually done first in the analog domain with the use of track-and-hold circuits. These circuits work in two phases, first an internal node follows the signal (track) and in the second phase that node stores the last value of the first phase (hold). The track-and-hold circuit is usually implemented with a switch that connects a capacitor to the input signal. When the switch is closed, one of the nodes of the capacitor follows the input signal. When the switch is open, the voltage difference of the capacitor is constant since no current can flow. The ADC performs the conversion on the second phase with the input voltage fixed at a constant value. Not all converters need a track-and-hold to process the input signal.

Since the signal is being discretized in the time domain, the bandwidth in the frequency domain is limited to half of the sampling rate by the Nyquist criterion. If this condition is not met by the input signal, aliasing will occur, in which high frequencies outside of the maximum supported frequency, known as Nyquist frequency, appear in the sampled signal. To prevent this, the input usually is conditioned using a low-pass filter.

More information on ADC terminology can be found in “IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters” (2011).

1.3 ADC architectures

A variety of different ADC architectures exists. Some are fast, but usually at the cost of high power consumption or poor SNDR. Other architectures will have trade-off between another set of figures of merit. Murmann (2019) provides a thorough comparison of different ADCs published

in mayor solid-state circuits conferences.

The following sections will introduce the architectures relevant for the proposed algorithm. These are the Flash architecture, the $\Delta\Sigma$ architecture and the Pipeline architecture.

1.3.1 Flash architecture

Flash converters take the analog input signal and distribute it to an array of comparators, each connected to a different threshold voltage, as shown on figure 1.6.

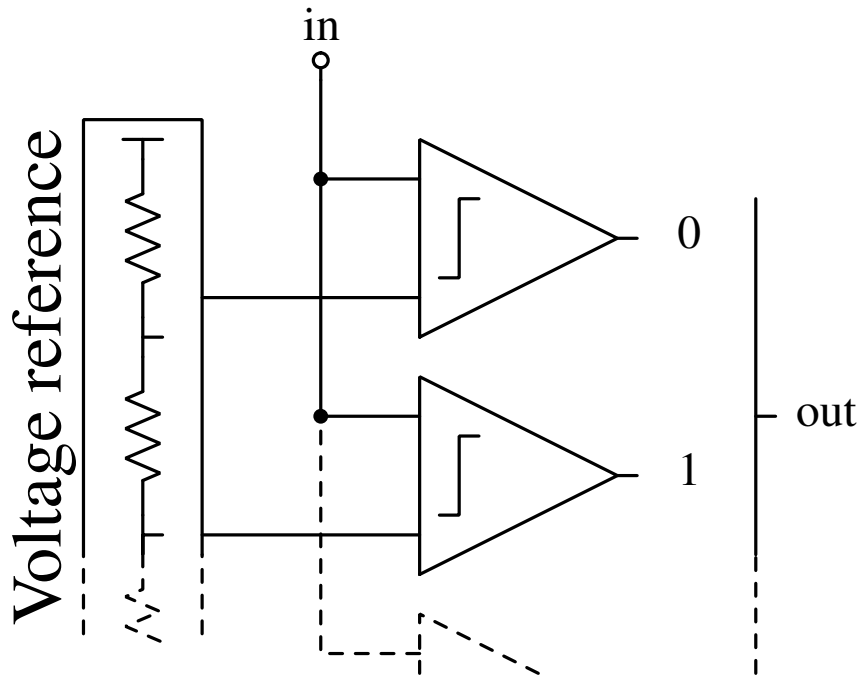


FIGURE 1.6. Flash ADC block diagram.

The output of the ideal flash converter is a string of M bits, each bit associated with a comparator. These indicate that the input voltage is higher than the reference voltage it was compared

to. This means the output of an ideal flash ADC is thermometer coded, if one bit is 1 all preceding it are 1 too. This restricts the output to $M + 1$ possible states which can be represented as an integer between 0 and M corresponding to the number of 1's in the string. Usually this output is then transformed into binary using a decoder. The number of bits required to represent a thermometer number in binary is $N = \lceil log_2(M) \rceil$.

This architecture is the fastest since it takes one clock cycle to convert the input signal and the clock frequency is only limited by the delay of a comparator operation. Other architectures such as pipeline or $\Delta\Sigma$ use a flash ADC as a component. The flash architecture suffers from poor scalability. In order to add an additional bit of resolution, the number of comparators must be doubled, which in turn brings circuit size problems and large capacitances at comparator input, since both of them scale at an order of $O(2^N)$.

Other limiting factor is the comparators offset voltage. The boolean equation for an ideal comparator is $V(a) > V(b)$, with a and b input ports of the device. Because of process mismatch errors, a comparator may have a static error adding or subtracting a constant voltage to the input voltages before comparing, leading to the equation $V(a) + err_a > V(b) + err_b \rightarrow V(a) > V(b) + offset$. This constant offset can be represented as a voltage at the input of one of the comparator ports depicted in the middle of figure 1.7.

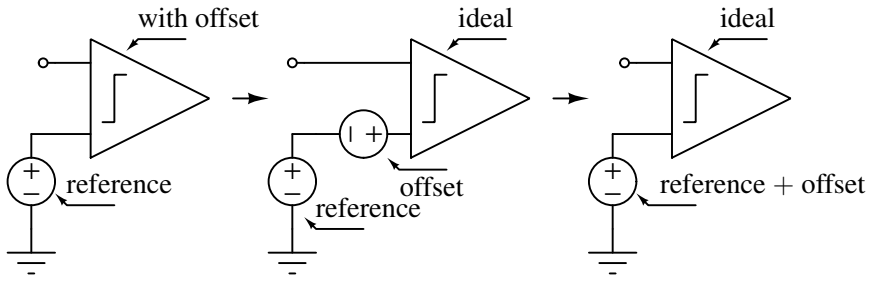


FIGURE 1.7. Comparator with different models for the same offset.

In an ideal flash converter, the reference voltages are uniformly distributed. All comparators would be ideal so the output of a comparator would change exactly at the moment the input signal is greater than the threshold voltage, as seen on figure 1.8. For a more realistic comparator, the reference voltages would not be uniformly distributed and the comparators will have offset. On figure 1.7 it is possible to see how the comparator offset can be lumped into the voltage reference for analysis purposes.

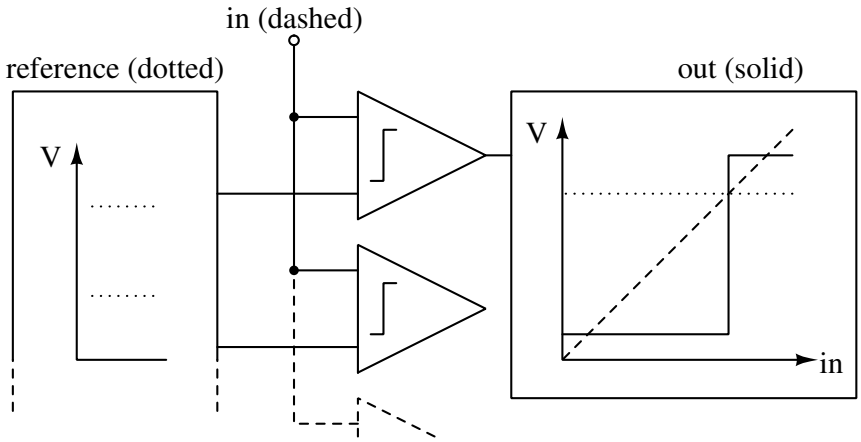


FIGURE 1.8. Flash ADC without errors, the output of a single comparator rises when the threshold voltage is greater than the input voltage.

In figure 1.9 a more realistic model is presented. The effects of the comparator offset and the threshold reference errors are shown as a statistical distribution for many flash ADC instances each with different static errors. These errors propagate to the output, generating a distribution of thresholds for the aggregate where the comparators changes its output.

1.3.2 $\Delta\Sigma$ architecture

The $\Delta\Sigma$ architecture name is a direct reference to its block diagram. As seen on figure 1.10, the difference (Δ) between the analog input signal and the ADC decision is integrated (Σ) to

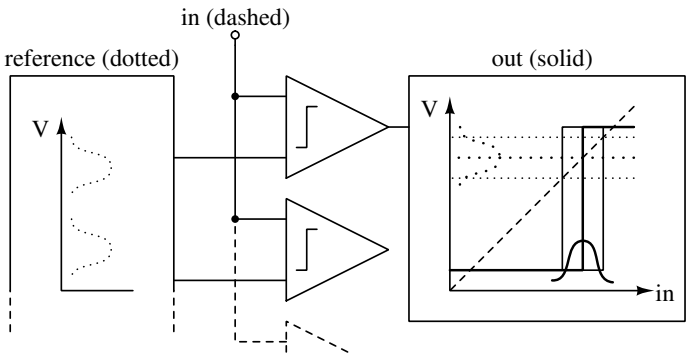


FIGURE 1.9. Aggregate of flash ADC with errors. The input voltage to change the output does not necessarily match the ideal threshold voltage. The tripping point for the extremes and the mean of the distribuiton is shown.

compute the conversion error. The output of the ADC is a stream of bits which have to be filtered, usually by a low-pass and a decimation filter, to obtain a meaningful output. Because of this, the signal has to be sampled at a rate much higher that its Nyquist frequency, making this an over-sampled architecture.

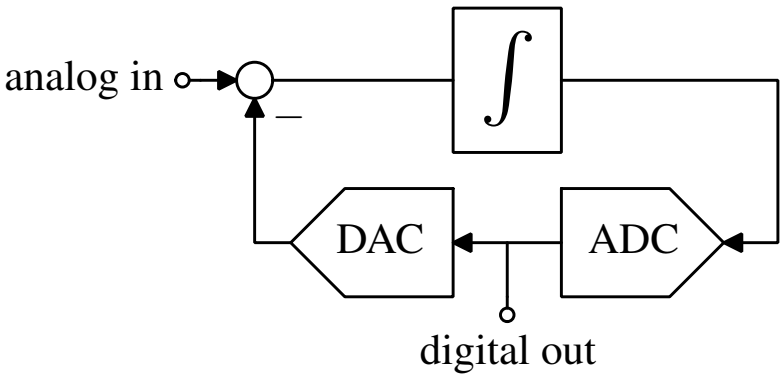


FIGURE 1.10. Block diagram of a $\Delta\Sigma$ ADC.

$\Sigma\Delta$ converters can achieve high resolution by noise shaping. To illustrate this property, the simplest first-order $\Sigma\Delta$ ADC will be used, as seen in figure 1.11. An error, for example quantization error ϵ, is injected at the analog-to-digital conversion section of the loop. The transfer

function of this system, after discretization, is:

$$out = in \underbrace{z^{-1}}_{STF} - \epsilon \underbrace{\left(1 - z^{-1}\right)}_{NTF}$$

where STF is the Signal Transfer Function and NTF is the Noise Transfer Function. Notice that the STF magnitude is 1 for all frequencies. On the other hand, the NTF magnitude grows with frequency, and it is close to 0 at low frequency ($z \rightarrow 1$). This way, if the signal is greatly over-sampled, a low-pass filter will reject most of the noise and keep the signal.

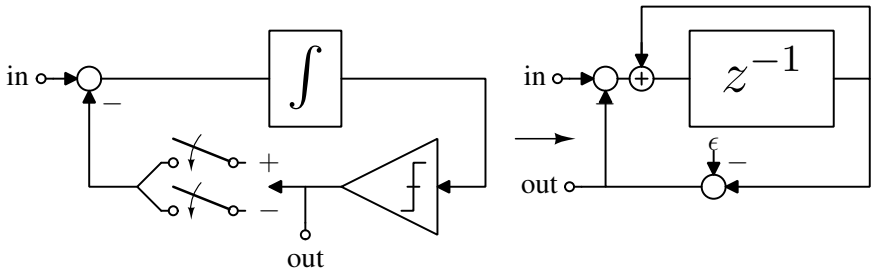


FIGURE 1.11. Block diagram of a 1-bit $\Delta\Sigma$ ADC and simplification.

To achieve high ENOB, the integrator loop must have minimum leakage, as described in Feely and Chua (1991). The integrator leakeage will generate distortion as shown in the static transfer curve of figure 1.12.

An important feature of $\Delta\Sigma$ converters is their insensitivity to errors in the feedforward path for DC signals. This is easy to demonstrate in a first-order $\Delta\Sigma$ loop. If an arbitrary gain G is inserted in the feedfoward loop, as shown in figure 1.13, then the transfer function of the system is:

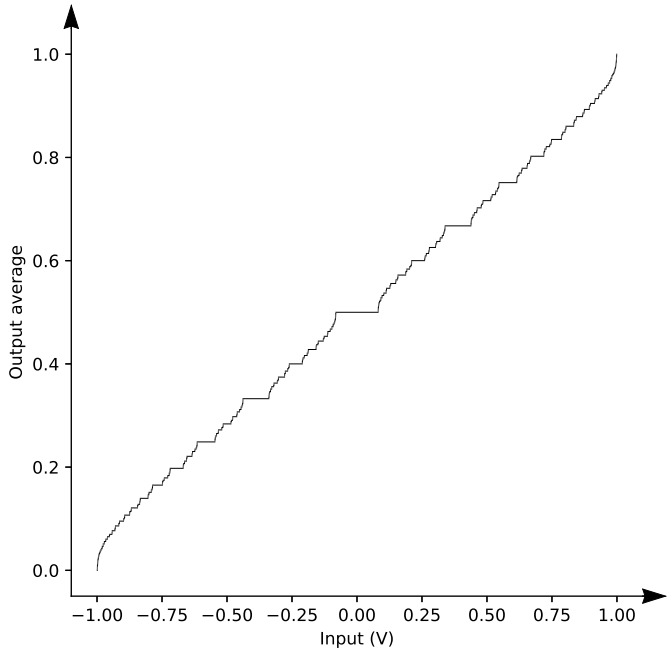


FIGURE 1.12. Static transfer curve for $\Delta\Sigma$ ADC with a 1-bit quantizer and a preservation of charge of 0.85 (1 = no leak).

$$out = \frac{Gz^{-1}}{1-(1-G)\,z^{-1}}in - \frac{1-z^{-1}}{1-(1-G)\,z^{-1}}\epsilon$$

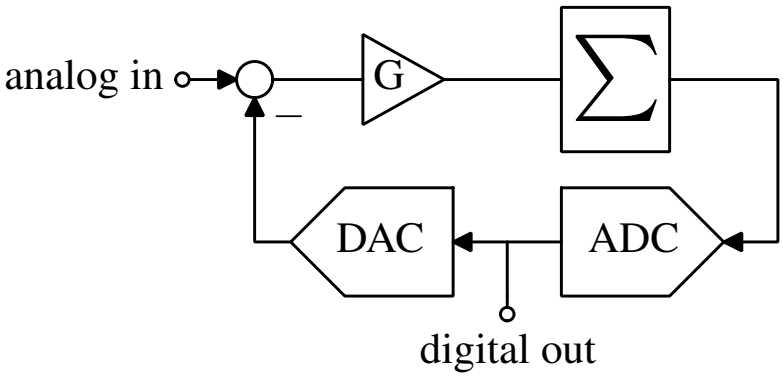


FIGURE 1.13. Block diagram of a 1-bit $\Delta\Sigma$ ADC with a feedfoward gain of G.

