NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATERIALS SCIENCE

DIOGO FILIPE DA CRUZ MARTINS

Bachelor in Micro and Nanotechnology Engineering

DESIGNING A SIGMA-DELTA MODULATOR USING PASSIVE INTEGRATORS AND CALIBRATION

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon

October, 2023

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATERIALS SCIENCE

DESIGNING A SIGMA-DELTA MODULATOR USING PASSIVE INTEGRATORS AND CALIBRATION

DIOGO FILIPE DA CRUZ MARTINS

Bachelor in Micro and Nanotechnology Engineering

Adviser: Prof. Nuno Filipe Silva Veríssimo Paulino
Associate Professor with Aggregation, NOVA University Lisbon

Co-adviser: Dr. Blazej Nowacki
Senior Analog Designer, Renesas Electronics Corporation

Examination Committee

Chair: Pedro Miguel Cândido Barquinha
Associate Professor, NOVA University Lisbon

Rapporteur: Hugo Alexandre de Andrade Serra
Invited Associate Professor, NOVA University Lisbon

Adviser: Prof. Nuno Filipe Silva Veríssimo Paulino
Associate Professor with Aggregation, NOVA University Lisbon

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon
October, 2023

Designing a sigma-delta modulator using passive integrators and calibration

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.5) [1].

To my mother.

Acknowledgements

First and foremost, I strongly acknowledge my Adviser Prof. Nuno Paulino and my Co-adviser Dr. Blazej Nowacki.

I am deeply grateful to Prof. Nuno Paulino for his availability, motivation and exceptionally valuable remarks. The insightful remarks have been instrumental in overcoming the many challenges that this work presented.

I would also like to state my thanks to Dr. Blazej Nowacki for his unwavering support and dedication throughout our journey together. Dr. Blazej Nowacki commitment to assisting me has been nothing short of exceptional and without his help, this work would not have been able to be developed.

I would like to express my gratitude to Renesas Electronics Corporation for the opportunity to work on such interesting project. It was genuinely a enriching experience, not only in terms of knowledge but also in connections made with people from the Lisbon Office in Renesas, who were always ready to lend a helping hand whenever I required assistance.

I would like to acknowledge the outstanding professors within the Electrical Engineering and of Materials Science of the School of Science and Technology of Universidade NOVA de Lisboa. They inspired a passion for learning that will continue to influence my academic and professional journey.

Finally, I would give my deepest thanks for my friends and family for supporting me throughout my life. To Ana Júlia Maximiliano, an unwavering pillar of strength who, even during the most trying times, consistently extended her support, motivation, and boundless patience. To Guilherme Santos, a dependable friend for exchanging thoughts and encouragement whenever I needed it most. To Mariana Cordeiro, for always being available to help even if it came at personal cost. Last but not least, to my mother Ana Catarina, who always prioritized my well being and demonstrated remarkable patience even when I was not the best state of mind.

”

“Aut inveniam viam aut faciam”

— Hannibal Barca

Abstract

An Analog to Digital Converter (ADC) serves as vital components bridging the analog and digital domains, facilitating the conversion of real-world data into digital format. Due to its essential role across various applications, the ADC is extensively researched in electronics. Ensuring their performance remains aligned with the ever-evolving technological landscape is crucial. To achieve this, optimization algorithms are employed to enhance architectural efficiency. This study explores the potential of a passive 2-1 Multi-stAge noise SHaping (MASH) Sigma-Delta Modulator ($\Sigma\Delta$M) architecture as presented in [2], complemented by calibration techniques detailed in [3]. Although the architecture [2] has previously undergone optimization, calibration has not been used. The introduction of calibration methods has the potential to improve performance. To tackle this issue, a genetic algorithm was implemented using the pre-existing MATLAB® code with modifications. Additionally, a C++ code was developed to enhance speed, enabling a more thorough analysis. The primary objective was to enhance the 2-1 MASH $\Sigma\Delta$M while minimizing Signal-to-Noise-and-Distortion Ratio (SNDR) degradation in the presence of process variations. Monte Carlo simulations were conducted to validate performance improvements. This thesis proposes a redesigned 2-1 MASH $\Sigma\Delta$M based on [3].

Keywords: Analog to Digital Converter(ADC), Sigma-Delta Modulator($\Sigma\Delta$M), 2-1 MASH $\Sigma\Delta$M, Calibration, Genetic Algorithm.

Resumo

Um ADC atua como componente vital, fazendo a ponte entre os domínios analógico e digital, facilitando a conversão de dados do mundo real para o formato digital. Devido ao seu papel essencial em várias aplicações, o ADC é extensivamente pesquisado em eletrónica.

Garantir que o seu desempenho esteja alinhado com o cenário tecnológico que está em constante evolução é crucial. De forma a que isso seja alcançado, algoritmos de otimização são utilizados para aprimorar a eficiência de arquiteturas. Este trabalho explora o potencial de um 2-1 MASH Modulador Sigma-Delta (MΣΔ) passivo, conforme apresentado em [2], complementado por técnicas de calibração detalhadas em [3].

Embora a arquitetura [2] tenha passado por otimização anteriormente, calibração não foi utilizada. A introdução de métodos de calibração tem o potencial de melhorar o desempenho. Para abordar isso, foi usado um algoritmo genético utilizando o código existente em MATLAB®, com modificações. Além disso, foi desenvolvido um código em C++ com maior velocidade, permitindo uma análise mais abrangente. O objetivo principal era aprimorar o MΣΔ 2-1 MASH, minimizando a degradação de SNDR na presença de variações de processo. Simulações de Monte Carlo foram realizadas para validar as melhorias de desempenho. Esta tese propõe um redesenho do MΣΔ 2-1 MASH baseado em [3].

Palavras-chave: Analog to Digital Converter(ADC), Modulador Sigma-Delta (MΣΔ), 2-1 MASH MΣΔ, Calibração, Algoritmo Genético

Contents

List of Figures

List of Tables

Acronyms

ΣΔ	Sigma Delta *(p. 4)*
ΣΔM	Sigma Delta Modulator *(pp. 1, 4)*
ADC	Analog to Digital Converter *(pp. v, vi, 1)*
CMOS	Complementary Metal-Oxide-Semiconductor *(p. 1)*
CT-ΣΔM	Continuous-Time ΣΔM *(p. 4)*
DC	Direct Current *(p. 6)*
DCL	Digital Cancellation Logic *(p. 6)*
DT-ΣΔM	Discrete-Time ΣΔM *(p. 4)*
ENOB	Effective Number of Bits *(p. 3)*
FoM	Figure of Merit *(p. 9)*
GA	Genetic Algorithm *(p. 8)*
HLM	High Level Model *(p. 12)*
IoT	Internet of Things *(p. 1)*
MASH	Multi-stAge noise SHaping *(pp. v, vi, 1)*
MC	Monte Carlo *(p. 13)*
OS	Output Swing *(p. 12)*
OSR	Oversampling Ratio *(p. 3)*
S/H	Sample and Hold *(p. 4)*

SAR Sucessive Approximation Register *(p. 3)*
SNDR Signal-to-Noise-and-Distortion Ratio *(pp. v, vi, 1, 3)*
SNR Signal-to-Noise Ratio *(p. 2)*

1 Introduction

1.1 Background Motivation

In our ever-changing world, an ADC has become integral for maintaining global connectivity through the Internet, commonly known as the Internet of Things (IoT). To sustain and enhance this connectivity, continuous improvements to the ADC is crucial. These improvements should focus on key aspects like area efficiency, power consumption, and the speed of analog-to-digital signal conversion. The demand for lower power consumption is particularly critical, given the increasing trend towards portable electronic circuits powered by an even smaller battery [4].

To achieve faster processing of analog information, reduce the required conversion area, and minimize power consumption in the ADC, it is essential to explore and optimize new architectural designs tailored to the specific frequency characteristics of the signals involved [5].

1.2 Contributions

This thesis describes the development of an optimization procedure base on a genetic algorithm for a two-stage 2-1 MASH Sigma Delta Modulator ($\Sigma\Delta$M) with 14-bit resolution, 12 MHz bandwidth, and 1.6 GHz clock frequency, originally described in [2]. The $\Sigma\Delta$M has been subsequently redesigned using UMC Complementary Metal-Oxide-Semiconductor (CMOS) 40 nm Low-Power technology with a new supply voltage of 1.1V.

In [6], a calibration method is introduced to enhance the performance of the 2-1 MASH $\Sigma\Delta$M. Following an analysis of the calibration and the 2-1 MASH $\Sigma\Delta$M, the genetic algorithm from [2] was modified to incorporate calibration and make specific code adjustments.

These adaptations result in a $\Sigma\Delta$M which achieves higher SNDR, is less less sensitive to component variations, and has lower output swing in integrators.

2

State of the Art

2.1 Analog to Digital Converters

An ADC converts analog signals to digital ones. The analog signal is, essentially, being discretized. Discretization level is dependent on the number of bits of the ADC.

Since an ADC cannot have infinite bits, there will always be a quantization error V_q. This error is dependent on the value of V_{LSB} and will be, ideally, between $[\frac{-V_{LSB}}{2}, \frac{V_{LSB}}{2}]$ if saturation of the ADC does not occur. V_{LSB} correlates with the number of bits in the following equation:

$$V_{LSB} = \frac{V_{REF}(V)}{2^{nbits}} \tag{2.1}$$

where V_{REF} is the reference voltage by which the ADC computes digital values, and $nbits$ is the number of bits of the ADC

Considering that the input signal has a random component with an amplitude equal to V_{LSB}, quantization error becomes uniformly distributed between $[\frac{-V_{LSB}}{2}, \frac{V_{LSB}}{2}]$. The input signal can now be seen as having a random signal independent from it. Thus, quantization error becomes quantization noise.

Therefore, in order to calculate the quantization noise, the subsequent equation is used:

$$V_{Qrms}^2 = \frac{V_{LSB}^2(V^2)}{12} \tag{2.2}$$

A metric used to provide valuable insight into the ratio of the desired signal's strength to the noise level within the system is the Signal-to-Noise Ratio (SNR), which is defined by the following equation:

$$SNR = 10 \log \left(\frac{P_s(W)}{P_{NQ}(W)} \right) \tag{2.3}$$

where P_s is the signal power and P_{NQ} is the noise power.