For DC signals ($z = 1$), the transfer function becomes:

$$out = \frac{G}{G} in - \frac{0}{G}\epsilon = in$$

which is independent of the feedfoward gain G. This will be relevant when choosing input signals for the proposed calibration algorithm.

1.3.3 Pipeline architecture

The pipeline architecture is composed by a number of cascaded stages. A block diagram of a stage is shown in figure 1.14. Each stage computes a low-resolution digital conversion of the input signal, which is quantized by the stage sub-ADC. That conversion is the digital output of the stage and it also feeds into a DAC that will convert it back to the analog domain. The difference between the DAC output and the input signal is the computed to obtain the quantization error. Then the quantization error of this stage is amplified and processed by the next stage.

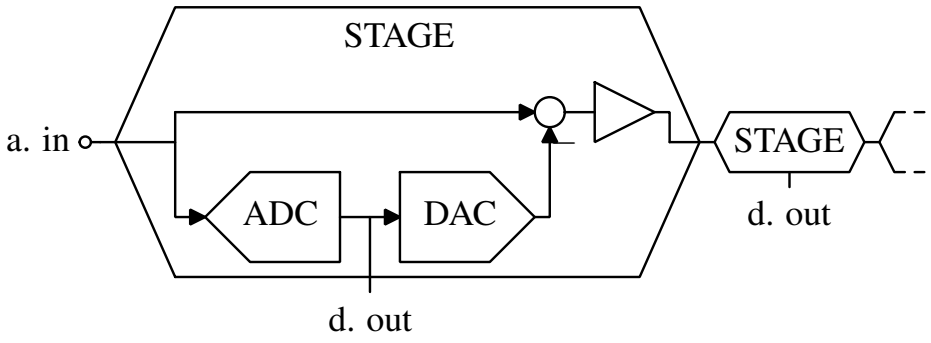


FIGURE 1.14. Block diagram of a Pipeline ADC, detail on one stage.

In switched-capacitor circuits an MDAC, which stands for Multiplying DAC, is commonly used. It is a circuit that computes the DAC output, the difference with the input signal, and the multiplication of the result in one cycle. A simple implementation of a 1-bit, *flip-around* MDAC

can be seen on figure 1.15.

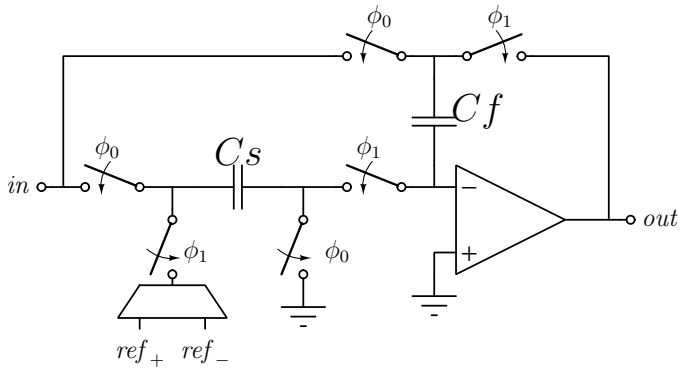


FIGURE 1.15. A 1-bit MDAC based on capacitive ratios.

The circuit operates in two phases: In ϕ_0, the feedback (Cf) and feedfoward (Cs) capacitors are charged with the input voltage. Then in ϕ_1, a reference voltage $ref_{+/-}$ is selected, usually based on the output of the stage sub-ADC, and is connected to Cs. Cf is connected closing the feedback path which, assuming an ideal amplifier, forces the feedfoward capacitor voltage difference to converge to the reference voltage. The charge difference associated with this voltage change is pumped into the feedback capacitor. Then the output voltage is:

$$out = in + \frac{Cs}{Cf}(in - ref_k), \tag{1.1}$$

With k either $+, -$, depending on the multiplexer state. To complete a pipeline stage, a fast ADC such as a flash ADC can be connected in with the MDAC and its output used to control the multiplexer. Notice that if Cs and Cf values are the same, the in signal has a gain factor of 2, which is the correct gain for the residual of a 1-bit stage.

For higher resolution stages, more Cs capacitors can be added in parallel, the number of voltages the multiplexer selects can be increased or both of these strategies can be used. In figure

1.16, a B.5 bit implementation is shown where each capacitor can select 3 references.

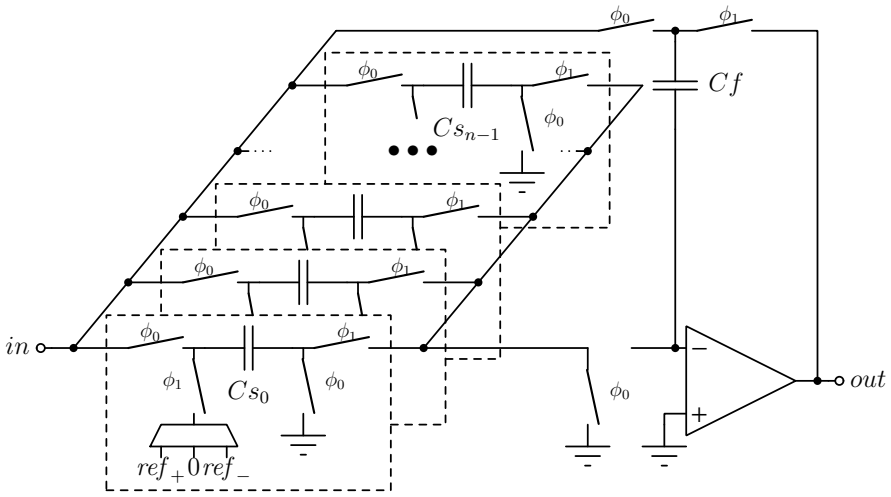


FIGURE 1.16. Switched-capacitor MDAC.

The residual voltage of this stage can be computed using charge conservation:

$$
\begin{aligned}
0 &= \Delta Qf + \sum_{k=0}^{n-1} \Delta Qs_k \\
0 &= Cf(Vin - Vout) + \sum_{k=0}^{n-1} Cs_k(Vin - Vref_k) \\
Vout &= Vin + \sum_{k=0}^{n-1} \frac{Cs_k}{Cf}(Vin - Vref_k) \\
Vout &= Vin \underbrace{\left(1 + \sum_{k=0}^{n-1} \frac{Cs_k}{Cf}\right)}_{Vin \text{ scaling}} - \underbrace{\sum_{k=0}^{n-1} \frac{Cs_k}{Cf} Vref_k}_{\text{DAC output}}
\end{aligned}
\tag{1.2}
$$

The ratios $\frac{Cs_k}{Cf}$ are chosen to be integers and that the sum of all ratios matches the desired residual gain for that stage. In this case $Cs_k = Cf$.

1.4 Error sources

Until now, the methods for characterizing errors have been presented. Now the origin of these errors will be examined. Since only static errors are going to be calibrated by the proposed algorithm, only these will be presented.

a) Finite gain

In the equations of the ADC architectures, all amplifiers were assumed to be ideal. If the transconductance amplifier has a finite gain, not all charge will be transferred. For a DC gain of $T0$ with an amplifier in a capacitive feedback network, the output voltage will be scaled as shown on (1.3) . This not only changes the output analog voltage for amplifying circuits. For integrators with an amplifier at its core this will translate to integrator leakage:

$$Vout_{finite} = \frac{T0}{1+T0} Vout_{ideal} \tag{1.3}$$

b) Capacitor mismatch

On-chip capacitors are be subject to mismatch errors because of process variations, which in turn will affect the capacitance ratios. (1.2) shows that if the capacitive ratios are no longer integer, the scaling of the input voltage will be non-integer, affecting the linearity of the DAC output.

c) DAC error

The DAC voltage can be affected by the capacitor mismatch and also by PVT errors in the unregulated ref voltages uses in the weighted sum.

d) sub-ADC error

Non-linearites in the stage sub-ADC can lead to the wrong output code and, in turn, to incorrect residual voltage calculation for a pipeline ADC.

1.5 Calibration techniques

For correcting the aforementioned errors, several calibration techniques have been developed. Some work by adding redundancy to the circuit whereas, others try to measure the errors to correct them in the analog domain or compensate for them later in the digital domain.

The proposed algorithm belongs to the last category but relies on another algorithm to prevent over-range in the internal components of the ADC. This calibration method will be presented and after it other calibration schemes will be explained.

1.5.1 Half-bit stages

In pipeline architectures, to compensate for possible sub-ADC errors, half-bit stages can be implemented. These add redundancy to compensate for possible offsets in the ADC step voltages. In figure 1.17 it can be compared the residual of a 1-bit stage with a 1.5-bit stage. Notice that the range of the 1.5-bit residual only touches the voltage limits on the extremes of the input FSR.

Let B be an integer. The relation between an B-bit DAC and it B.5 counterpart is $LSB_{half} = LSB_{int}/2$ for all bits between the extremes and $LSB_{half} = 3/4LSB_{int}$ for the extreme codes.

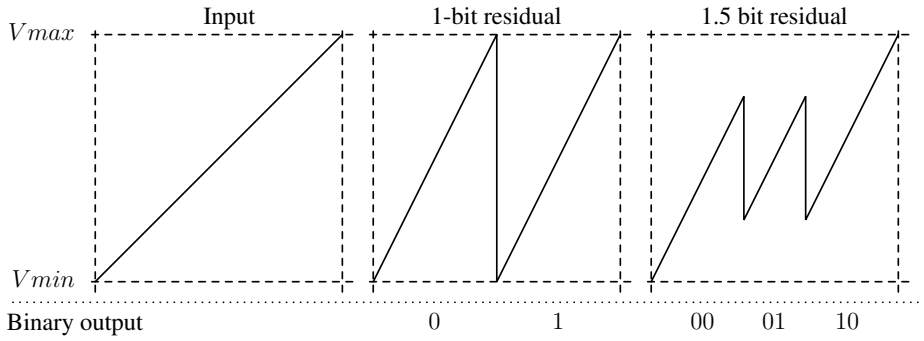


FIGURE 1.17. Residues for 1 and 1.5 bit stages.

Since a half-bit stage LSB is a half of its integer counterpart, the residual range is also reduced by a half. If an error occurs in the DAC conversion, an over-range or under-range will not happen unless the DAC is off by at least a $LSB_{half}/2$ as seen on figure 1.18.

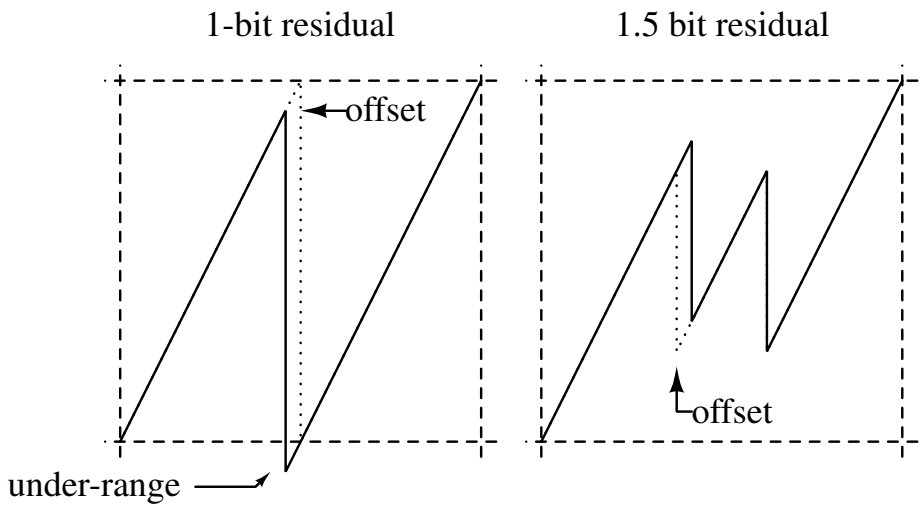


FIGURE 1.18. Residues for 1 and 1.5 bit stages with offset.