Utilizing equation (2.3) and taking into account that maximum signal power occurs when the signal amplitude is equal to $\frac{V_{REF}}{2}$, considering that $V^2_{Qrms} = P_{NQ}$ and using all previous equations (2.1) and (2.2), SNR_{MAX} in dB for an ideal ADC and is determined by:

$$SNR_{MAX} = 6.02 \times nbits + 1.76 \tag{2.4}$$

SNR_{MAX} for an ideal ADC is therefore, limited by $nbits$, the number of bits. Although it is possible to increase $nbits$ by using a oversampling ADC, it also decreases our bandwidth compared to the achievable one with a Nyquist rate ADC.

A metric that will indicate to us how close the ADC's real-world performance is to that of an ideal scenario, the equation (2.4) would need to be solved for $nbits$ obtaining the Effective Number of Bits (ENOB):

$$ENOB = \frac{SNR(dB) - 1.76}{6.02} \tag{2.5}$$

It is also worth noting that if equation (2.3) also takes into consideration the effect of distortion on the signal, as presented in the equation (2.6) as the distortion power P_D, results in the expression of the SNDR:

$$SNDR = 10 \log \left| \frac{P_s(W)}{P_D(W) + P_{NQ}(W)} \right| \tag{2.6}$$

One of the most famous architectures in Nyquist rate ADCs is Sucessive Approximation Register (SAR) ADC since it has a lower power consumption when compared to flash ADCs. While this type of ADC typically employs a SAR-assisted pipeline architecture, which offers the advantage of more relaxed comparator noise performance when compared to using just the SAR alone [7], it is essential to note that for such ADCs, ensuring alias-free sampling in Nyquist rate ADCs requires the signal frequency to satisfy the inequality $f_{\text{Signal}} < f_{\text{sampling}}/2$ [8].

As previously stated, oversampling ADCs have a distinct advantage over Nyquist rate ADCs. They often employ fewer bits while achieving significantly higher resolution. This improvement is made possible by utilizing a substantially higher sampling frequency denoted as $f_{sampling}$, well beyond the Nyquist rate. The Oversampling Ratio (OSR), a key parameter in these systems, is defined as:

$$OSR = \frac{F_s(Hz)}{2B(Hz)} \tag{2.7}$$

Where F_s is the sampling frequency of the ADC and B is the signal bandwidth of the signal. Since oversampling ADCs have a higher sampling frequency $f_{sampling}$, it facilitates the design of an anti-aliasing filter because the order of the anti-alias filter is dependent on the desired alias suppression and on the ratio between the signal and the alias band.

The main types of oversampling modulators are the delta modulator and sigma-delta modulator. However, the focus of this work will be on the latter.

2.2 Sigma-delta modulators

Sigma Delta (ΣΔ) architecture is based on oversampling and noise shaping [9]. ΣΔ ADC has an averaging behavior that prohibit its application where sample-to-sample conversion is required [10]. This type of converter is also extremely appealing for designing ADCs due to having a high resolution combined with a low production cost [11].

ΣΔ ADCs are not exclusively composed of ΣΔM; they also necessitate the incorporation of other critical elements, including analog filters and digital filters. This comprehensive system configuration is depicted in Figure 2.1, based on an illustration from [9].

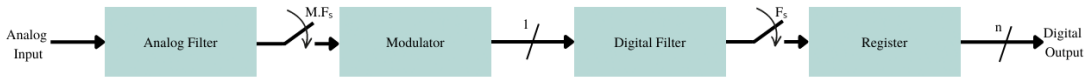

Figure 2.1: Schematic of Sigma Delta ADC

Using the model presented in [9], Figure 2.2 shows a first-order ΣΔM, which consists of an ADC, DAC, and an integrator. These systems typically employ a loop filter with high gain in the signal band, resulting in the strong attenuation of in-band quantization noise within this frequency range. This phenomenon is referred to as *noise shaping* [9].

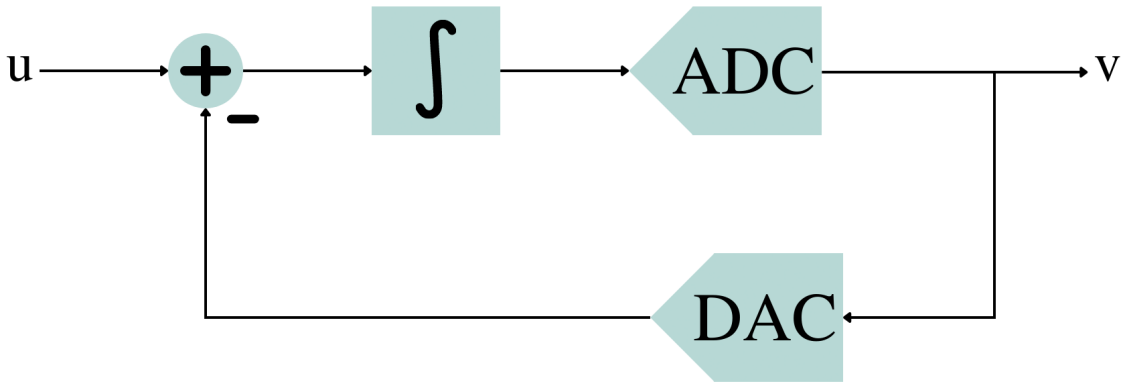

Figure 2.2: Schematic of first order ΣΔM

The ΣΔM can be sub-categorized into Continuous-Time ΣΔM (CT-ΣΔM) or Discrete-Time ΣΔM (DT-ΣΔM). After the input is filtered by the anti-aliasing filter, it is sampled by the Sample and Hold (S/H) block at point u, as depicted in Figure 2.2. From there, the signal continues its journey through the rest of the circuit, briefly describing how a CT-ΣΔM works.

Concerning a CT-ΣΔM, even though a continuous time integrator is being used, the discretization of the signal still occurs. Consequently, the output signal remains in discrete time, much like that of the CT-ΣΔM.

The key advantages that the CT-ΣΔM offers, when compared to CT-ΣΔM, are higher maximum sampling frequencies, a higher speed, a lower power consumption, a lower thermal noise, and an intrinsic anti-aliasing filtering [12], [13], [14]. On the other hand, CT-ΣΔM are less susceptible to noise and are easier to implement due to using switched-capacitor circuits [15].

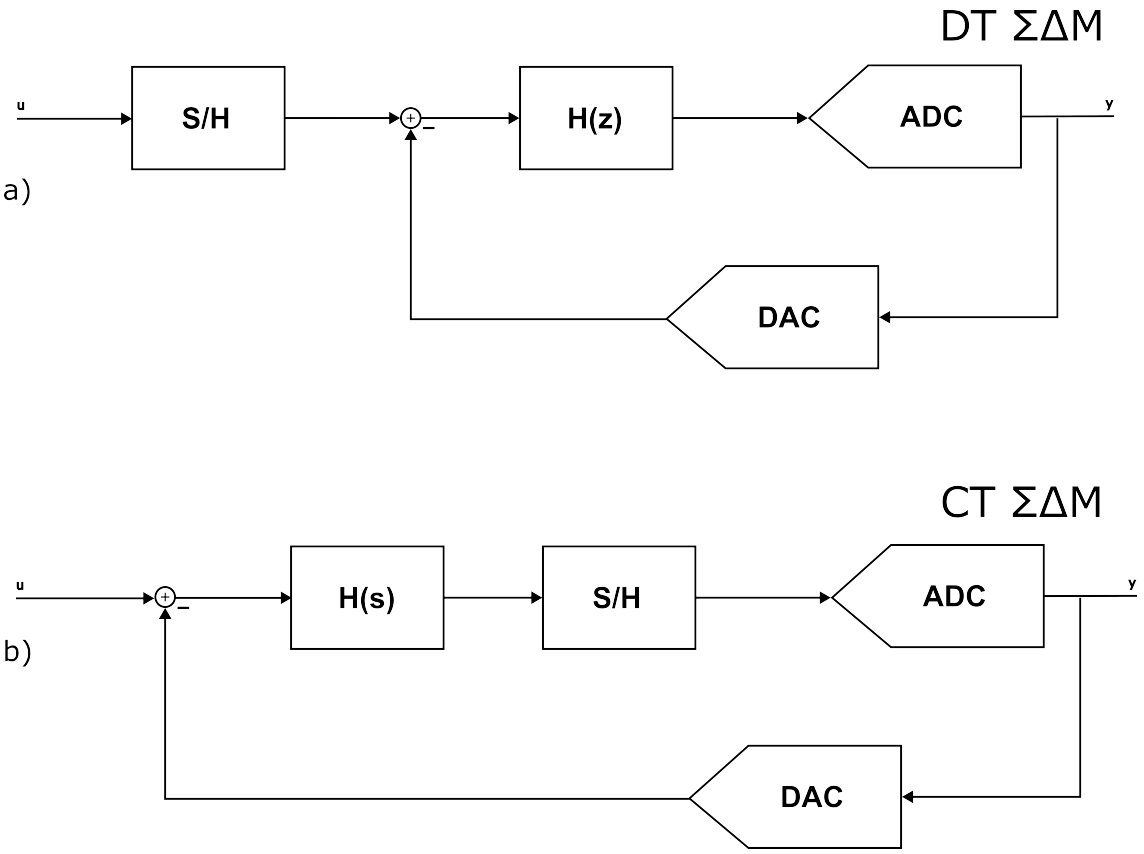

Figure 2.3: Schematic of a CT-ΣΔM and CT-ΣΔM

Depending on the elements that constitute the ΣΔM, it can be categorized as a passive ΣΔM or active ΣΔM. Active ΣΔM employ active elements, generally op-amps, allowing for the distribution of their gain across the integrators [2].

Passive ΣΔM use a passive integrator circuit instead of an active one. In contrast to active ΣΔM, the gain on passive ΣΔM is mainly concentrated within the quantizer. These modulators commonly employ a comparator as a single-bit quantizer instead of a multi-bit one. This choice is driven by the reduced input signal amplitude of the quantizer due to passive integrators only attenuating the signal [2].

Compared to the active ΣΔM, the passive ΣΔM has an improved linearity and a low power consumption [16]. Using them, however, also has the drawback of an increased area

used for low sampling frequencies[17] and the thermal noise being more significative [18]. Due to the latter, in order to achieve a higher SNR, a larger capacitor is needed [19]. On the other flip side, active ΣΔM offer in-band noise suppression since the first amplifier in the modulator provides a significant in-band gain, which will, in turn, suppress in-band noise and nonlinearities from subsequent circuits [20]. Another advantage is that, as long as Direct Current (DC) gains from the amplifiers are sufficiently large, the active ΣΔM will not be significantly affected by process variations, supply voltage fluctuations and temperature changes [20].