In ideal integer bit stages, the result is combined as follows:

$$code_{out} = \sum_{k=0}^{s-1} b_s \prod_{h=0}^{k-1} 2^{N_h}$$

Where N_s is the number of bits of the stage and b_s is the ADC value. For example an ADC with stages of 3, 1 and 2 bits, in that order, represented as $S_2S_2S_2$, S_1 and S_0S_0. The final conversion is computed as:

$$\begin{array}{rcccccc} & 0 & 0 & 0 & 0 & S_0 & S_0 & \\ & 0 & 0 & 0 & S_1 & 0 & 0 & // \prod_{h=0}^{1-1} 2^{N_h} = 4 \text{ (shift 2)} \\ + & S_2 & S_2 & S_2 & 0 & 0 & 0 & // \prod_{h=0}^{2-1} 2^{N_h} = 4 \cdot 2 = 8 \text{ (shift 3)} \\ \hline & S_2 & S_2 & S_2 & S_1 & S_0 & S_0 & \end{array}$$

For half-bit stages, since $LSB_{half-bit}$ is $LSB_{int-bit}/2$, the last bit is associated to the resolution of the most significant bit of the next stage. This can be interpreted as adding binary numbers with one decimal place. Then the combination is computed as follows:

$$code_{out} = \sum_{k=0}^{s-1} b_s \prod_{h=0}^{k-1} 2^{\lfloor N_h \rfloor}$$

where $\lfloor x \rfloor$ is the floor function, so b_s last bit is decimal if the stage is half-bit. For example an ADC with stages of 3.5, 1.5 and 2 bits, in that order, represented as $S_2S_2S_2.S_2$, $S_1.S_1$ and S_0S_0. The final conversion is computed as:

$$\begin{array}{rcccccc} & 0 & 0 & 0 & 0 & S_0 & S_0 & \\ & 0 & 0 & 0 & S_1 & .S_1 & 0 & // \prod_{h=0}^{1-1} 2^{\lfloor N_h \rfloor} = 4 \text{ (shift 2)} \\ + & S_2 & S_2 & S_2 & .S_2 & 0 & 0 & // \prod_{h=0}^{2-1} 2^{\lfloor N_h \rfloor} = 4 \cdot 2 = 8 \text{ (shift 3)} \\ \hline & S_2 & S_2 & X & Y & Z & S_0 & \end{array}$$

where X, Y and Z are a combination of bits either aligned in the sum or result of a carry. This way, assuming the MDAC is ideal, the ADC is protected from possible sub-DAC errors in the ADC conversion in each stage up to $LSB_{half-bit}/2$. This is because the residue will not overload (over-range or under-range) and the next stage will be able to manage the conversion. The first appearance of this technique can be found on (Lewis, Fetterman, Gross, Ramachandran, & Viswanathan, 1992).

The proposed algorithm does not require for this technique to be implemented in the chip to work, but it effectiveness will improve if this technique is also applied.

1.5.2 Variable components

The previous technique can compensate DAC errors, but will not be useful against MDAC errors, such as mismatch. If the capacitive ratios are not as designed, the feedback loop will have a gain error. The effect of this error can be visualized in figure 1.19, where the residue will not overload the output, but its value passed onto the next stage will be incorrect.

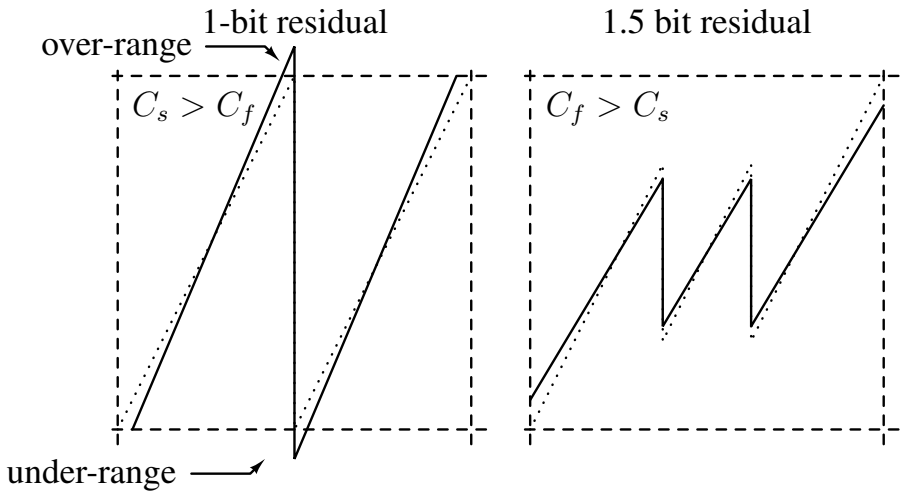


FIGURE 1.19. Residues for 1 and 1.5 bit stages with gain error.

In order to solve this problem it has been proposed to use variable capacitors as in Goes, Vital, and Franca (1996) and calibrate offline by measuring the residual of an artificial input created by a DAC.

1.5.3 Least Means Squares (LMS) filter

Another method for calibrating MDAC errors is to measure the ADC or an ADC stage output and estimate the conversion errors. For example, it has been proposed to attach a $\Delta\Sigma$ ADC to the first stage of a pipeline converter (Tzi-Hsiung Shu, Bang-Sup Song, & Bacrania, 1995) in order to estimate the error on the first stage as seen on figure 1.20. The error is then digitally subtracted from the digital output.

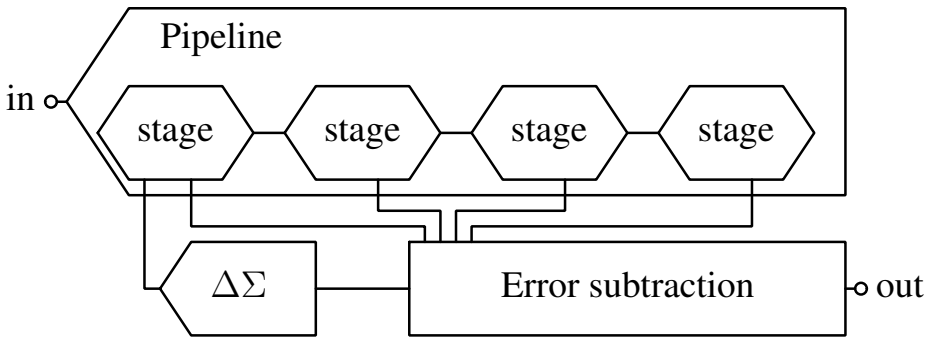


FIGURE 1.20. Simplified diagram of parameter estimation example.

A popular method of parameter estimation is by using least mean squared (LMS) filters. In its first inception in (Abou-El-Kheir, Abbas, & Khedr, 2014), the output of a slow but precise ADC is compared with the output of a fast ADC every number of cycles in order to estimate the error. Then the LMS filter estimates the parameters of the model such that the error is minimized.

For more information on analog-to-digital converters theory and design please refer to Pelgrom (2016).

1.5.4 The proposed calibration algorithm

The proposed calibration algorithm will estimate the ADC electrical parameters based on offline measurements of internally-generated test signals. Then, during normal conversion, these parameters can be used to combine the digital output of every stage taking into account the static errors of each one.

To perform the measurements in order to obtain the parameters, some reconfiguration of the ADC electrical circuitry is required. The next chapter will discuss the necessary changes needed to perform such reconfiguration.

2. ALGORITHM CIRCUITAL REQUIREMENTS

This chapter will explore the system and circuital modifications needed to perform the calibration algorithm. In order for these modifications to be effective, two requirements are needed: the ADC must be a pipeline converter, and it must be implemented with switched capacitors. These two constrain the applicability of the algorithm but they are necessary since the first will enable the instanciating of a different architecture between pipeline stages, and the second will enable the use of amplifiers as integrators.

2.1 Calibration algorithm general scheme

The proposed algorithm runs offline. It will stop the conversion operation during calibration or part of the calibration process. The calibration estimates electrical parameters of the converter, which is done in two steps. First, the normal conversion stops and calibration data is generated and stored to be used in the second step. Second, the converter may resume operations while a set of equations using the previous data is solved in background. The result of this step is an estimation of the electrical parameters of the converter. After an estimation of the converter electrical parameter is obtained, the estimated parameters are used to digitally calibrate the output of the ADC.

The data obtained in the first step is the output stream of codes of $\Delta\Sigma$ converters instanciated in the boundary between pipeline stages. The input of these converters are generated by the stages themselves and the exact voltage values do not need to be known in advance. To understand how it is possible to make a pipeline converter behave in such a way, the following statements will be analyzed:

- It is possible to instanciate $\Delta\Sigma$ ADCs in the interface between pipeline stages.

- The instanciated $\Delta\Sigma$ converter in a stage boundary is not unique, but it is a multiplicity of combinations of elements that give a number of different converters.
- These converters can generate their own input and set their initial condition.

The rest of this chapter will explain how to implement each of the aforementioned statements.

2.2 Instancing $\Delta\Sigma$ converters in the pipeline stages boundaries

2.2.1 System-level modifications

To instanciate one ADC with the elements of another we will first check if the building blocks required to make the pipeline stage contain the building blocks required to make a $\Delta\Sigma$ converter.

To build a pipeline stage, using as reference figure 1.14, the following circuital blocks are required: an ADC (the stage sub-ADC), a DAC (part of the MDAC), a voltage subtractor and an amplifier (part of the MDAC). To build a $\Delta\Sigma$ converter, using as reference figure 1.10, it is required to have an ADC, a DAC, a voltage subtractor and an integrator. The only different building block between these two architectures is an amplifier in the $\Delta\Sigma$ ADC that should be an integrator.

In SC circuits, this integrator can be implemented using the same amplifier without any circuital modifications; if the charge is held in the feedback capacitors instead of being flushed with every clock cycle, then the amplifier becomes an integrator. With this in mind it is possible to treat the amplifier as an integrator when comparing the building blocks of both systems and notice that both architectures share the same building blocks.

To instantiate a $\Delta\Sigma$ modulator between two pipeline stages, referred to as head for the predecessor and tail for the successor, the following system-level modifications are needed: in both stages, the sub-ADC is disconnected from the sub-DAC. Then the sub-ADC of tail is connected to control the sub-DAC of head. If the bit count of both stages differs, logic has to be added to

make the sub-ADC output and the sub-DAC input compatible. If the tail number of bits is less than head number of bits, the matching logic has to pad with 1's and 0's the extremes of the sub-ADC thermometer output until the number of bits matches the input bits of the sub-DAC. In the inverse case, the sub-ADC output extreme bits can be discarded. Figure 2.1 illustrates the previous reconfiguration. The logic block that handles the communication between the stage sub-ADC, the stage DAC and the next stage sub-ADC will be referred to as bit matching logic (BML).

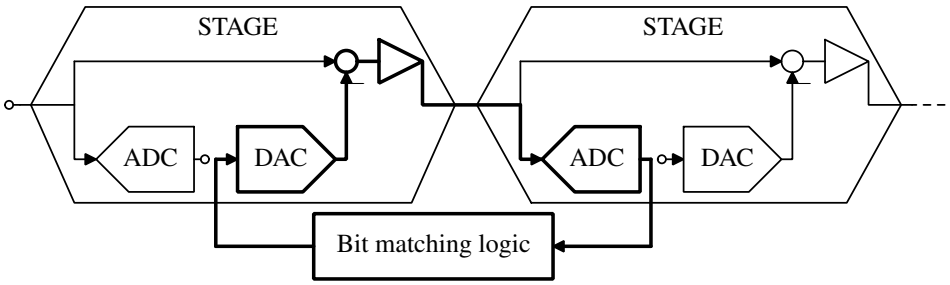


FIGURE 2.1. Instanciating of a $\Delta\Sigma$ loop in the interface of two pipeline stages. Thick line show the $\Delta\Sigma$ ADC.

2.2.2 Circuital modifications

The previously described ADC reconfiguration require circuital modifications. The conventional implementation of a SC, *flip-around* pipeline stage is depicted in figure 2.2. It consists of $n-1$ feedfoward capacitors (Cs) connected in parallel, and a single feedback capacitor (Cf). The total number of capacitors (n) matches the desired stage gain, which is $n = 2^B$ for B or $B.5$ bits when each Cs capacitor is connected to 2 or 3 reference voltages, respectively.

The circuit is operated by a two-phase clock, with non-overlapping phases ϕ_0 and ϕ_1. During ϕ_0, switches sw_{cm}, sw_0 and sw_3 are closed, charging all Cs and Cf capacitors with the input voltage. When ϕ_1 is high, switches sw_1 and sw_2 close and the stage's sub-ADC selects the appropriate reference voltages for every Cs capacitor. Assuming perfect charge redistribution, the

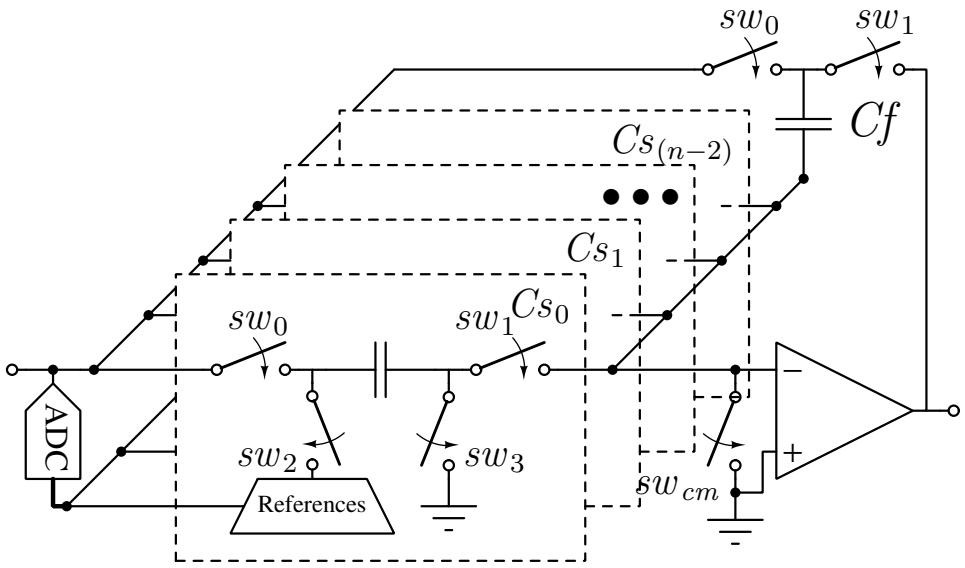


FIGURE 2.2. Conventional pipeline stage implementation.

residual voltage is (2.1):

$$Vout = Vin \cdot \underbrace{\left(1 + \sum_{s=0}^{n-2} \frac{Cs_s}{Cf}\right)}_{\text{Stage gain}} - \sum_{s=0}^{n-2} \frac{Cs_s}{Cf} Vref_s \tag{2.1}$$

where $Vref_s$ is the selected reference voltage by the analog multiplexer of each Cs_s capacitor. This voltage will be referred to as the selected voltage.

The modifications that enable the use of the calibration algorithm are depicted in figure 2.3. The feedback capacitor is eliminated and the $n-1$ feedforward capacitors are replaced by n modified capacitors (C). These capacitors have the same architecture as the feedfoward capacitors but have one additional switch (sw_4) which enables one or many capacitors to be connected in feedback configuration. Also extra logic, the BML, has to be added between the stage sub-ADC and the MDAC. This block receives data from both the stage sub-ADC and the next stage

sub-ADC and a digital register, to convert arbitrary input digital signals with the MDAC.

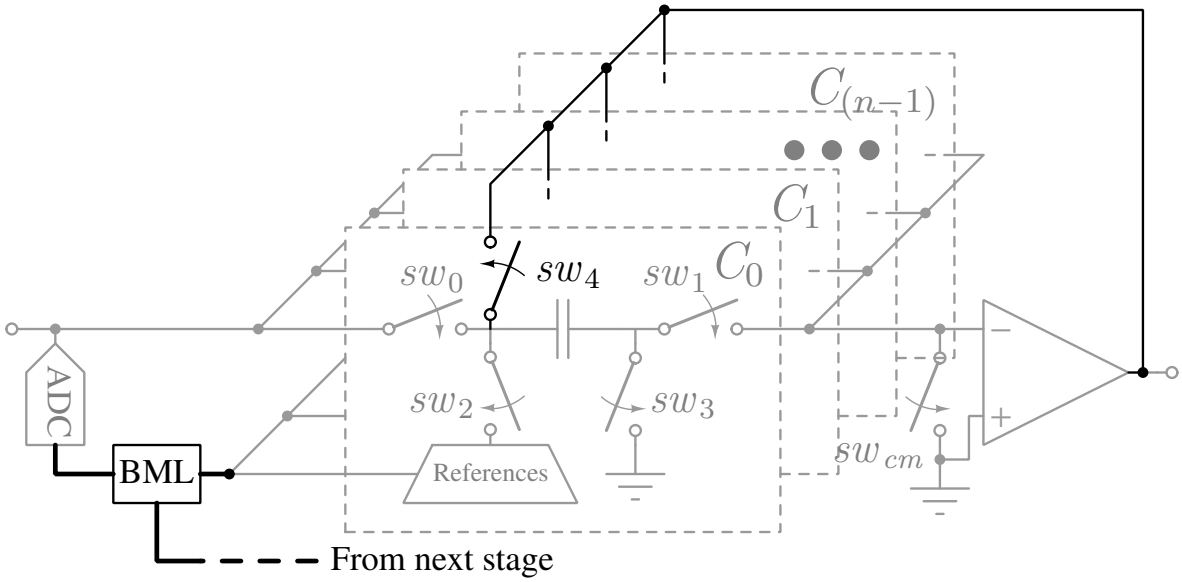


FIGURE 2.3. Modified pipeline stage implementation, the modified parts are highlighted in the diagram.

2.3 $\Delta\Sigma$ number of configurations

Several $\Delta\Sigma$ converters can be instantiated at the interface between two pipeline stages. This section will address the number of valid combinations that can be created.

2.3.1 Naive combinations

The modified capacitors, C can be used as either feedback or feedforward capacitors. As long as there is at least one feedback capacitor and one feedforward capacitor in the MDAC, a mathematically valid $\Delta\Sigma$ converter can be created. For a particular capacitor configuration, set S will contain the indexes of the capacitors selected to operate in feedfoward. Set F will contain the complement of set S and will have the indexes of the capacitors that will be used as feedback,

such that the sum of the cardinalities of both sets are equal to the number of capacitors, this is, $n = |S| + |F|$. These sets are represented in figure 2.4.

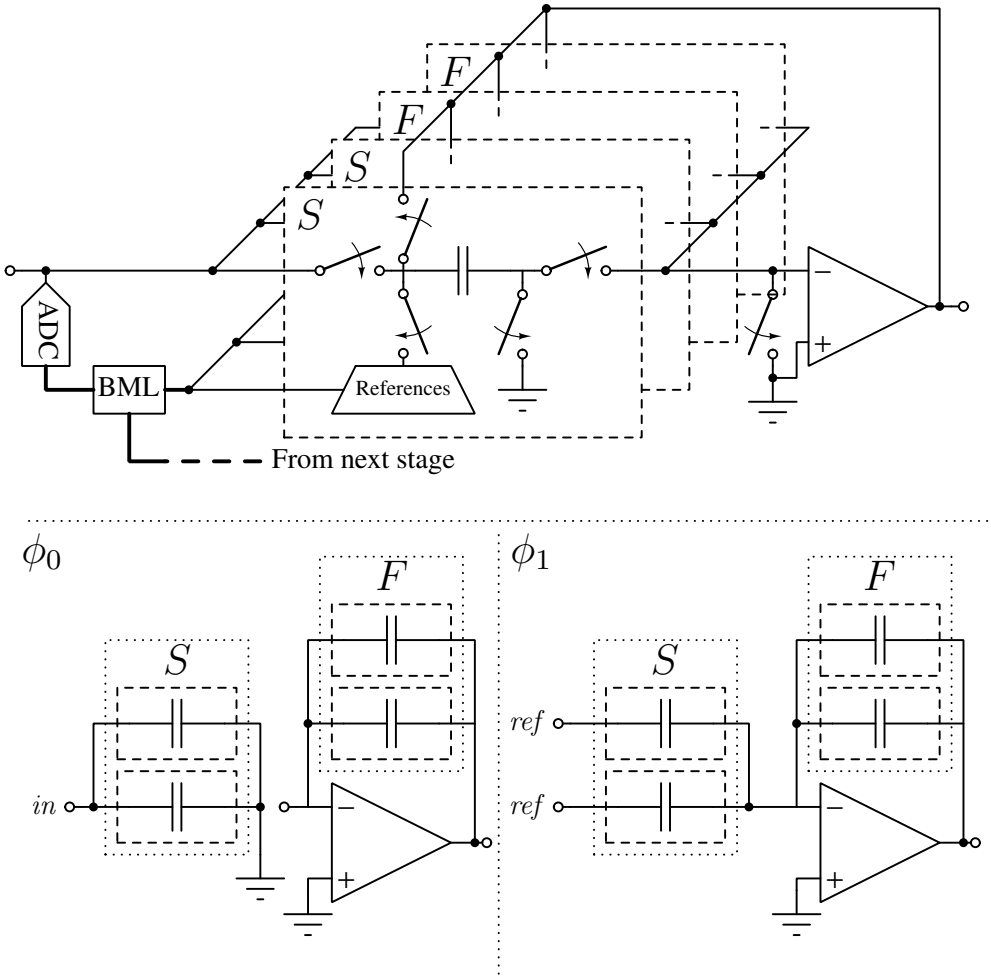


FIGURE 2.4. The capacitors of the stage can be separated into feedfoward capacitors (set S) and feedback capacitors (set F).

The number of unique configurations is given by summing all possible capacitor permutations for a specific S cardinality, and summing for all possible S cardinalities, as expressed in (2.2).

$$N_{\Delta\Sigma} = \sum_{|S|=1}^{|S|=n-1} \binom{n}{|S|} \tag{2.2}$$

2.3.2 Feedfoward gain limit

The previous result is valid only if the converter internal nodes could handle any voltage without over-ranging or going over the acceptable operating voltages. Real ADCs cannot work in those conditions and usually high non-linearities arise if some internal voltage exceeds the operating range. In this case, the signal that could overload $\Delta\Sigma$ loop is the output of the integrator. Assuming an ideal converter, the change in voltage for a single conversion step of the loop is in (2.3).

$$\Delta V = \sum_{s\in S} \frac{C_s}{\sum_{f\in F} C_f} \left(V_{in} - V_{ref}\right) \tag{2.3}$$

To prevent the integrator residual to go over the designed range, the feedfoward gain $\sum_{s\in S} \frac{C_s}{\sum_{f\in F} C_f}$ must be at most 1. For an ADC where all capacitors have the same value, as it is the case for the example MDAC implementation of 2 or 3 voltage references for each capacitor, this means the cardinality of S must be at most the cardinality of F. When capacitor mismatch is present, in order to make sure the feedfoward gain is less than or equal to 1, the cardinality of S must be less than the cardinality of F. This limits the number of valid combinations to (2.4).

$$N_{\Delta\Sigma} = \sum_{|S|=1}^{|S|=\lfloor (n-1)/2 \rfloor} \binom{n}{|S|} \tag{2.4}$$

An example of this restriction is shown in figure 2.5, where residuals of the same input signal are tested for different cardinalities of set S in a 3.5-bit pipeline stage. The envelope of the residual increases with the number of elements of S and the envelope will get past the FSR when $|S| > \lfloor n/2 \rfloor$.

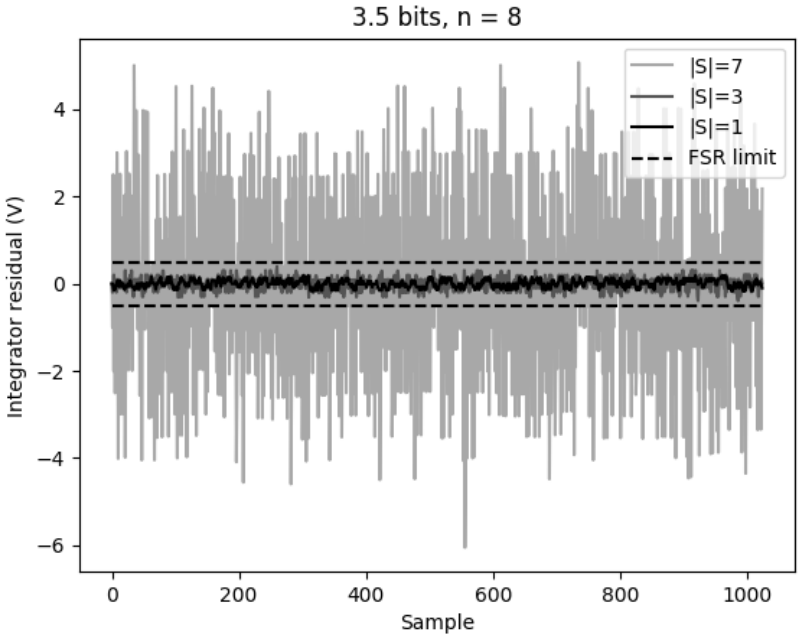


FIGURE 2.5. Integrator residual for $\Delta\Sigma$ converters built from a of 3.5 bits pipeline stage.

2.3.3 Bit matching

Until now it has been mentioned that the BML will condition the codes from the tail stage to be used in the head stage when operating as a $\Delta\Sigma$ loop, but it has not been explained how this match will be performed. For a conventional pipeline ADC, the codes of the stage sub-ADC are mapped to the MDAC using a table which is hardcoded into the stage circuitry. This table starts with the minimum code selecting the minimum references of all feedfoward capacitors (C_s) and escalate to the maximum code selecting the maximum references of all (C_s). The escalation is done by changing the reference of only one capacitor to its successor with each code increase. In table 2.1, a 3.5-bit mapping is presented as an example. The units of the mapping are LSBs and the Total row shows the output of the DAC associated with the MDAC. It is linear with the input codes.

For $\Delta\Sigma$ operation, the signals that have to be matched are the tail sub-ADC with the head sub-DAC after being modified by the $\Sigma\Delta$ configuration. This means that the number of control

TABLE 2.1. Code to references mapping for a 3.5 bit pipeline converter. The header are the digital codes of the ADC output and the rows units are in the DAC LSB.

DAC \ ADC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Cs_0	-1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
Cs_1	-1	-1	0	0	0	0	0	0	0	1	1	1	1	1	1
Cs_2	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1	1
Cs_3	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1
Cs_4	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1
Cs_5	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1
Cs_6	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1
Total	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

signals the head sub-DAC recieves is not the same as during normal conversion, because they are dependent on the cardinality of the S set.

The BML has to be able to map the sub-ADC codes in three situations: when the sub-DAC resolution is less than that of the sub-ADC, when both have the same resolution and when the sub-DAC resolution is greater than that of the sub-ADC.

When the resolutions of the sub-ADC and sub-DAC match, the table for a conventional pipeline stage mapping can be used. When the sub-DAC has a greater resolution that the sub-ADC, the conventional pipeline mapping used for the sub-ADC can be padded with the extreme values to fill all the code space of the sub-DAC. In table 2.2, an example of this padding is shown to expand a 2.5-bit sub-ADC output to match a 3.5-bit sub-DAC input.

This padding will introduce a non-linearity in the analog-digital-analog conversion. With the increase of the number of padded codes, the transfer function of the system will approximate a step function, as seen on figure 2.6. This non linearity is equivalent to adding a feedback gain to the system and clipping the output. It would generate conversion errors if the $\Delta\Sigma$ stream of codes is then decimated, but as it will later explained, the complete digital stream from the conversion

TABLE 2.2. Padding of a 2.5 bit ADC map to fit a 3.5 bit DAC.

DAC \ ADC	0	1	2	3	4	5	6
Cs_0	-1	0	0	0	1	1	1
Cs_1	-1	-1	0	0	0	1	1
Cs_2	-1	-1	-1	0	0	0	1
Total	-3	-2	-1	0	1	2	3

→

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1	1	1
-1	-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1	1
-1	-1	-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1
-3	-3	-3	-3	-3	-2	-1	0	1	2	3	3	3	3	3

is used and these distortions do not cause any problems in the calibration process.