2.3 ΣΔM MASH ADC

A ΣΔ ADC's performance can be enhanced using MASH architecture, allowing for a higher-order noise-shaping [21]. The enhancement in performance can be done by cascading single-loop architecture. This architecture is also appealing because of its improved stability compared to single-loop structures [22].

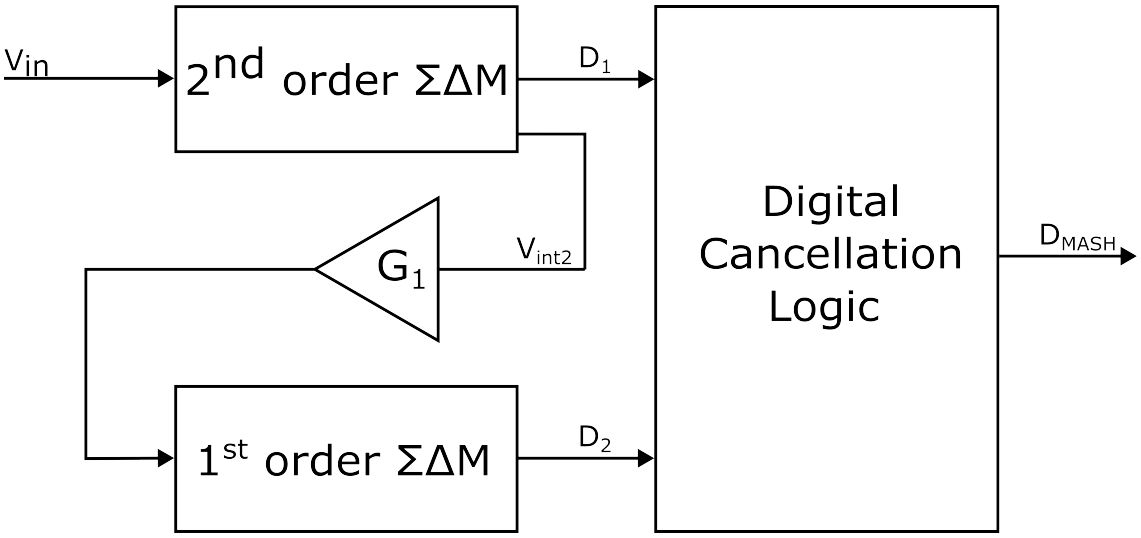

Figure 2.4: Block diagram of a MASH 2-1 ΣΔM

However, this architecture also has its challenges. The main challenge is that if the analog and digital transfer functions are not perfectly matched, performance will deteriorate significantly.

In a MASH architecture, one modulator's quantization noise is modulated by another, leading to increased noise order. This process can be cascaded, creating a cascade of higher-order noise-shaping stages. Using the schematic depicted in Figure 2.4, which shows the block diagram of a 2-1 MASH ΣΔM, it is possible to observe that the first order modulator is then modulating the quantization noise from the second order modulator. These two outputs are combined to cancel out the quantization error of the first modulator, resulting in a third-order noise shaping. These two outputs are combined by the Digital

Cancellation Logic (DCL) block. It is important to note, however, that real world imperfections or variations can affect DCL performance, which can impact the effectiveness of the quantization error cancellation. Therefore, MASH architectures often incorporate calibration mechanisms to achieve a better performance from the ADC, as was done in the case of this ADC.

2.4 2-1 MASH ΣΔM

This thesis aims to optimize the 2-1 MASH ΣΔM presented in [2] paired with the calibration system created in [3]. More specifically, this work aims to optimize the integrators from each ΣΔM present.

For better understanding, in this work, the denomination used for each integrator was Integrator1 and Integrator2 for the 1^{st} stage of the MASH and Integrator1A for the 2^{nd} stage. Their components and coefficients also follow this denomination.

The passive integrators used have switched-capacitor feedback, as shown in Figure 2.5. In this integrators, the feedback capacitor C_f , depending on the signal D, are pre-charged with V_{ref} or $-V_{ref}$.

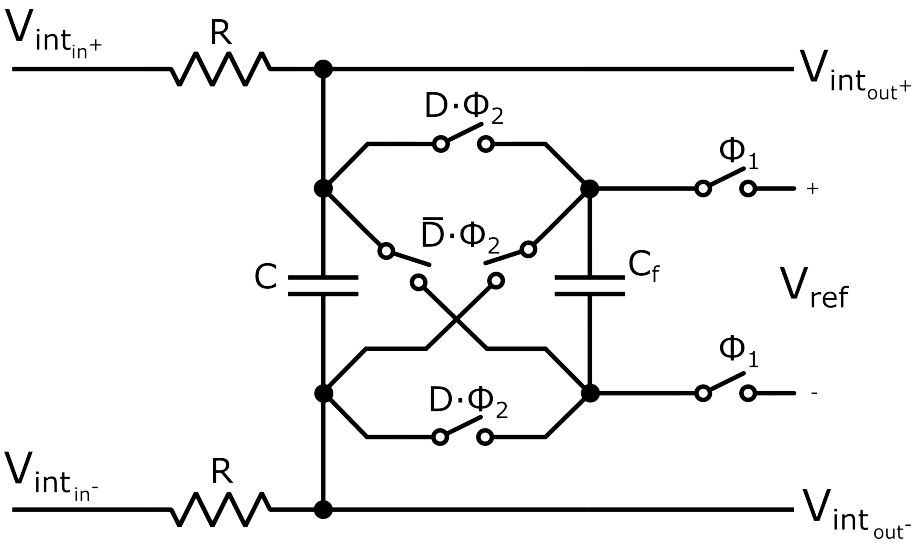

Figure 2.5: Schematic of the differential passive RC integrator with 1-bit feedback

According to [23], the Z transfer function for the passive integrator is as follows:

$$H(z) = \frac{Vint_{out}(z)}{Vint_{in}(z)} = \frac{\alpha}{1 - \beta.z^{-1}} \tag{2.8}$$

The expression for α and β can be found in [23]:

$$\begin{aligned} \alpha &= \frac{T_s}{2 \cdot R \cdot (C + C_f)} \\ \beta &\approx 1 - \alpha \end{aligned} \tag{2.9}$$

Where T_s in (2.9) is the sampling period. Knowing the sampling frequency, f_s;

$$T_s = \frac{1}{f_s} \tag{2.10}$$

The coefficient b given in [23], defines the integrator's feedback path:

$$b \approx \frac{V_{ref}}{\alpha} \cdot \frac{Cf}{C + Cf} = \frac{2 \cdot V_{ref} \cdot C_f \cdot R}{T_s} \tag{2.11}$$

This feedback coefficient, b, is significant since if it falls below the nominal value due to process variations, it can cause the output of the integrator to saturate, which, in turn, can lead to low-frequency oscillations driving the integrator outputs between the high and low saturation levels [2].

As previously stated, calibration is of utmost importance for MASH architecture since, if there is an imperfection in the outputs of any ΣΔM present in this architecture, when applied to the DCL block and if this was a two-stage MASH, the quantization error from the first modulator would not be fully cancelled. This imperfection could prove problematic for the system presented in [2], severely affecting performance. Due to that, this architecture was paired together with a measuring circuit developed in [3].

The measuring circuit will measure $\frac{RC}{T_s}$. Therefore, knowing T_s, it is possible to obtain the RC value in which the variation can change in a range of [-31% 37%] in the used CMOS technology, which is the UMC 40nm LP process.

To maintain the feedback coefficient above nominal, the value of V_{ref} will shift from 0.5V to 1.1V. Since the nominal value V_{ref} will be between these intervals, an increase in the feedback capacitors is needed. The specific level to which V_{ref} will transition depends on the measurement obtained from the measuring system. 1

2.5 Genetic Algortihm

A Genetic Algorithm (GA) is a robust optimization algorithm inspired by natural selection and genetic inheritance principles. It belongs to a larger class of evolutionary algorithms.

These algorithms harness the concept of evolution to iterative search to find the best solutions to complex problems. They also offer algorithms offer a time-efficient approach by strategically narrowing down potential solutions. Instead of exhaustively testing every possible solution, which would be time-consuming, they intelligently explore and optimize candidates, significantly reducing the overall time spent in finding effective solutions.

In GA, the population is a group of possible solutions, and each one is often referred to as a chromosome.

The GA is composed of four key processes that resemble the natural world:

1. **Selection:** Organisms with advantageous traits have a better chance to survive and reproduce. These traits have a better chance of passing their genes to the next generation. Similarly, in the GA, individuals with better fitness scores, which

measure their suitability for solving the problem, have a higher chance of being selected for reproduction.

2. **Crossover:** This process happens since, typically, to reproduce, there is a combination of genetic material from two individuals, the parents. These traits are inherited by their offspring and can inherently advantageous or disadvantageous characteristics. To better mimic this process in GA, the inheritance of traits involves assigning random values to determine which parent's traits will be passed on to the offspring.

3. **Mutation:** Genetic diversity is derived from the random mutations in an organism's DNA. These mutations can be beneficial or not for an organism. In GA, with the help of random values, an individual's genetic information can be modified.

4. **Replacement:** In nature, if an individual is not apt, it will not survive. Likewise, after these steps, the algorithm must decide which individuals in the population should be replaced. The number of individuals that survive depends on the parameters used for the algorithm. However, individuals with lower fitness are typically replaced with offspring, maintaining population diversity. In this work, the replacement was carried out using the elitism method, where the top 2 chromosomes were selected to be passed on to the next generation. To determine the offspring, the ranking method in GA was employed, wherein chromosomes were ranked according to their fitness scores. Chromosomes with higher ranks had a greater probability of being chosen.

After all these steps, the population is reevaluated, and the process repeats until an optimal value is obtained.

2.6 Literature Review

As previously mentioned, ADC need to be at a constant improvement in performance so it can keep up with the current world. One of the ways to measure this performance is with Figure of Merit (FoM). FOM take into account the main performance metrics of ADC. The FOM that will be considered is the following:

$$FOMS = SNDR(dB) + 10\log_{10}\frac{[B_\omega(Hz)]}{P(W)} \tag{2.12}$$

Wooley and Rabii proposed FOMS [24]. Suggested by Schreier and Temes in [9], (2.12) can be rewritten as:

$$FOMS = P_{sig|_{dBFS}} - NSD|_{dBFs/Hz} - 10\log_{10}[P(W)] \tag{2.13}$$

where $P_{sig|_{dBFS}}$ is the input signal power computed in DB referred to the full-scale range of the converter, and $NSD|_{dBFs/Hz}$ stands for the noise spectral density with P_{nd} being noise

plus distortion referred to full scale. For the ΣΔADC used on the electrical simulations the FOMS is 164 dB.