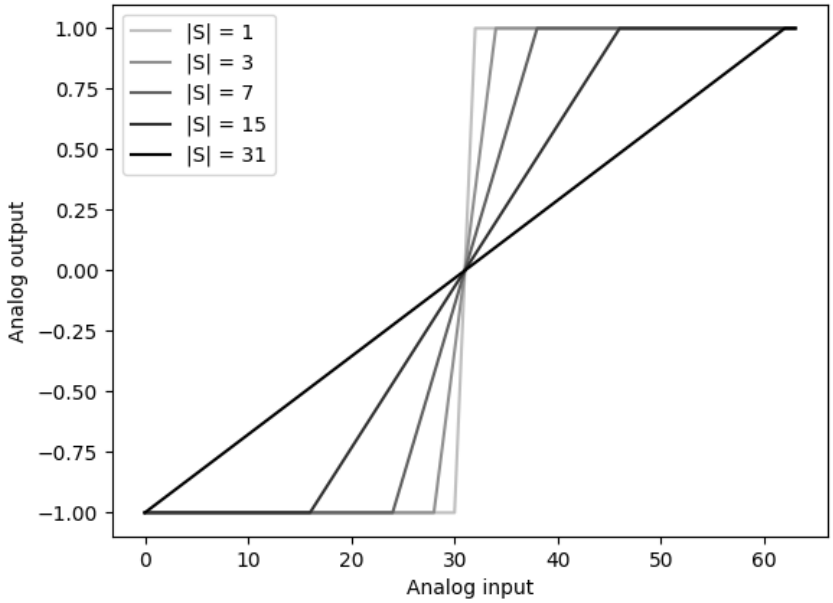


FIGURE 2.6. Different transfer functions for analog-digital-analog conversion for a 6.5 bit stage emulating $\Delta\Sigma$ loops of different resolutions.

When the sub-DAC has a lower resolution that the sub-ADC, the conventional pipeline mapping used for the sub-ADC can be chopped to fit the sub-DAC code space. In order to prevent problems arising from strong non-linearities, the remaining codes must be evently spaced from each other. This can be achieved by stripping all even codes from the code mapping, thus halving the resolution of the code map. An example of chopping from 3.5 bits to 2.5 bits is shown in table

2.3. This process can be applied recursively up to 1 or 1.5 bits. If after chopping the relation is inverted, that is, now the sub-DAC has a greater resolution that the sub-ADC, then the previous padding process can be applied.

TABLE 2.3. Chopping of a 3.5 bit ADC map to fit a 2.5 bit DAC.

DAC \ ADC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Cs_0	-1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
Cs_1	-1	-1	0	0	0	0	0	0	0	1	1	1	1	1	1
Cs_2	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1	1
Cs_3	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1
Cs_4	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1
Cs_5	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1
Cs_6	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	1
Total	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

$\rightarrow$

0	1	2	3	4	5	6
0	0	0	0	1	1	1
-1	0	0	0	1	1	1
-1	0	0	0	0	1	1
-1	-1	0	0	0	1	1
-1	-1	0	0	0	0	1
-1	-1	-1	0	0	0	1
-1	-1	-1	0	0	0	0
-6	-4	-2	0	2	4	6

The previous methods were exemplified with half-bit stages, but these methods can also be applied to integer resolution stages. An offset will be added in the conversion from analog to digital and then back to analog, but it will not affect the $\Delta\Sigma$ operation.

2.4 Artificial input generation and $\Delta\Sigma$ initial conditions

For any mode of operation, the MDAC is able to ignore the input node and auto generate its own input. The only necessary modification for this is to close sw_2 instead of sw_0 during ϕ_0. This will make the stage read the capacitor references instead of the input node. Then any signal that can be generated with the stage sub-DAC can be fed to the stage.

At a system-level, this process can be conceptualized as cloning the sub-DAC of the stage and connecting it to the stage input. Then this DAC is controlled by the BML with a digital signal, which is converted to analog by the DAC and then converted again to digital with the ADC, as depicted in figure 2.7.

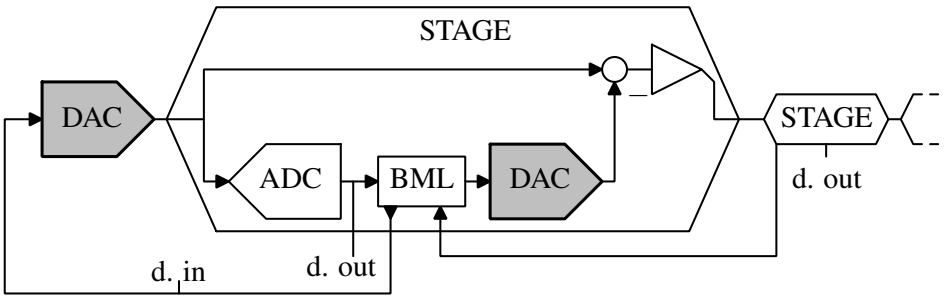


FIGURE 2.7. Block diagram of equivalent system for input self generation. The highlighted elements are identical.

This same feature can be used to pre-charge the $\Delta\Sigma$ loop. Normally the capacitors assigned to set F would be flushed of any charge and then fixed to the feedback position. To have the ability of pre-charging the loop, on the first ϕ_0 period, before fixing the capacitors into feedback position, the capacitors references can be connected closing sw_2. Then, when the capacitors flip to be fixed into the feedback position, the charged reference voltage in each capacitor will be averaged with the capacitor values as weights. This voltage is the $\Delta\Sigma$ initial condition.

This capacity enables the ADC to do not depend on any external voltage reference, since the tests signals are generates from within using the same components that are used for normal conversion operation.

2.5 Stage modes of operation

In this section, a review of the operating modes of the ADC is presented before merging the previously explained facts into an algorithm capable of performing calibration.

2.5.1 Pipeline mode

During normal conversion the stages are configured in pipeline mode. In this mode one of the C capacitors is selected to function as flip-around feedback capacitor. This capacitor is referred

to as C_f. The feedfoward capacitors will be named $C_{s,k}$ for $k \in [0..n-2]$. In ϕ_0, all capacitors sample the input signal by closing sw_0 and sw_3. In ϕ_1 capacitor C_f flips to feedback position by closing sw_4 and sw_1, and each $C_{s,k}$ capacitor is connected to its own reference by closing sw_2 and sw_1. This process is repeated for each clock cycle. A diagram of the switching performed on each period is shown in figure 2.8.

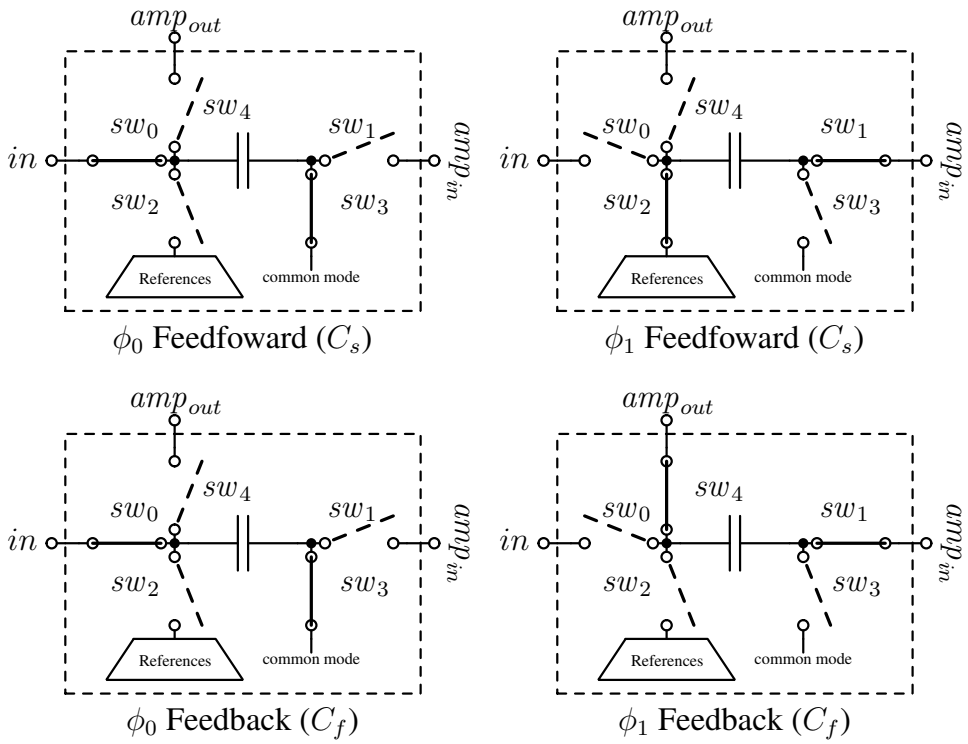


FIGURE 2.8. Switches states during pipeline mode.

2.5.2 $\Delta\Sigma$ mode

During the data capture stage of the calibration process, the $\Delta\Sigma$ mode is enabled. For a specific set S and F, the switching procedure is as follows:

For capacitors in set F on the first ϕ_0 cycle, switches sw_2 and sw_3 are closed in order to charge the initial condition of the $\Delta\Sigma$ loop. The reference selection is performed by a register of the BML. Then, on the first ϕ_1 and for the rest of that $\Delta\Sigma$ conversion, switches sw_4 and sw_1 are closed to fix those capacitors in feedback position. Figure 2.9 shows a diagram of this switching.

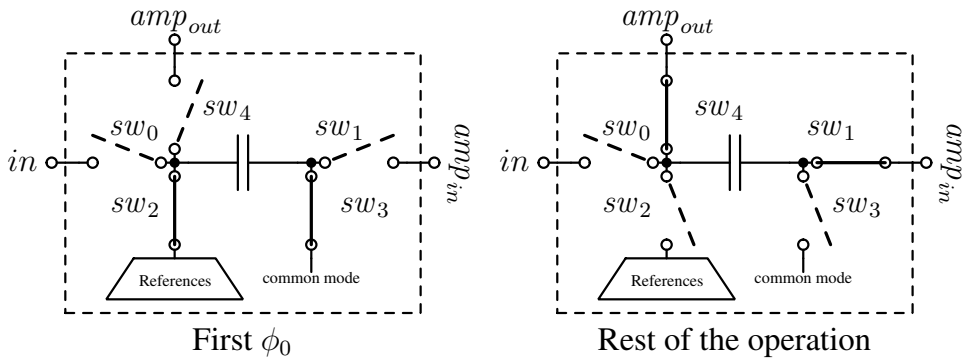


FIGURE 2.9. Switches states during $\Delta\Sigma$ mode for switches in capacitors of set F.

For capacitors in set S, on every ϕ_0 cycle, switches sw_2 and sw_3 are closed in order to generate an artificial input. The reference selection is performed by a register of the BML which value changes on every clock cycle.. On every ϕ_1, switches sw_2 and sw_1 are closed to connect the capacitors on feedfoward position, and the reference selection is performed by the BML based on the output code of the next stage sub-ADC and following the bit matching explained in section 2.3.3. Figure 2.10 shows a diagram of this switching.

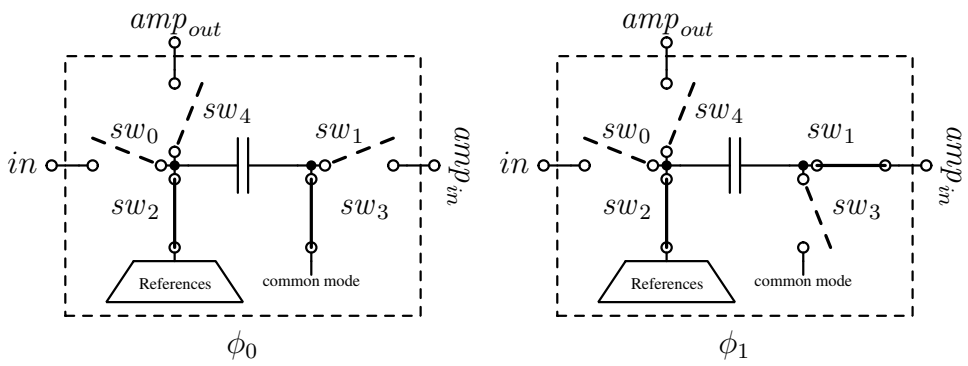


FIGURE 2.10. Switches states during $\Delta\Sigma$ mode for switches in capacitors of set S.

3. PROPOSED ALGORITHM

This chapter will explain how the previous circuit modifications enable the calibration of the pipeline converter.

3.1 $\Delta\Sigma$ model

This algorithm requires the $\Delta\Sigma$ loop to be modeled by a mathematical function. The closer this model is of the actual ADC, the closer the estimation parameters will be to the actual electrical parameters.

The election of the $\Delta\Sigma$ model is up to the designer of the ADC. For this work an ideal $\Delta\Sigma$ converter with linear charge loss will be used for modeling the ADC. The model equation is:

$$V_{out}[k] = \eta \left(V_{out}[k-1] + \sum_{s \in S} \frac{C_s}{\sum_{f \in F} C_f} \left(V_{in_s}[k] - V_{ref_s}[k] \right) \right) \tag{3.1}$$

where $V_{out}[k]$ will be referred to as the integral residual output, $V_{in_s}[k]$ is the input voltage for each capacitor. This voltage is self-generated as explained in section 2.4. $V_{ref_s}[k]$ is the reference voltage selected during $\Delta\Sigma$ conversion by the output of the next stage sub-ADC and mapped using the BML as explained in section 2.3.3. $\eta \in [0..1]$ is the charge loss factor, a value of 1 means no charge loss, as an ideal ADC.

3.2 Cost function

With the previous model, a cost function can be defined to measure the mismatch between the integral residual generated by a set of estimated parameters and the measurements obtained in the first step of the calibration process. This cost function can be conceptualized using figure 3.1. It pictures two ADCs being compared, one is the actual ADC and the other is a simulated

one using the mathematical model described in the previous section, it will be referred to as the estimated ADC. The estimated ADC will generate different outputs for the same conditions when the estimated parameters fed into the model change.

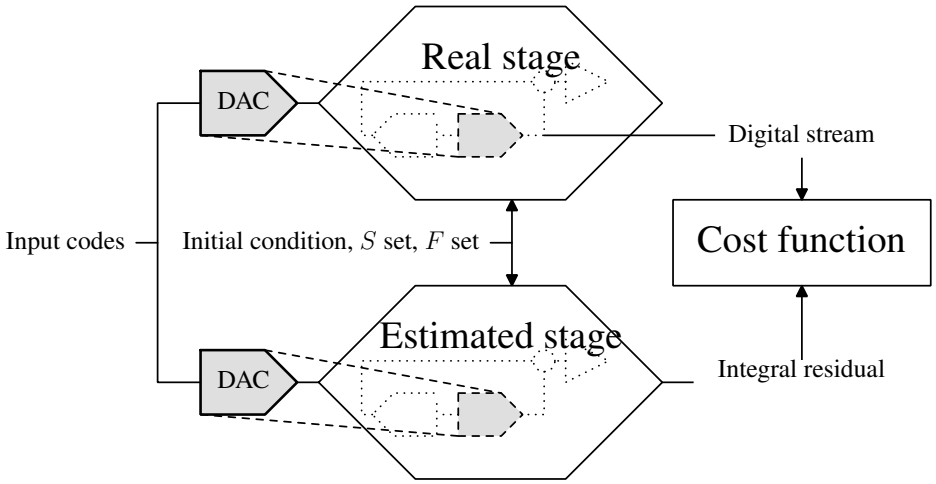


FIGURE 3.1. Calibration conceptualization, comparing the output of a actual ADC with the residual of an estimated ADC. The actual ADC "calibrates" the parameters of the estimated ADC to match the actual ADC.

Both ADCs inputs are connected to a clone of their respective sub-DACs, that are in turn controlled by the same digital input signal, have the same capacitor configuration and are initialized with the same initial conditions. The digital stream of codes from the actual ADC is stored and the integral residual from the estimated ADC is compared with the stream of codes using the cost function to be defined. The closer the estimated parameters are from the actual parameters, the lower the cost function should be.

When computing the error it is important to notice that the stream of codes generated by a $\Delta\Sigma$ loop is the quantized version of the integral residual. If the ADC is assumed to be ideal, then this means that for every sample of the stream of codes, the integral residual is contained in the code-bin. This enables to define a naive cost function which takes the distance of the integral

residual from the code-bin range for each sample and sums the squared value across all samples:

$$EF_n(x,c) = \sum_{k=0}^{N-1} E_n(bin_{min}(c[k]), bin_{max}(c[k]), x[k])^2 \tag{3.2}$$

with

$$E_n(a,b,x) = \begin{cases} 0 & \text{if } a < x < b \\ min\left(|a-x|, |b-x|\right) & \end{cases} \tag{3.3}$$

where x is the integral residual, c is the code stream and bin_{min}, bin_{max} are the functions that obtain the minimum and maximum transition voltages at the extremes of the code-bin associated with a code.

The cost function for a single sample can be zero if it lies in the code bin, which is a range of values. This means that there is a number of parameter sets that have a cost function equal to zero and no further parameter refinement can be performed. These parameter sets satisfy that all samples of the integral residual associated with them lie inside the code bin associated with each sample. The zone comprised by the plot of the union of all the code bins associated with the codes generated by the actual ADC will be referred to as the deadband, because there is no way to compare parameter sets were the integral residuals generated lie inside this band. A plot of a deadband example is pictured in image 3.2.

This naive cost function does not take into account PVT errors in the sub-ADC thresholds of the $\Delta\Sigma$ converter. These errors add uncertainty to the locations of the transition voltages for each code-bin. In order to take into account this uncertainty, the range of each code-bin can be expanded by a tolerance parameter (tol). This parameter can be selected such that the code-bin of the actual converter will be contained in the expanded code-bin with an arbitrary confidence, at the cost of a larger deadband. The extent of the loss of calibration precision because of a larger

deadband vs the loss of calibration precision because non-overlap of the actual code-bin with the deadband will be measured in the next chapter.

$$EF(x,c) = \sum_{k=0}^{N-1} E(bin_{min}(c[k]), bin_{max}(c[k]), x[k])^2 \tag{3.4}$$

$$E(a,b,x) = \begin{cases} 0 & \text{if } a - tol < x < b + tol \\ min\left(|a - tol - x|, |b + tol - x|\right) \end{cases} \tag{3.5}$$

This cost function process is depicted in figure 3.2, the samples where $E > 0$ are shown with a gray bar.

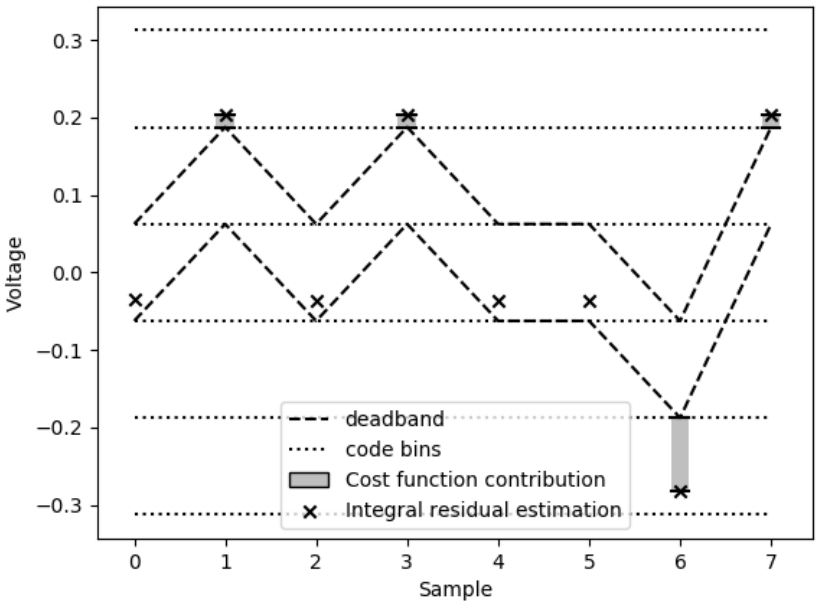


FIGURE 3.2. Cost function computation process, the samples that have a gray bar add to the value of E.

3.3 Minimization problem

The previous cost function is defined for a single $\Delta\Sigma$ configuration. A configuration is a specific combination of an S set, an initial condition and an input signal. A single configuration alone is not enough to calibrate the converter. This is because if we plot the cost function for a slice of the estimated parameters, as depicted in figure 3.3 , the cut being performed for an MDAC reference voltage and a capacitor, it is possible to see that the cost function has a minimum zone instead of a point, so multiple parameters minimize the function. This minimum zone corresponds to all the parameter sets where the generated integral residual is contained inside the deadband.

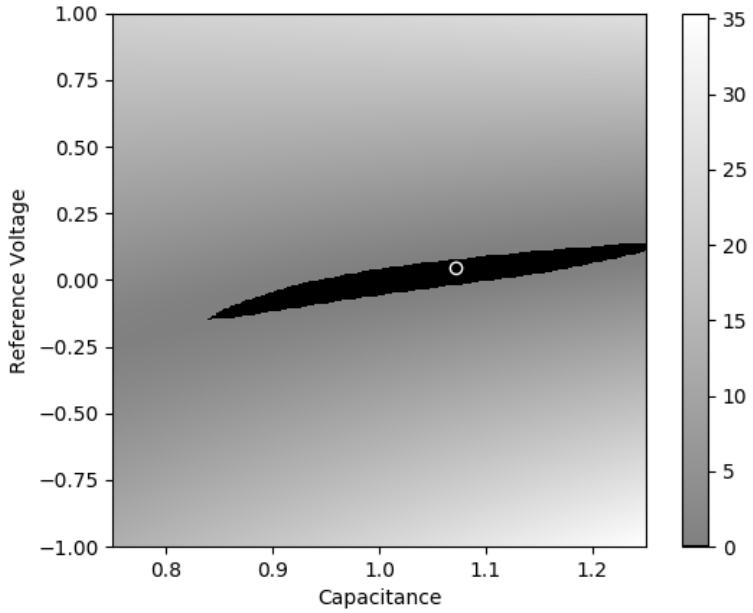


FIGURE 3.3. Slice of the cost function for a single configuration for a specific MDAC reference voltage and a capacitor. The actual parameters are marked with a circle.

A slice of the same two parameters, for another configuration with different set S but same input signal and initial condition, presented in figure 3.4, shows that the minimum zone has a different shape, but the actual parameter set is in both minimum zones, since the actual parameters must be a solution of the cost function.

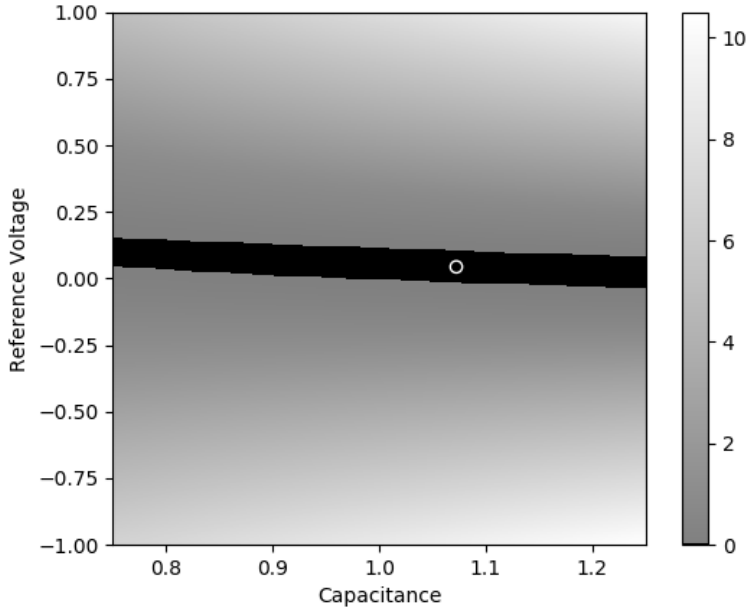


FIGURE 3.4. Slice of the cost function for another single configuration for the same MDAC reference voltage and same capacitor. The actual parameters are marked with a circle.

By adding these two functions, the new minimum zone will be intersection of the minimum zones of the cost functions. By intersecting the minimum zones of multiple combinations of configurations, initial conditions and input signals, the resulting minimum zone should shrink around the actual parameters of the ADC, as seen on figure 3.5, yielding better estimated parameters after solving the system of equations.

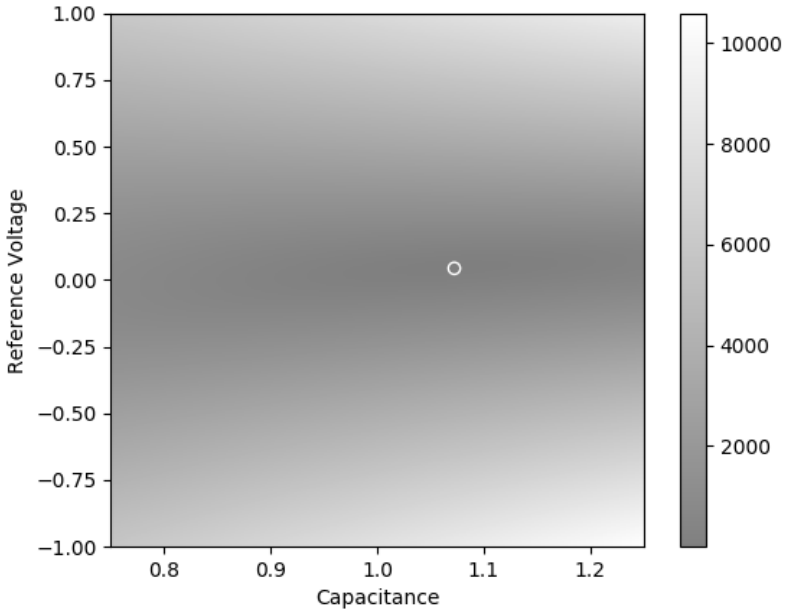


FIGURE 3.5. Slice of the sum of the cost functions of many configurations for the same MDAC reference voltage and same capacitor. The actual parameters are marked with a circle.

It is important to use different initial conditions during data collection. By inspecting the $\Delta\Sigma$ model presented in 3.1 it can be seen that the equations are always expressed in terms of voltage differences. This implies that any calibration result will have calibrated the voltage differences and not single-ended values. Having initial conditions different that zero adds absolute terms to the calibration equations, since the output voltage for an initial condition, using the same assumptions as for the $\Delta\Sigma$ model is:

$$V_{out} = \frac{1}{\sum_{f \in F} C_f} \sum_{f \in F} C_f V_{ic_f} \tag{3.6}$$

where V_{ic_f} is the selected reference voltage used to charge the initial condition. Another caveat is the selection of the input signal. This signal cannot be a DC signal since as shown on section

1.3.2, the $\Delta\Sigma$ loop output is insensitive to feedforward gain for DC inputs.

The final equation to minimize has to take into account the cost functions of the configuration, inputs and initial condition combination tested during the first stage of the calibration process. For this work the combination was performed by adding all the cost functions together. Depending on the minimization scheme used, a vector of all the cost function values can be passed to the minimization solver. The minimization algorithm used for this work is least squares, but many minimization algorithms can be applied to the system of equations.

3.3.1 Sub-ADC thresholds

The inclusion of the sub-ADC thresholds in the estimated parameters would eliminate the need of modeling the threshold variance and computing the tolerance factor, since the deadband edges would adjust to the transition voltages, if the calibration estimates correctly the sub-ADC threshold voltages. But when the thresholds are included in the estimation parameters, multiple solutions are added to the cost function. These solutions, unlike the multiple of solutions related to the deadband, can exist in an arbitrary distance from the actual parameters.