In order to be able to put ΣΔADCs into perspective, it is imperative to compare their performance with Nyquist-Rate ADCs.

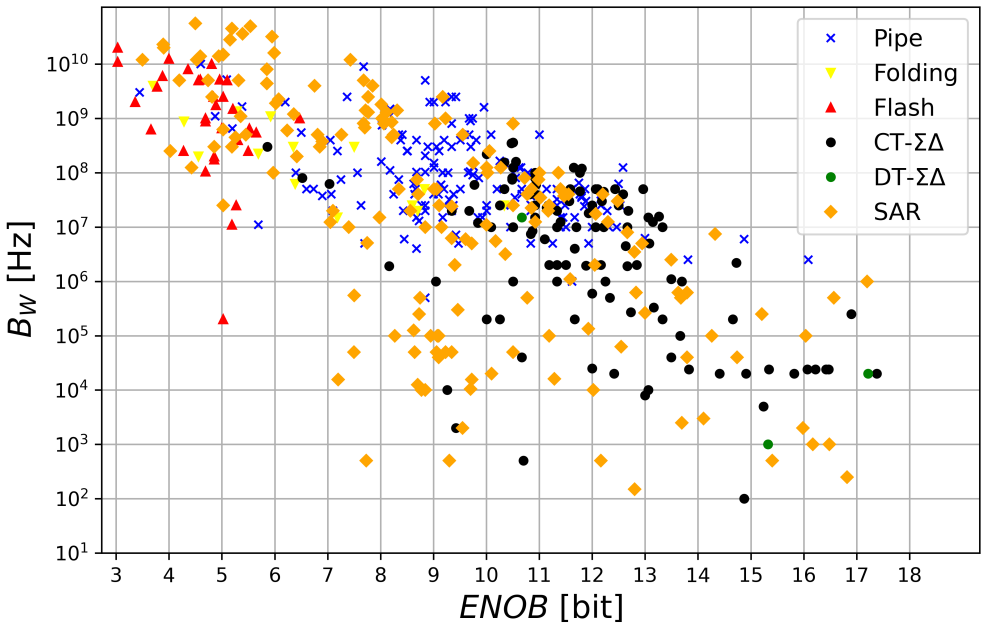

Figure 2.6: Aperture plot of state-of-the-art ADCs obtained from [25].

From Fig. 2.6, it is possible to conclude that the state of the art is dominated by ΣΔ, Pipeline and SAR ADCs.

The parameters depicted in Fig. 2.6 are not the sole considerations. As previously mentioned, power consumption represents another crucial factor. Achieving lower power consumption while maintaining identical results leads to improved efficiency. According to [26], efficiency can be quantified by measuring the amount of energy per converted sample. This concept is also referred to as conversion energy, E, and is defined by:

$$E = \frac{P(W)}{f_{snyq}(Hz)} \tag{2.14}$$

where P stands for power dissipated and f_{snyq} is the effective Nyquist Rate

In Fig. 2.6, it is possible to observe that CT-ΣΔADCs mainly cover a conversion region between $0.1MHz < B_\omega < 100MHz$ and $10bit < ENOB < 17bit$.

In Fig. 2.7, it is possible to conclude that SAR ADCs are the most energetically efficient.

In Fig. 2.8, we can assert that the most efficient designs are CT-ΣΔADCs, DT-ΣΔADCs and SAR ADCs being able to reach a FOMS of 180 dB.

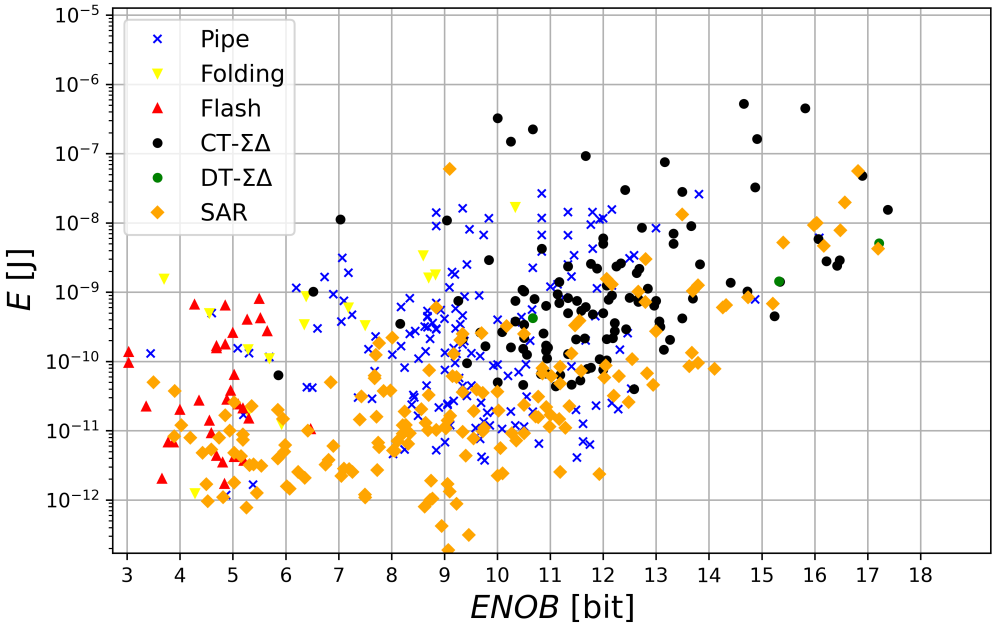

Figure 2.7: Energy plot of state-of-the-art ADCs obtained from [25]

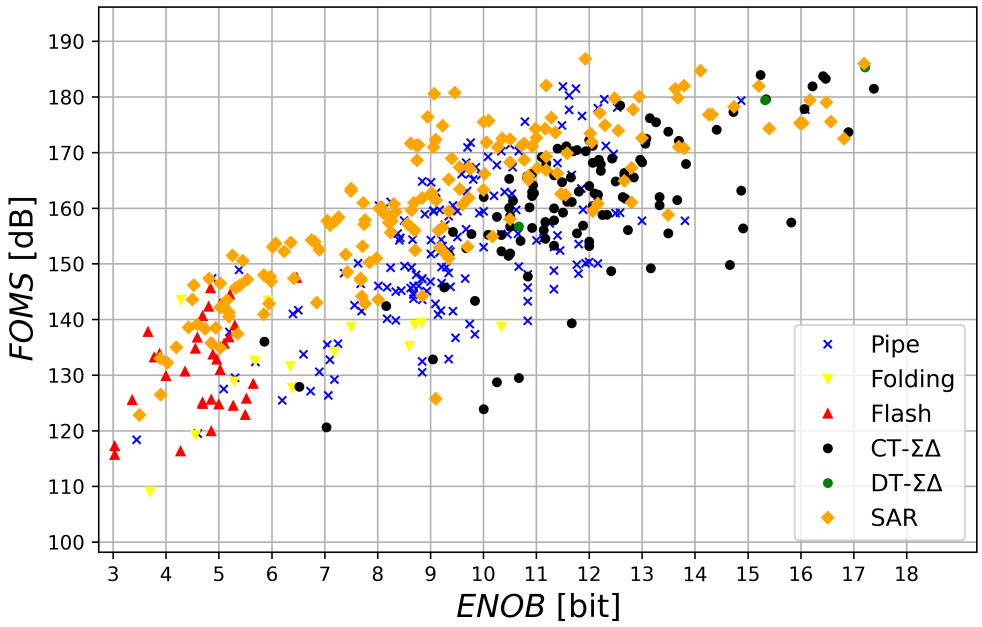

Figure 2.8: FOMS vs ENOB plot of the state-of-the-art ADCs obtained from [25]

3

Methodology

This chapter presents the methodology for designing and optimizing the 2-1 MASH ΣΔM presented in [2]. It will present the parameters used for the Genetic Algorithm (GA) to maximize the performance of this 2-1 MASH ΣΔM.

3.1 Initial Optimization Procedure Using Genetic Algorithm

The first step of the GA is to generate a population of circuits created by attributing random values to the parameters of the circuit's components R_1, C_1, Cf_1, R_2, C_2, Cf_2, R_{1A}, C_{1A}, and Cf_{1A}. The block diagram of the ΣΔ ADC is shown in Figure 3.1. This components can be observed in Figure A.1. Each component's value is limited to a range with a maximum and minimum corresponding to the acceptable variations for that particular element. For R the relation between R_{max} and R_{min} was around 5000, the For C the relation between C_{max} and C_{min} was around 100. This ranges were however adapted if any change was needed. These random values are generated with a linear distribution. Each chromosome is initially evaluated in 5 corners and the nominal case. This corners were from [2] and they didn't take into account the problems of the introduction of the calibration. Due to this they were quickly changed. After the evaluation of these chromosomes, specific vital parameters are extracted, such as the Output Swing (OS) of the two integrators of the 1^{st} stage of the 2^{nd} order ΣΔM, as it can be seen on Figure 2.4, and the SNDR of the MASH ΣΔM.

Afterwards, each chromosome is then attributed a score. This score is based on the SNDR of the nominal case, which was believed to be the peak SNDR of each chromosome and the minimum SNDR from the corners. It is worth noting that if a chromosome is unstable in any corner, it will score 0. This score of 0 is most frequent in the starting generations. It was also one way to optimize the code by just stopping to evaluate the chromosome. If the OS of the integrators is above 0.08V the score of the chromosome is reduced progressively. Limiting the OS in the circuit increases its sensitivity to thermal noise. This limitation is necessary because the High Level Model (HLM) of the ΣΔM does not include the non-linear behavior of the amplifiers, and this is a simple way of mitigating

the potential impact of gain block distortion introduced by the gain blocks present in the 2^{nd} order ΣΔM and between the 2^{nd} and 1^{st} stage, as illustrated by Figure A.1.

At the start of the optimization the total number of chromosomes is large in order to allow the GA to explore more of the design space. This requires a long time to evaluate the complete population. During the first five generations the number of chromosomes is reduced by half. This allows to obtain a compromise between exploring the complete design space and the convergence speed.

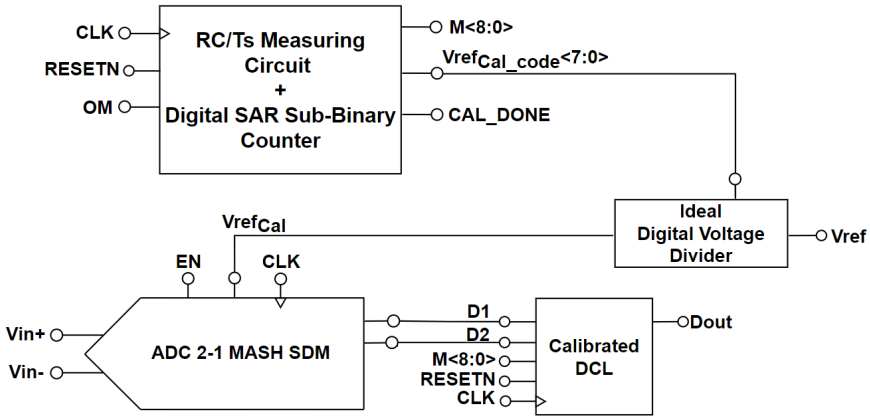

Figure 3.1: Block Diagram of MASH 2-1 ΣΔM with DCL and RC calibration circuit developed in [3]. Diagram from [3].

Upon assessing all the chromosomes, since we are using the elitism method, the 2 best chromosomes with the highest score is selected and retained for the subsequent generation. To populate the remainder of the generation, for each new chromosome, two chromosomes are chosen following their scores, favoring those with higher scores for a greater chance of selection, as per the rank method in GAs. These chosen chromosomes then undergo component inheritance from their respective parents, with a potential for a change in value called mutation. This degree of change is controlled, with the extent of potential alterations to a component being reduced every 40 generations, allowing the fine tuning of the chromosomes.

The chromosome with the highest score in each generation were saved. This was a simple way to evaluate the progression of each generation towards the optimal solution.

To calibrate the original circuit reference voltage, V_{ref}, had to be reduced. Consequently, all feedback capacitors, Cf, had to be increased so the feedback coefficient remains constant. The calibration system was a 3-bit system that varied V_{ref} linearly from 0.5V to 1.1V. Upon further inspection, it was noted that certain corners would fall below the nominal values due to V_{ref} fluctuations. A lower feedback coefficient is worse than a higher one due to the stability of the circuit. To address this issue, the minimum V_{ref} was increased from 0.5V to 0.55V. This adjustment enhanced the minimum feedback coefficient during a Monte Carlo (MC) analysis, which considered uniform random variations in circuit components.

3.2 Adapting the MATLAB® code of the GA

In order to adapt the MATLAB® code to include the calibration procedure, some parameters in GA had to be changed. Specifically, it was necessary to increase the number of corners utilized during optimization because of how the calibration works. Without calibration the ΣΔM has consistently bad performance in the same corners, using calibration however, overall improves their performance, but if not adjusted correctly, their performance may become subpar. The end result is that with calibration it is very difficult to predict which corner is the worst corner. Leaving any corner untested risked revealing that the unexamined corners did not meet the desired standards, potentially undermining the performance of otherwise promising chromosomes. Consequently, the need for comprehensive testing required an expansion in the number of corners examined to ensure the reliability of the GA. However, this expansion came at the price of significantly extending the time required to run the GA.

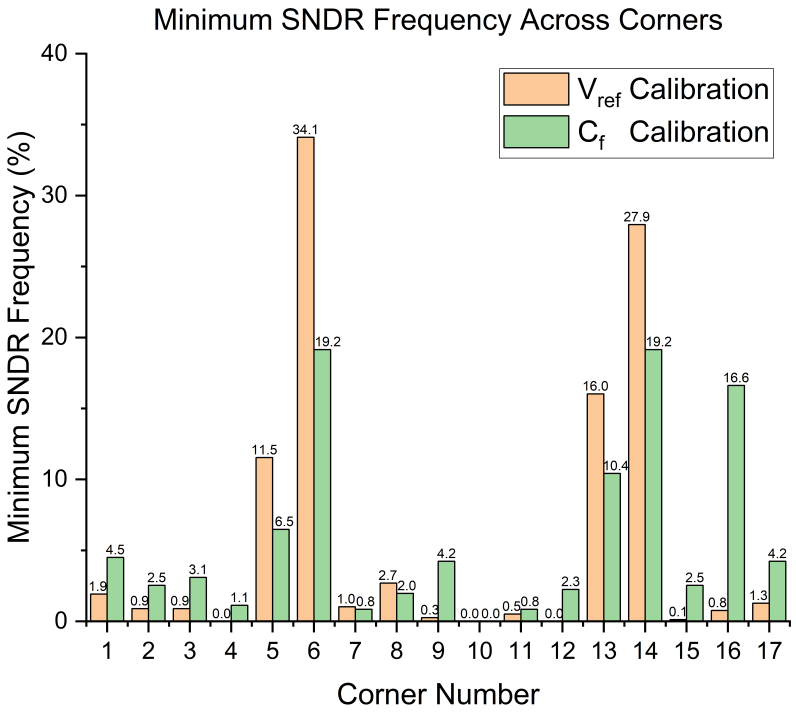

Figure 3.2: Worst Corner throughout each generation with the different calibrations. These corners are listed in Table 3.1

After an extended period of running the GA, a method was developed to gather all the previous best chromosomes of each generation and, taking account their calibration, analyze them and select the most frequent corner in which SNDR is minimum, relative to all the other corners utilized. The graph was produced through several GA simulations that were executed up to this point, evaluating all 17 corners. From this, it was possible to make a graph of all the most frequent minimum SNDR corners, as it is possible to see in Figure 3.2. After this, the code was changed to pick the first five most common worst

corners and the other two random corners. The number of random corners picked was increased by five every 50 generations till it reached the maximum, which, in this case, was 17. This method proved to be a great success in reducing the number of corners requiring evaluation, consequently improving the time spent with the GA, allowing for more chromosomes to run in each generation.

A different method to achieve enhanced results within a shorter time frame involves adopting a logarithmic distribution for the initial random population generation instead of a linear distribution. However, this approach proved unsuccessful because it tended to prioritize lower values, ultimately converging towards minimum values of C_1, C_2, and C_{1A}. This preference for lower values did not make sense in reducing circuit sensitivity to thermal noise since the circuit would be more sensitive, as mentioned in chapter 2. Nonetheless, it was noteworthy that with more generations, the genetic algorithm would correct itself when targeting higher SNDR.

The high-level calibration model presented in [3] was modified to replicate a finite set of Look Up Tables (LUT) employed in the DCL as it can be observed in 2.4. This adjustment aimed to achieve a more accurate alignment with CADENCE for analysis purposes.

Different types of calibration were also studied during the adaptation of the MATLAB® code. First, it was studied if reference voltage calibration of 4 bits was beneficial. Another type of calibration studied but not implemented due to time constraints was the feedback capacitor calibration instead of the reference calibration. Implementing this form of calibration is complex, yet it offers the potential to address certain challenges associated with calibrating V_{ref}. The calibration of V_{ref} has its own set of issues, including increased sensitivity to noise when lowering V_{ref}.

Another change was the fixing the values of R_2 and R_{1A} to the same values of the original circuit. Fixating these resistors may have compromised the maximum achievable performance. However, this change was made due to time constraints since the amplifier would need to be redesigned if R_2 and R_{1A} differed from the original values.

Another change involved fixing the values of R_2 and R_{1A} to match the original circuit's values. While fixing these resistors may have compromised the maximum achievable performance, this adjustment was made due to time constraints. Redesigning the amplifier would have been necessary if R_2 and R_{1A} differed from their original values.

In addition to using the GA code in MATLAB®, it is a similar code with various settings to evaluate individual chromosomes better. This code was utilized for conducting a Monte-Carlo analysis to study the re-evaluation of promising chromosomes obtained from the GA.

3.3 C++ code of the GA

A C++ code was developed mirroring the GA MATLAB® code to have a faster runtime. Furthermore, C++ can operate on any machine without the need for expensive software licenses. Since this code is much faster it allows the use of a significantly larger number

of corners and chromosomes per generation. The corners used for evaluation in C++ are the ones in Table 3.1. The most significant contrast between the two codes lies in C++'s utilization of parallel processing, a feature unavailable for me in MATLAB® at the time. Even without parallel processing, the C++ code ran 1.25 times faster than its MATLAB® counterpart. However, the improvement was limited due to the need to generate many random numbers. Various random number generators were explored, but ultimately, the *mt19937*, based on the Mersenne Twister algorithm, the default option in C++, had to be employed to meet the criteria of high-quality randomness for a large set of numbers. While a potential solution involving using a graphics card to expedite these results was considered, it could not be realized due to time constraints. The main libraries used for this code were the *Armadillo* package due to being easier to replicate MATLAB® functions and the *Sigpack* a library for signal processing. It is also worth noting that Open Multi-Processing, *OpenMP*, was also used.

Table 3.1: All corners considered for the GA C++ code

	$\frac{\Delta R}{R}(\%)$	$\frac{\Delta C}{C}(\%)$	$\frac{\Delta G}{G}(\%)$	$\frac{\Delta V_{ref}}{V_{ref}}(\%)$
Corner1	0	0	0	0
Corner2	+18	+16	+13	+5
Corner3	-18	+16	+13	+5
Corner4	+18	-16	+13	+5
Corner5	-18	-16	+13	+5
Corner6	+18	+16	-13	+5
Corner7	-18	+16	-13	+5
Corner8	+18	-16	-13	+5
Corner9	-18	-16	-13	+5
Corner10	+18	+16	+13	-5
Corner11	-18	+16	+13	-5
Corner12	+18	-16	+13	-5
Corner13	-18	-16	+13	-5
Corner14	+18	+16	-13	-5
Corner15	-18	+16	-13	-5
Corner16	+18	-16	-13	-5
Corner17	-18	-16	-13	-5

In GA C++ code, an enhancement was also made by adjusting feedback capacitors value around their respective integrating capacitors value to which they provided feedback. These range of values were determined through a study involving results obtained with GA MATLAB® code, revealing a consistent pattern. A random scaling factor, that falls within a specific range of values, is applied to the integrating capacitor to obtain the respective feedback capacitor. This change provides a better control over the feedback capacitors. Consequently, thanks to this fine-tuned feedback mechanism, chromosomes started yielding acceptable results more swiftly. However, the limits for feedback capacitors remained, keeping them within predefined maximum and minimum values. Should any

of these values fall outside the established thresholds, the function automatically corrected them, ensuring they remained within the desired range.

From the start, the main challenge was improving the circuit's performance in the worst case by a small margin of 1dB to 2 dB. The use of randomness caused SNDR to change unless we used the same random numbers, but that would lead to errors since a specific noise was being used, which, in turn, produced results which would not be accurate. A way to fix this was using a more time-consuming method: averaging the ΣΔM output. Instead of one output, the chromosome ran n_{avg} times, reducing variation by n_{avg} but increasing each chromosome's runtime by the same factor. This change was successfully proven and led to the current chromosome evaluation method using a $n_{avg} = 2$.