An example of these solutions is adding a constant, K, to the sub-ADC threshold voltages and the MDAC reference voltages of an existing solution. The addition of the constant to the threshold voltages will offset all the deadbands by the value of K. At the same time, the addition of K to the reference voltages will offset the $\Delta\Sigma$ integral residual by the same amount. Then when computing the cost function the relative position of the deadband with the $\Delta\Sigma$ integral residual will be the same and the cost function value will be preserved. Since the addition of the constant can be done to an already existing solution, the new parameters can be shifted by an arbitrary amount and still be a solution.

Figure 3.6 exemplifies this property. The cost function plot axis are a sweep of two constants. The x axis constant is the offset added to all sub-ADC threshold voltages and the y axis constant

is the offset added to al the MDAC references voltages. The diagonals $x = y$ of the plot have the same cost function value.

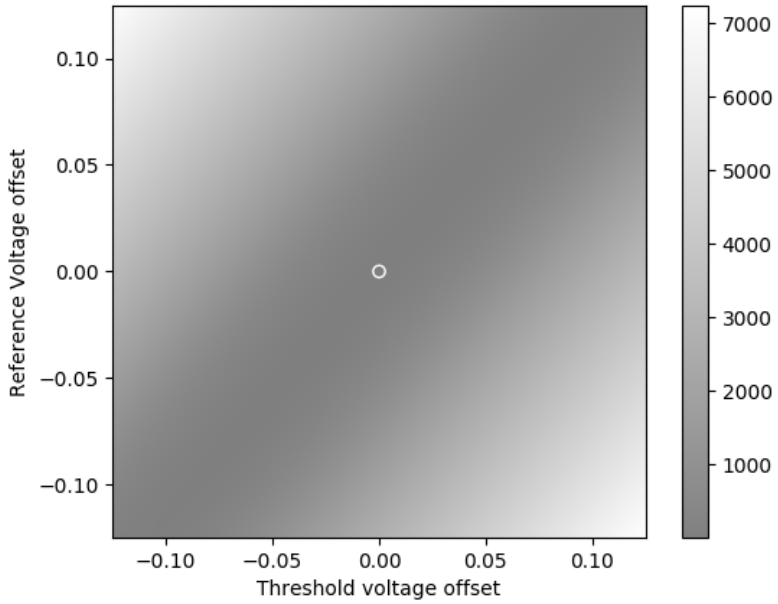


FIGURE 3.6. Sweep of the sum of the cost functions of many configurations. The axes correspond to the offset of threshold voltages and the reference voltages.

3.4 Correction equation

The correction will be performed by inverting the estimated pipeline transfer function of each stage and propagating the quantized value of the last sub-ADC, connected to the output last stage, to the input node of the pipeline ADC. In order to do this, a mathematical model of the pipeline ADC is required. The model for this work is the pipeline analogous to the $\Delta\Sigma$ model presented in 3.1. For a specific code from the stage sub-ADC, the output of the stage will be:

$$V_{out} = \eta \left(\sum_{s \in 0}^{N-2} \frac{C_s}{C_{N-1}} \left(V_{in} - V_{ref_s} \right) \right) \tag{3.7}$$

where V_{ref_s} is the reference voltage that corresponds to the code. This model assumes that the last capacitor, C_{N-1} is being used as feedback capacitor. If other capacitor is being used in feedback configuration instead, the indexes has to be adjusted accordingly. Inverting this model yields the following equation:

$$V_{in} = \frac{V_{out} + \eta \sum_{s=0}^{N-2} \frac{C_s}{C_{N-1}} V_{ref}}{\eta \left(1 + \sum_{s=0}^{N-2} \frac{C_s}{C_{N-1}} \right)} \tag{3.8}$$

The estimation of the input has to be processed with higher numeric precision that the resolution of the ADC, to prevent numerical errors arising from finite precision. The estimation of the input involves the steps depicted in figure 3.7:

1. The ideal quantized value associated with the code of the last sub-ADC is used as a starting point for iteration. This value represents the estimated output voltage (V_{out}) of the last stage. The ideal quantized value is used since there is no estimation of the sub-ADCs thresholds, otherwise the quantized value associated with estimated code-bins could be used.
2. From the last to the first stage, the V_{out} of that stage is applied to the inverted transfer function, yielding the input voltage (V_{in}) of that stage and the V_{out} of the previous stage. Directly inverting transfer function is not possible because for an input value many output values may be possible, but for a specific code, the function can be inverted which yields equation (3.8), this is shown in figure 3.2 as a gray area in the inverted transfer function.
3. After iterating though all stages, the obtained value is the estimation of V_{in} of the first stage. This high precision decimal value has to be digitally quantized to output the calibrated code of the ADC out of the device.

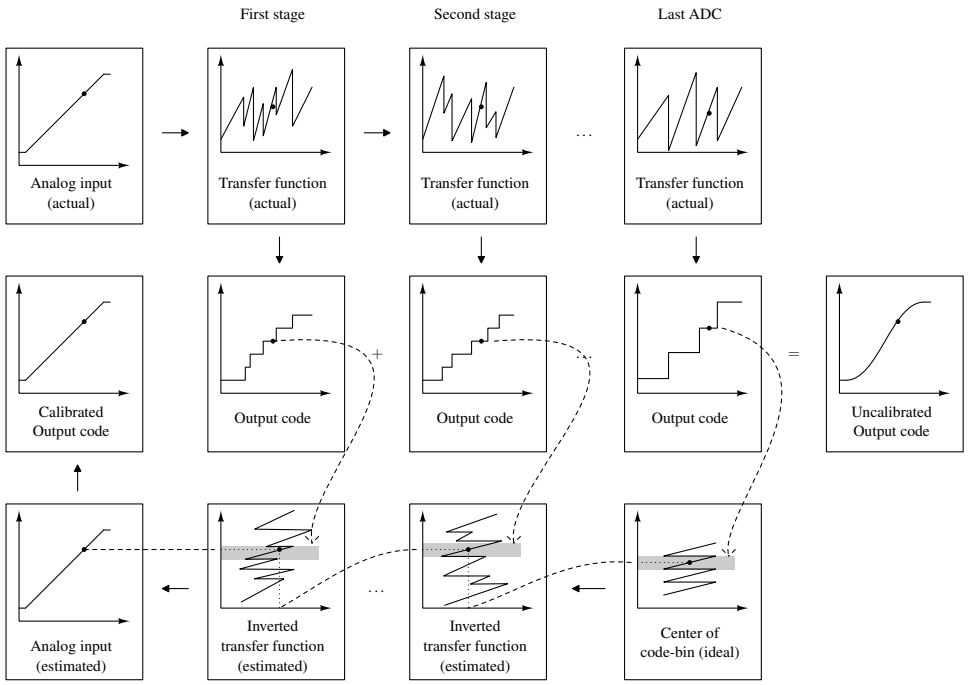


FIGURE 3.7. Process of applying the estimated parameters to the output of a conversion. The code of each stage is used to select the appropriate range of the inverted transfer function. The quantized value of the last ADC is propagated to the input of the first stage. The estimated input is quantized to the ADC resolution.

4. SIMULATION

The algorithm was tested using system level simulations. The simulation model employed for simulations is the same presented in 3.1 with the addition of Gaussian noise used to simulate thermal noise. This noise is applied to the voltage references of the capacitors and the threshold voltages of the sub-ADC comparators. The system-level simulations were performed in Python 3.7.3 using Numpy 1.16.4 and Numexpr 2.6.10dev0 for acceleration.

4.1 Simulated tests

A number of instances of a pipeline ADC were simulated, with stages of $[3.5, 3.5, 2.5]$ bits each one and a final flash converter of 4 bits. The nominal value of the capacitors is not important, since for pipeline and $\Delta\Sigma$ modes the conversion operation is capacitor-ratioed. To aid the convergence of solutions when optimizing, the capacitors nominal value is set to $1F$. The mismatch of capacitors is variable and depends on size, type and technology. Sin et al. (2009) found mismatch in the range of $0.05\% - 0.25\%$ for parallel capacitors. A mismatch in the range of $0.8\% - 1.2\%$ for fringe capacitors was found by Tripathi and Murmann (2014). And Omran, Alahmadi, and Salama (2016) found a mismatch in the range of $0.11\% - 0.3\%$ for different MoM capacitors. Taking into account all this information a mismatch of 0.25% was used.

The mean of η for each stage is selected to ensure the pipeline residual settles with at least half of the least significant bit (LSB) of the resolution of the remainder of the ADC: $\mu(\eta)_h = MAX(0.95, 1 - LSB(1 + \sum_{j=h+1}^{H+1} \lfloor bits_h \rfloor)/2)$. This factor is modeled to distribute lognormal with a standard deviation equivalent to an underlying settling time normal distribution with half a time-constant of standard deviation.

Three tests were performed, each test is a bivariate sweep of simulation parameters. For each point of the sweep, four ADCs are instantiated in a Monte-Carlo fashion. Each ADC is simulated

and the calibration algorithm is executed on each stage of the ADC. Then the un-calibrated and calibrated ENOB are computed, and the Monte-Carlo results are aggregated into the mean.

For the three tests one of the variables swept is the sub-ADC threshold voltages' standard deviation (actual stdev.). The sweep range of this variable if from zero to $1/6\ LSB$. The other variable swept for each test is: *Test 1:* sub-ADC threshold voltages standard deviation (modeled stdev.), *Test 2:* MDAC voltage references noise and *Test 3:* sub-ADC threshold voltages noise. These variables were selected to measure the effect of choosing a wrong deadband size, measure the effect of the estimated parameters' noise and measure the effect of the deadband noise, respectively. The noise standard deviation (σ) sweep range starts at $\sigma = 0$ and ends at $3\,\sigma = 11/250$ of the full scale range for each stage. For Test 1 the ADC simulated is noiseless and for Test 2 and 3 the modeled stdev. is zero because, as it will be shown in the next section, the algorithm performs best when the tolerance parameter of the cost function is set to zero.

These calibration simulations took 2 days to compute in a 8-core machine at 3.20GHz.

4.2 Simulation results

The results are plotted with the actual threshold variance sweep as the x axis. The aggregated ENOB is plotted in the y axis and each test specific sweep is plotted as different curves. For comparison purposes, two more curves are added to the plot. The naive ENOB, which is obtained by using the ideal electrical parameters for signal reconstruction. And the perfect knowledge ENOB, which is obtained by using the real electrical parameters for reconstruction. The mean for each aggregated value is plotted in a solid line while one standard deviation is plotted with a dashed line around the mean.

As seen in figure 4.1, it does not matter what the actual threshold variance is. The best calibration is achieved when no threshold variance is assumed by the algorithm, which implies a

tolerance parameter (tol) of zero for the cost function. Because of this, the next tests will all use $tol = 0$.

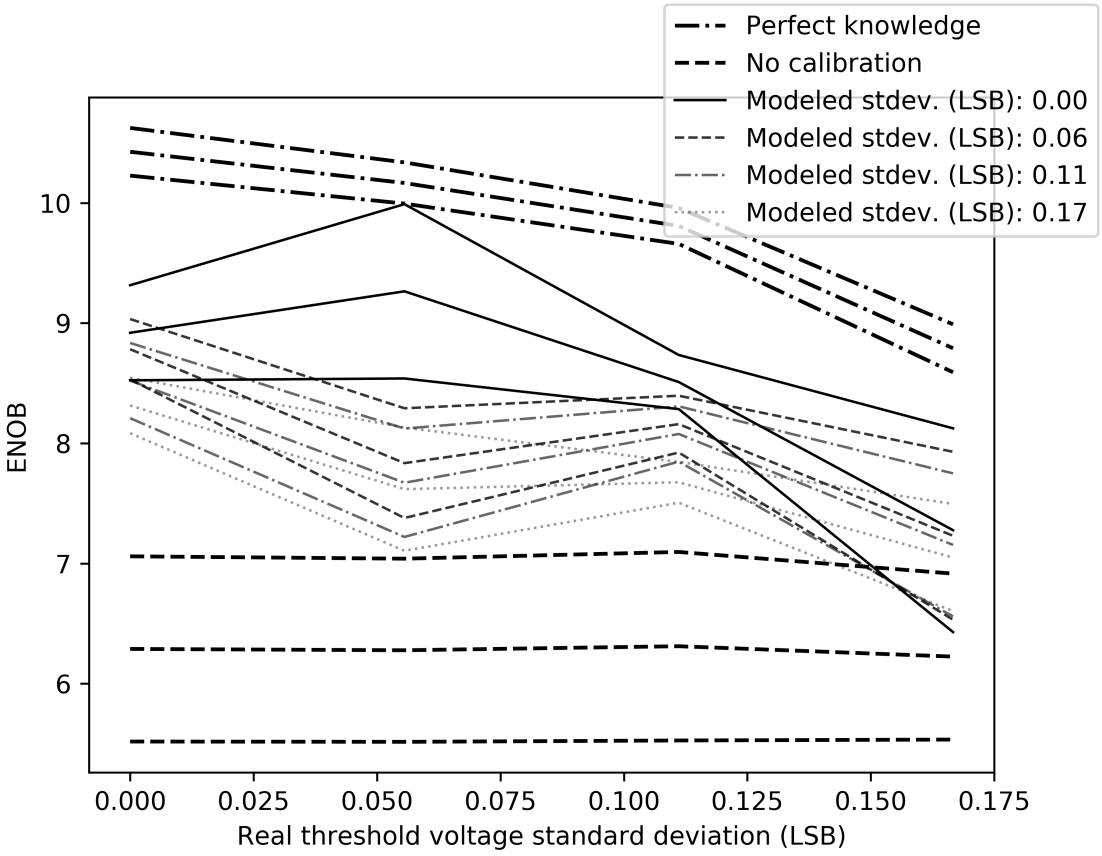


FIGURE 4.1. Calibrated and uncalibrated ENOB for different threshold variance assumptions.