Because the current chromosome appeared highly sensitive to fluctuations in calibration values, a method was implemented to make it less susceptible to such variations. The method involves evaluating a chromosome without any calibration errors. Then, it is run twice: once with the maximum calibration error added and once with the maximum calibration error subtracted. Due to the implementation of the previous improvement, the calibration's randomness was removed to provide a set of more consistent results. This change was only possible since it was already being tested the insensibility to calibration values. This modification, however, is also more time consuming, making the code take three times as much. Although these results seem promising, due to time constraints, these results were obtained but not taken into account for this work.

The primary motivation behind rewriting this code stemmed from the inability to implement the Parallel Processing. In the context of GAs, Parallel Processing is a crucial tool that enables the generation of meaningful results within a shorter time frame.

Table 3.2: Speed Analysis: C++ and MATLAB® with Parallel Processing vs without Parallel Processing

Tool	w/o Parallel Processing	w/ Parallel Processing
MATLAB®	1x	————
C++	1.25x	12x

To assess the computational performance of the code in both C++ and MATLAB®, a total of 250 chromosomes, each with all 17 corners, were executed for each case. These 250 chromosomes were the resulting best chromosomes of a prior iteration of the GA done in C++. The necessity for a large number of chromosomes stems from the utilization of parallel processing, since to take full advantage of parallelization a high number of chromosomes needs to be evaluated. Subsequently, the resulting time of each case was compared, and the results are presented in Table 3.2.

Table 3.2 demonstrates that, even in the absence of parallel processing, the C++ code outperforms the MATLAB® code by 25%. This performance gap widens significantly when parallel processing is introduced, boosting the speed of the C++ code by nearly 10 times.

4

Simulation Results

This chapter presents the analysis of ΣΔM with calibration system developed in [3]. These analyses were vital for understanding the advantages of reference and feedback capacitor calibration. We wanted to determine if the additional effort put into feedback capacitor calibration would be worthwhile and estimate the potential benefits.

This chapter also presents the analysis comparing promising ΣΔM to the sizing obtained from GA optimization to [3].

4.1 Monte-Carlo Analysis of MATLAB® model

In order to ensure a high level of reliability in the analysis, it is necessary to run MC analysis with a substantial number of cases, more precisely 10,000 cases. For each case, the variation of capacitors, resistors, gain and reference voltage was randomly distributed uniformly. This approach allowed us explore worst-case scenarios and assess the performance of factors under examination, such as the chromosome or type of calibration.

For a more detailed analysis, histograms were made displaying the SNDR. Tables were also created to calculate the average, maximum, minimum, and standard deviation of the OS of the integrators in the ΣΔM, the SNDR, the feedback coefficients b_1, b_2, and b_{1A}, as well as the digital output of the ΣΔM, D_{MASH}, as shown in Figure 2.4.

These histograms were generated using the Excel add-in called *Analysis ToolPak*. The *Analysis Tools*, specifically the *Histogram* function, was utilized to create them. A Visual Basic for Applications, VBA, script was developed to expedite this process.

4.1.1 ΣΔM's feedback factors calibration using adjustment of value of feedback capacitors

Feedback capacitor calibration was implemented to address the shortcomings of reference calibration. This calibration is implemented with a capacitor array just like in Figure A.2, A.3, A.4. In the early stages of this work, on MATLAB® code, optimizations were found that with the same amount of generations and starting population, the feedback capacitor calibration would reach better results, as it can be seen in table A.13. For this reason,

feedback capacitor calibration has the potential for future research and could further improve this architecture's maximum performance.

This type of calibration was also tested on the ΣΔM presented in [3] and, when compared to the previous reference voltage calibration, it yielded a better performance.

Feedback capacitor calibration is akin to reference calibration because it takes the calibration value and uses it to adjust the feedback capacitor. This adjustment is achieved by either connecting or disconnecting a group of unitary capacitors as shown in Figure A.2, A.3, A.4. This system relies on binary weighting of the values of capacitors. The interval in which the feedback capacitor would need to vary for compensation of all corners was from 0.785 to 1.535 times of its original value. This interval was obtained by calculating the capacitor variation value required to keep the feedback coefficient constant. Notably, to calibrate the capacitor, the easiest method was to reduce the original feedback capacitor value by 21.5%. This would mean in Figure A.2, A.3, A.4 the λ would be $1 - 0.215 = 0.785$.

To determine the intervals, the feedback capacitor required for compensating the nominal feedback coefficient was calculated. This was done only in corners characterized by maximum RC variations, which determine the minimum and maximum calibration values.

Disregarding capacitor variations, the feedback capacitor's value necessary to compensate for the feedback coefficient relies solely on the changes in V_{ref} and R, as illustrated in (4.1). The integrating capacitor has no impact on the feedback coefficient.

$$1 + \frac{\overline{\Delta C_f}}{C_f} = \frac{1}{\left(1 + \frac{\Delta R}{R}\right) \cdot \left(1 + \frac{\Delta V_{ref}}{V_{ref}}\right)}, \tag{4.1}$$

where $\frac{\overline{\Delta C_f}}{C_f}$ is the change of feedback capacitor needed to compensate for a nominal feedback coefficient without considering capacitor variation.

The ideal feedback capacitor must, however, account for the capacitor variations. This is shown in (4.2).

$$1 + \frac{\overline{\overline{\Delta C_f}}}{C_f} = \frac{1 + \frac{\overline{\Delta C_f}}{C_f}}{\left(1 + \frac{\Delta C}{C}\right)} = \frac{1}{\left(1 + \frac{\Delta R}{R}\right) \cdot \left(1 + \frac{\Delta V_{ref}}{V_{ref}}\right) \cdot \left(1 + \frac{\Delta C}{C}\right)}, \tag{4.2}$$

where $\frac{\overline{\overline{\Delta C_f}}}{C_f}$ is the change of feedback capacitor needed to compensate for a nominal feedback coefficient considering capacitor variation, $1 + \frac{\Delta C}{C}$.

It's important to note that, for each RC variation, the maximum feedback capacitor variation determines the intervals.

Table 4.1 shows the selected variation was -23% instead of -30%, as opting for the latter would lead to an undercompensation of the feedback coefficient on corners where -23% variation was ideal. This is shown in Table A.14 Prioritizing overcompensation over undercompensation is preferred to prevent potential stability issues.

As for Table 4.2, for the same reasons previously stated, the feedback capacitor variation chosen was +53%, in alignment with Table A.15.

Referring to the tables 4.2 and 4.1, we can determine the intervals needed for capacitor value compensation, ranging from 0.77 to 1.53 of the original value. However, through experimentation with the ΣΔM presented in [3], it was discovered that increasing the lower limit by 0.15 and the upper limit by 0.05 in the previously mentioned range, would yield higher benefits.

Table 4.1: Maximum positive RC variation Corners

Feedback coefficient Calibration Comparison	Parameter Analysis	Corner2&6	Corner10&14
Uncalibrated Integrator	$\frac{\Delta R}{R}(\%)$	+18	+18
	$\frac{\Delta C}{C}(\%)$	+16	+16
	$\frac{\Delta V_{ref}}{V_{ref}}(\%)$	+5	-5
	$\frac{\Delta b}{b}(\%)$	+44	+30
Calibrated Integrator for Nominal Feedback Coefficient	$\overline{\frac{\Delta C_f}{C_f}}(\%)$	-19	-11
	$\overline{\overline{\frac{\Delta C_f}{C_f}}}(\%)$	-30	-23

Table 4.2: Maximum negative RC variation Corners

Feedback coefficient Calibration Comparison	Parameter Analysis	Corner5&9	Corner13&17
Uncalibrated Integrator	$\frac{\Delta R}{R}(\%)$	-18	-18
	$\frac{\Delta C}{C}(\%)$	-16	-16
	$\frac{\Delta V_{ref}}{V_{ref}}(\%)$	+5	-5
	$\frac{\Delta b}{b}(\%)$	-28	-35
Calibrated Integrator for Nominal Feedback Coefficient	$\overline{\frac{\Delta C_f}{C_f}}(\%)$	+16	+28
	$\overline{\overline{\frac{\Delta C_f}{C_f}}}(\%)$	+38	+53

The value of the feedback capacitor would determine the maximum number of bits the calibration system could have. If it was required to calibrate a feedback capacitor with 3 bits of resolution, it would mean that the system would had to have a capacitor seven times lower. Since the values of capacitors for C_{f2} and C_{f1A} were already low, it was impossible to implement a system of 3 bits since a value below 5fF was deemed too low to build as a unitary capacitor.

The major problem with this type of calibration is the variation of the capacitors. As a result of this, and seeing that the measuring circuit only measures RC, the same calibration value could need different feedback capacitor variations to compensate for nominal feedback coefficient. Since these feedback capacitor variations were not known to the measuring circuit, the approach of compensating for the worst case of each calibration

value was done. This approach allowed for the feedback coefficient do not fall below nominal values and, therefore, the system remained stable. To overcompensate as little as possible, all the possible variations for resistor and capacitor values was swept and the ideal number of calibrating capacitors calculated. These generated in the non-linear intervals for the feedback capacitor calibration. The values of SNDR were also omitted as its average values remained stable throughout the calibration analysis. The results obtained can be observed on tables 4.3 and 4.4 and where each number represents the number of bits for the feedback capacitor calibration used for integrator1, integrator2 and integrator1A respectively. Figures A.2, A.3, and A.4 illustrate the schematics corresponding to the different number of bits used on feedback calibration. One can conclude that the method was successful due to the absence of any cases in MC where an undercompensated feedback coefficient was observed. However, it is worth noting that integrators employing a 2-bit feedback capacitor calibration exhibited an average and maximum feedback coefficient considerably higher than the nominal value. This is attributed to the low resolution of 2-bit system, which, in certain scenarios, requires significant overcompensation to ensure that undercompensation in other cases never occurs. This enhancement however, becomes even more beneficial in the case of 3-bit and 4-bit systems, as the increased resolution allows for a value closer to nominal.

Table 4.3: Feedback coefficients derived from a Monte Carlo analysis involving 10000 cases, using feedback capacitor calibration with non-linear intervals.