The results of test 2 can be seen in figure 4.2. The calibration difference between no noise and the maximum amount of simulated noise is less than $\frac{1}{4}LSB$.

The results of test 3 can be seen in figure 4.3. The calibration difference between no noise and the maximum amount has a maximum of $1LSB$.

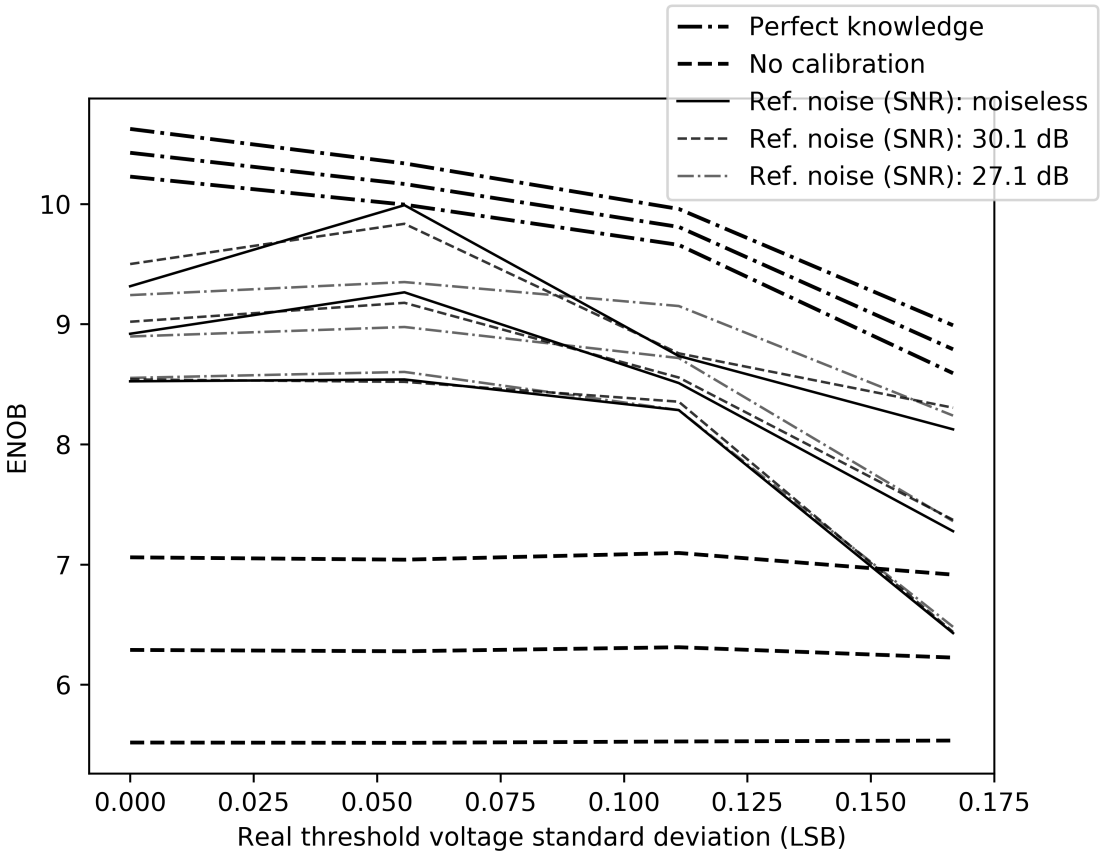


FIGURE 4.2. Calibrated and uncalibrated ENOB for different MDAC reference noise levels simulation sweeps.

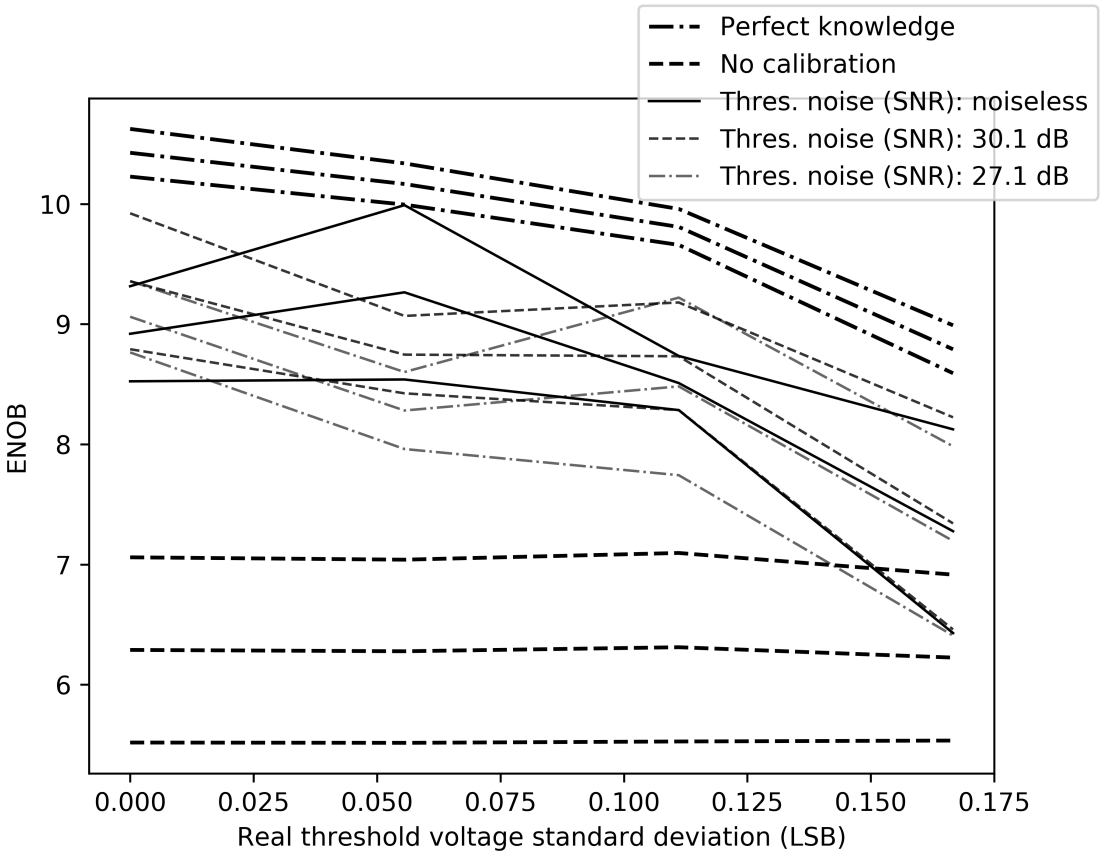


FIGURE 4.3. Calibrated and uncalibrated ENOB for different threshold noise levels simulation sweeps.

5. CONCLUSION

A novel ADC calibration scheme for SC pipeline ADCs, based on internal components reconfiguration, has been presented. The calibration scheme does not require any external references or dedicated circuitry to be used as *true ground*. But relies on the internal sub-ADC to compute bands of voltages where the estimated integral residual should lie.

The assumption that the sub-ADC threshold voltages are ideal undermines the hypothesis that suggests that it is possible to use this algorithm to calibrate the ADC without assuming or knowing beforehand any circuit parameter. As shown by the simulated results, the calibration algorithm tolerates some variance in the values of the actual threshold voltages without significant loss in calibration precision. However, a systematic error in the threshold values or a significant amount of variance of the threshold values will impact the calibration performance. This assumption of the threshold values may be eliminated and make the algorithm assumption-free if a way of introducing the threshold values can be found without creating the unbounded solutions explained in 3.3.1.

Even if the sub-ADC noise uncertainty is incorporated into the algorithm model, the calibration performs best when a noiseless sub-ADC is assumed. For for the simulated dataset, the algorithm is capable of calibrating to an ENOB up to 1 bits of an exact calibration for low levels of sub-ADC threshold and MDAC references voltage noise.

The effect of the reference noise on the calibration is small and no clear trend could be deduced from the simulated data set. On the other hand, some degradation of the algorithm performance has been noticed from the simulations with the increase of sub-ADC threshold noise.

5.1 Strengths and weaknesses of the algorithm

One strength of the calibration scheme is that it does not require external stimuli in order to perform calibration. And since the input signal is also being estimated, the precise voltages values do not need to be known. This relaxes the design constraints on the elements generating the input signal, allowing for static variance in those circuital elements and as seen in the results, noise in the MDAC's voltage references does not impact significantly in the calibration quality.

Another strength of this algorithm is that the resolution increase after calibration does not depend on the resolution of an external calibration circuitry, such as a signal generator or another ADC. This means that the algorithm may be applied even for circuits that have been aged or are partially damaged.

The main drawback of the proposed algorithm is the amount of computing resources required to perform calibration. The minimization optimization requires to compute the gradient and jacobian of a function of $O(2^B)$ dimensions of a sum involving $O(2^B \times configurations \times samples)$ elements. The time it takes to solve a calibration problem in a computer, which is in the order of hours, greatly exceeds the acceptable amount of time usually dedicated to devices calibrated by foreground algorithms. These are measured in the order of seconds, since the algorithms are usually run with the start-up of the system they are embedded into.

Another drawback of this algorithm is the dependence on the sub-ADC static variance and noise. It may be argued that this is the source of the algorithm precision, since deviations of this element from non-idealities have the biggest effect on calibration results. This limits the application of the algorithm to systems where error is not dominated by the sub-ADC threshold voltages.

5.2 Products of this work

A paper presenting this work entitled “A blind calibration scheme for switched-capacitor pipeline analog-to-digital converters” was submitted and accepted for the Latin American Symposium on Circuits and Systems (LASCAS) of 2020.

5.3 Future work

Future development of the algorithm should be centered around the main two weaknesses of the current scheme: Processing resources and the calibration performance dependence on the sub-ADC threshold voltages.

The first issue could be addressed by using an optimization scheme suitable for the specific mathematical problem the algorithm purposes. The optimization method used in this work, least squares gradient descend, is a generic method that has many applications and assumes few properties of the functions being optimized. By the same token, the optimization algorithm does not exploit the nature of the equations to converge faster to a solution. The specific model used in this work has a simple form of sum of polynomials which may be exploited by other optimization methods.

Another way the processing resources could be reduced is by using an optimization method that iterates through one deadband constraint a time, instead of all of them at the same time. A method like projection onto convex sets (POCS) could be used if the set of parameters that satisfies a specific deadband could be transformed in order to make that set convex. Since right now the constraints are expressed as a sum of products, the set that satisfies those conditions is not convex.

The dependence of the calibration precision on the number of unique configurations used to collect data for parameter estimation was not studied in this work. It is possible that fewer configurations than the ones simulated could obtain similar results. This would also reduce the

computational resources required to perform a calibration.

An important question that remains is if it is possible to include the sub-ADC threshold voltages into the estimated parameters. Previously it has been shown that for the current problem formulation, the addition of those variables added local minima outside the actual solution and reduces the calibration effectiveness. It may be possible that a different formulation of the problem could include the threshold voltages to the estimation system. This would eliminate the need to assume ideal values for the deadband voltage ranges and for the quantized value used to propagate to the input of the first stage when applying the correction procedure of section 3.4.

REFERENCES

Abou-El-Kheir, N. T., Abbas, M., & Khedr, M. E. (2014, July). An adaptive digital background calibration technique using variable step size lms for pipelined adc. In *2014 9th international symposium on communication systems, networks digital sign (csndsp)* (p. 835-840). doi: 10 .1109/CSNDSP.2014.6923943

Feely, O., & Chua, L. O. (1991, Nov). The effect of integrator leak in sigma - delta modulation. *IEEE Transactions on Circuits and Systems*, *38*(11), 1293-1305. doi: 10.1109/31.99158

Gines, A. J., Peralias, E. J., & Rueda, A. (2009, Aug). A survey on digital background calibration of adcs. In *2009 european conference on circuit theory and design* (p. 101-104). doi: 10.1109/ECCTD.2009.5274976

Goes, J., Vital, J. C., & Franca, J. E. (1996, May). A cmos 4-bit mdac with self-calibrated 14-bit linearity for high-resolution pipelined a/d converters. In *Proceedings of custom integrated circuits conference* (p. 105-108). doi: 10.1109/CICC.1996.510522

Ieee standard for terminology and test methods for analog-to-digital converters. (2011, Jan). *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)*, 1-139. doi: 10.1109/IEEESTD.2011 .5692956

Lewis, S., Fetterman, H., Gross, G., Ramachandran, R., & Viswanathan, T. (1992). A 10-b 20-msample/s analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, *27*(3), 351–358. doi: 10.1109/4.121557

Murmann, B. (2019). *Adc performance survey 1997-2019*. ([Online; accessed 25-July-2019])

Omran, H., Alahmadi, H., & Salama, K. N. (2016, June). Matching properties of femtofarad and sub-femtofarad mom capacitors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *63*(6), 763-772. doi: 10.1109/TCSI.2016.2537824

Pelgrom, M. (2016). *Analog-to-digital conversion*. Springer. Retrieved from https://www.xarg.org/ref/a/3319449702/

Sin, S., Wei, H., Chio, U., Zhu, Y., Seng-Pan, U., Martins, R. P., & Maloberti, F. (2009, Nov). On-chip small capacitor mismatches measurement technique using beta-multiplier-biased ring oscillator. In *2009 ieee asian solid-state circuits conference* (p. 49-52). doi: 10.1109/ASSCC.2009.5357165

Tripathi, V., & Murmann, B. (2014, Aug). Mismatch characterization of small metal fringe capacitors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *61*(8), 2236-2242. doi: 10.1109/TCSI.2014.2332264

Tzi-Hsiung Shu, Bang-Sup Song, & Bacrania, K. (1995, April). A 13-b 10-msample/s adc digitally calibrated with oversampling delta-sigma converter. *IEEE Journal of Solid-State Circuits*, *30*(4), 443-452. doi: 10.1109/4.375965