C_f calib. 3 2 2 w/ non-linear intervals				C_f calib. 4 2 2 w/ non-linear intervals		
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}
Avg	1.13	1.20	1.20	1.10	1.20	1.20
Min	1.01	1.02	1.01	1.01	1.02	1.01
Max	1.26	1.45	1.46	1.19	1.45	1.46

Table 4.4: Feedback coefficients derived from a Monte Carlo analysis involving 10000 cases, using feedback capacitor calibration with linear intervals

C_f calib. 3 2 2 w/ linear intervals				C_f calib. 4 2 2 w/ linear intervals		
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}
Avg	1.18	1.18	1.18	1.18	1.18	1.18
Min	0.97	0.90	0.90	1.01	0.90	0.90
Max	1.31	1.39	1.39	1.29	1.39	1.39

Notably, a higher number of bits on this system would cause the overcompensation to be lower. This way, it was able to be obtained the minimum values of the capacitor that had to be turned on for the feedback coefficient not to fall below the nominal value, which would otherwise cause the integrators to saturate. This process was repeated when parasitic capacitances were introduced. This way, the intervals used to determine how much the feedback capacitor value increased were not linear.

4.1.2 ΣΔM's feedback factors calibration using reference voltage adjustment

Another possible solution that was studied in this thesis was the reference voltage calibration. Looking at the current corners, the minimum reference voltage needed for the feedback coefficient not to fall below nominal in most cases was 0.55 instead of 0.5V. This process is similar to the one previously done for feedback capacitor calibration. By altering the expression in (4.1) and (4.2) to what is calibrated.

$$1 + \overline{\frac{\Delta V_{ref}}{V_{ref}}} = \frac{1}{\left(1 + \frac{\Delta R}{R}\right) \cdot \left(1 + \frac{\Delta C}{C}\right)}, \tag{4.3}$$

where $\overline{\frac{\Delta V_{ref}}{V_{ref}}}$ is the change of reference voltage needed to compensate for a nominal feedback coefficient without considering their variation.

$$1 + \overline{\overline{\frac{\Delta V_{ref}}{V_{ref}}}} = \frac{1 + \overline{\frac{\Delta V_{ref}}{V_{ref}}}}{\left(1 + \frac{\Delta V_{ref}}{V_{ref}}\right)} = \frac{1}{\left(1 + \frac{\Delta R}{R}\right) \cdot \left(1 + \frac{\Delta V_{ref}}{V_{ref}}\right) \cdot \left(1 + \frac{\Delta C}{C}\right)}, \tag{4.4}$$

where $\overline{\overline{\frac{\Delta V_{ref}}{V_{ref}}}}$ is the change of reference voltage needed to compensate for a nominal feedback coefficient considering its variation, $\frac{\Delta V_{ref}}{V_{ref}}$.

As it is possible to observe the final expression (4.4) will be the same as (4.2). It is possible to conclude that the interval for the reference voltage used for calibration should be from 0.77 to 1.53 of the nominal V_{ref} value. Due to using a 1.1V supply voltage therefore, the V_{ref} adjustment factor has to be recalculated. Since the new V_{ref} nominal value decreases, in order to maintain the feedback coefficient constant, the feedback capacitors of the integrators should be increased by 1.53 to maintain the same value of feedback factors as before (2.11). This increase in feedback capacitors is not done in GA.

To calculate the range of values V_{ref} takes and knowing that the $MaximumV_{ref} = 1.1V$, the $MinimumV_{ref}$ is calculated through the following formula:

$$MinimumV_{ref} = \frac{MaximumV_{ref} \times (0.77)}{1.53} = 0.55V \tag{4.5}$$

With these changes, the new intervals would be the ones present in Table 4.5.

Table 4.5: Comparing the V_{ref} Intervals: Current vs. Proposed

	Code1	Code2	Code3	Code4	Code5	Code6	Code7	Code8
Original [3]	1.1	1.01	0.93	0.84	0.76	0.67	0.58	0.50
Proposed	1.1	1.02	0.94	0.86	0.79	0.71	0.63	0.55

Another aspect to consider was the number of bits used in the reference voltage calibration. It was noted that increasing the number of bits would lead to a higher minimum feedback coefficient, as seen in Table 4.6.

Table 4.6: Feedback coefficients derived from a Monte Carlo analysis involving 10000 cases, using reference voltage calibration comparing: Current vs. Proposed

	Original 3-bit Vref calibration			Proposed 3-bit Vref calibration		
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}
Average	1.11	1.11	1.11	1.13	1.13	1.13
Min	0.92	0.92	0.92	1.00	1.00	1.00
Max	1.26	1.26	1.26	1.27	1.27	1.27

This calibration using V_{ref} adjustment was utilized for ΣΔM optimization using GA and for schematic implementation. Due to time limitation the calibration based on adjusting the feedback capacitors was not implemented in the schematic.

4.1.3 Defining sizing of ΣΔM using feedback factors by reference voltage adjustment

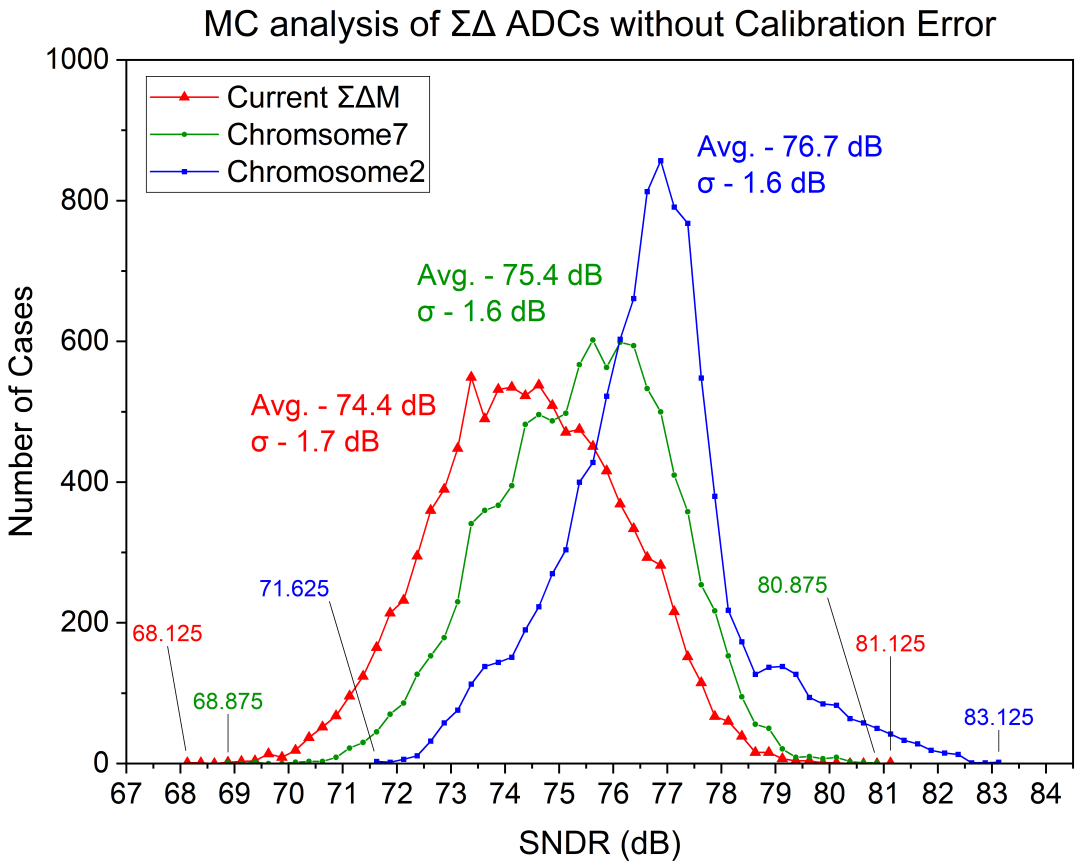

Figure 4.1: Monte Carlo Analysis of Current ΣΔM [3] vs Chromosome2 vs Chromosome7 with ideal calibration

A corner analysis using the 17 corners in Table 3.1 is done. This results can be observed on Tables A.16, A.17, A.18. After that, MC analysis of the chromosomes was carried out. The names Chromosome 2 and 7 are used solely for differentiation purposes and do not

carry any specific meaning or significance. The objective was to confirm whether these chromosomes indeed outperformed the current ΣΔM design. Various parameters were examined, including the D_{out} which is the D_{MASH} in Figure 2.4. If the D_{out} value exceeded 2^{14}, it signified the need for a higher number of bits in the DCL. Additionally, the OS of the integrators were also observed to ensure they did not exceed the initial limitations set during the GA optimization, ensuring the mitigation of non-linear effects as well as, for Integrator2 and Integrator1A, ascertain the stability of the circuit. The OS of Integrator1A is also checked since a higher minimum value would help the operations done with the comparator. The OS values were also essential for validating the schematic design in CADENCE.

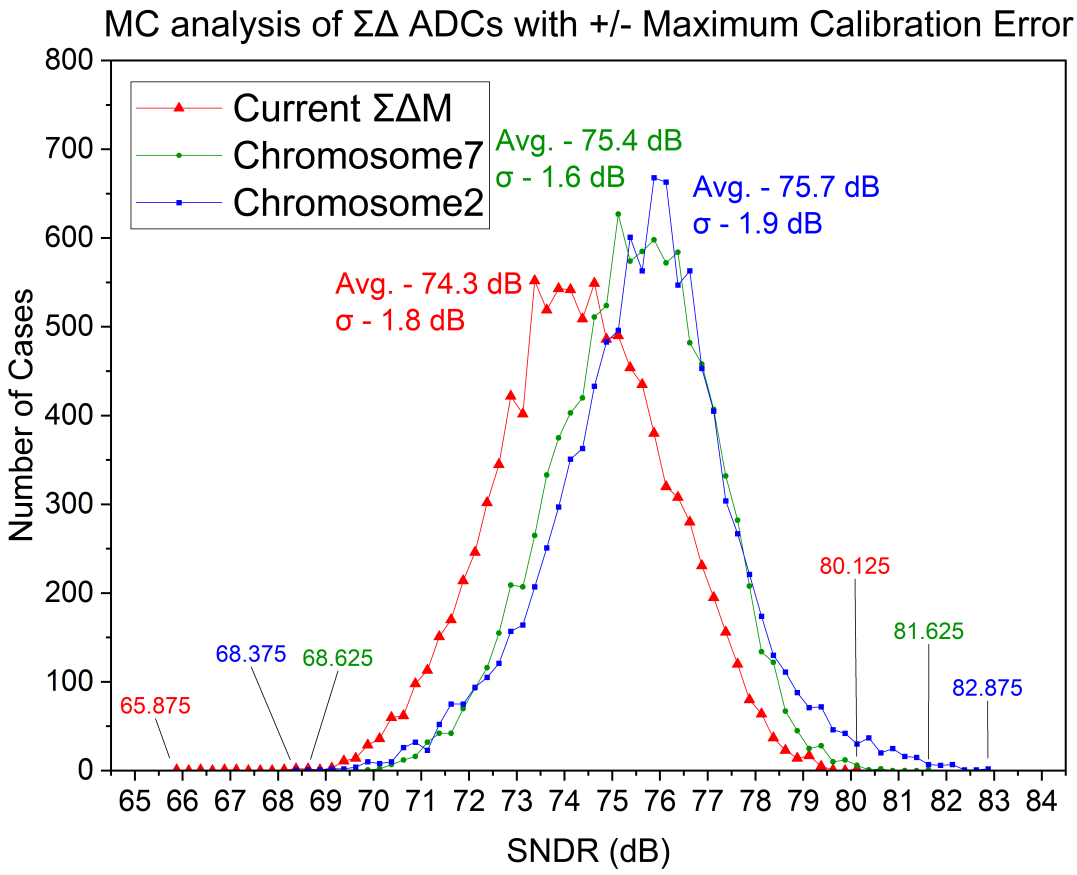

Figure 4.2: Monte Carlo Analysis of Current ΣΔM [3] vs Chromosome2 vs Chromosome7 while adding or subtracting maximum calibration error

Lastly, it was examined the SNDR with special attention to its minimum value and the standard deviation of the SNDR values. If a lower standard deviation was observed it would indicate that the ΣΔM architecture would be less sensitive to component variations or calibration values, depending on the specific test scenario.

Two main MC analyses were done: one with the proposed V_{ref} calibration method and no errors in the calibration values, allowing us to assess how well the ΣΔM performed with an ideal calibration, The second MC analysis incorporated the proposed V_{ref} calibration

but included the maximum possible errors in calibration values. To further confirm the latter, two other MC analyses were done where the maximum calibration value was added and other where the negative maximum calibration value was subtracted as it can be seen on Figure 4.3 and 4.4 respectively. All of this MC analyses were helpful to evaluate the behaviour of the chromosomes to calibration errors as well as variation of parameters.

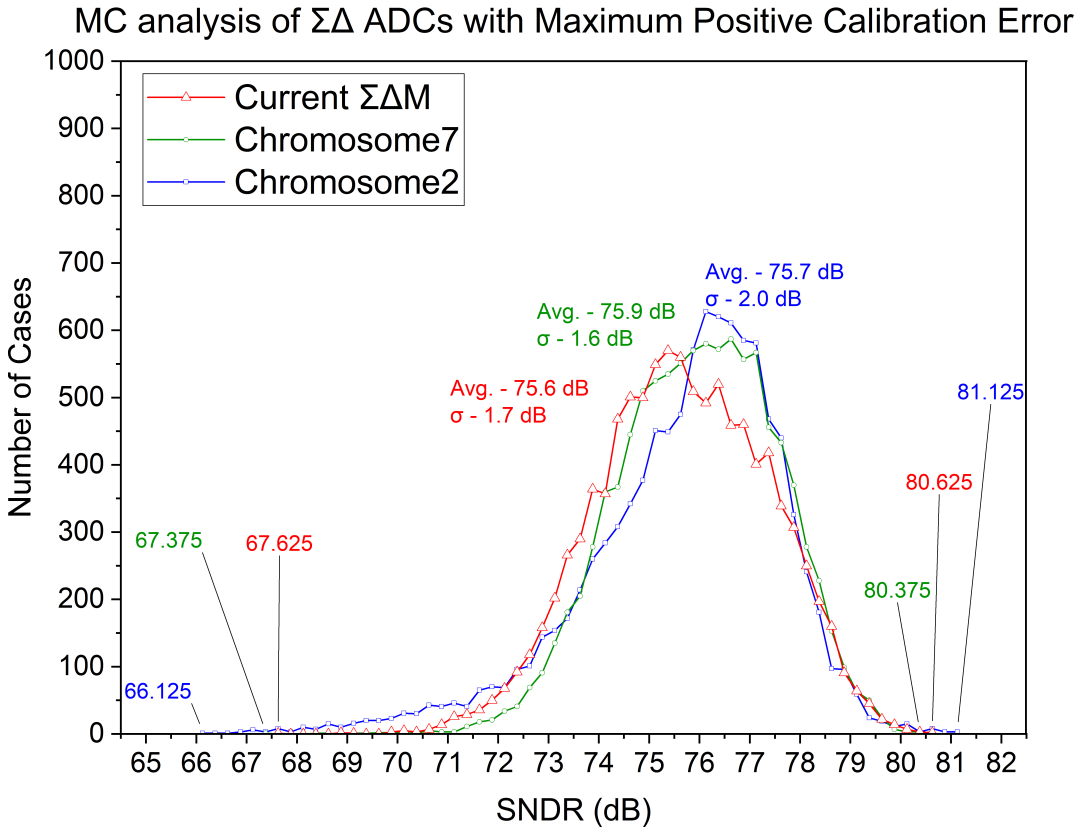

Figure 4.3: Monte Carlo Analysis of Current ΣΔM [3] vs Chromosome2 vs Chromosome7 while adding maximum calibration error

To compare the MC analyses between each other, considering the bins on the x-axis and the frequency on the y-axis, it is possible to draw a line that represents a histogram. This will aid us in comparing the possible solutions with each other.

Figure 4.1 presents a comparison of SNDR values between the current ΣΔM and two promising chromosomes, labeled Chromosome 2 and 7, selected for their performance, OSs, values of components, and other metrics. The comparison between them is made without considering calibration errors. For a more detailed breakdown of these comparisons between MC, please refer to Tables A.1, A.5, and A.9. To see individual histograms from Figure 4.1, please refer to Figure A.5, A.9, A.13.

From Figure 4.1 and Tables A.1, A.5, and A.9 it is possible to infer that, with an ideal calibration Chromosome2 has the best performance. It has an extremely high minimum SNDR, 71.6 dB, as well as a clear insensitivity to component variations as it can be confirmed by the standard deviation being 1.6 dB as shown in Table A.5 and by looking at

Figure 4.1. Chromosome7 also shows promise however it does not have the performance of Chromosome2. Both chromosomes are better than the current.

Figure 4.2 illustrates a comparison of the SNDR values between the current ΣΔM and the two chromosomes with the maximum calibration error. For a more detailed breakdown of these comparisons, please refer to Tables A.2, A.6, and A.10. To see individual histograms from Figure 4.2, please refer to Figure A.6, A.10, A.14.

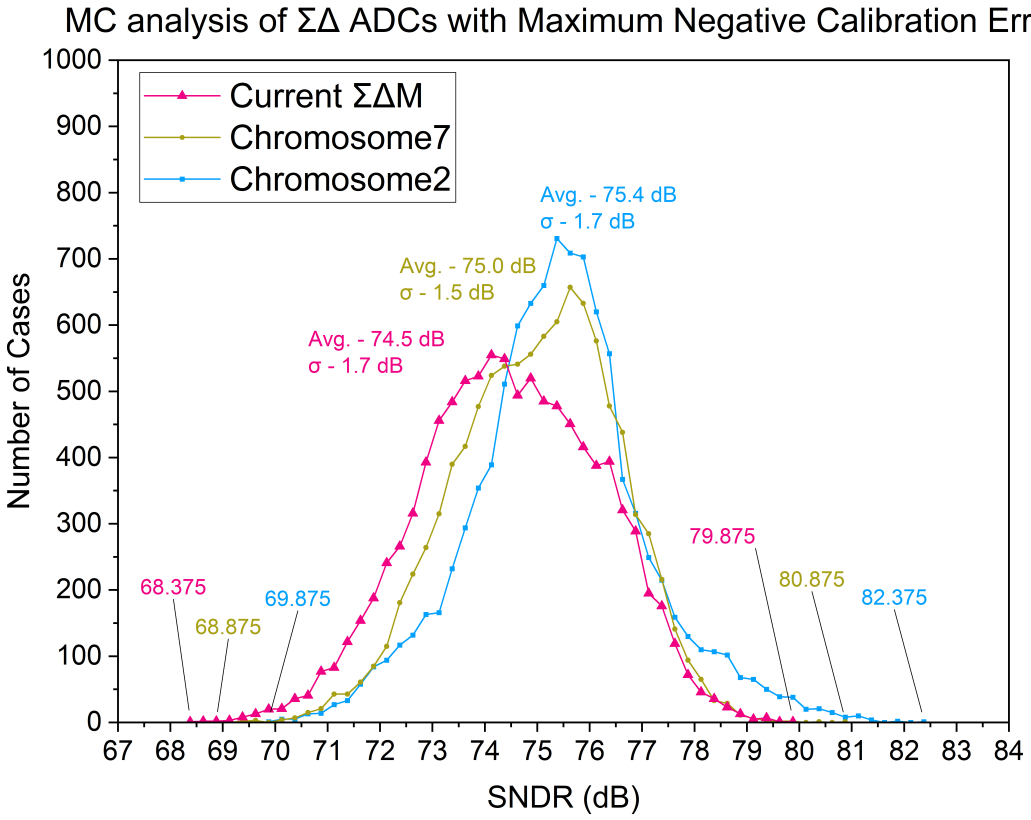

Figure 4.4: Monte Carlo Analysis of Current ΣΔM [3] vs Chromosome2 vs Chromosome7 while subtracting maximum calibration error

The Figure 4.2 shows interesting results. Chromosome2 shows a clear degradation of SNDR due to the calibration error. It also loses main property of being insensible to component variations. Chromosome7 on the other hand the performance is not degraded at all. Comparing Tables A.9 and A.10 it is possible to observe the average value of SNDR remaining constant. It seems Chromosome7 was not affected by the error in calibration at all. It is, however, worth reminding that the error being used is the maximum error and it just shows a worst case scenario not the real scenario.

From Figure 4.3 and 4.4 is possible to determine that adding the maximum positive calibration error is what achieves the worst possible results. The results show that the current ΣΔM is the most insensible to the maximum calibration error. As previously seen, and as expected, Chromosome2 achieves the worst results for the maximum positive calibration error, and when comparing both analysis it seems the main culprit of the loss

of insensitivity to component variations is due to the positive calibration error. It also seems the Chromosome7 is greatly affected by this error. For a more detailed analysis please refer to Tables A.3, A.7, A.12, A.4, A.8 and A.12. To see individual histograms from Figures 4.3 and 4.4, please refer to Figure A.7, A.8, A.11, A.12, A.15, A.16.

With all this taken into account, and although, the Chromosome2 is sensible to calibration error, the average SNDR value it offers is still superior.

4.2 Electrical simulation

The simulations presented were done for the Chromosome2 previously analyzed, using the tool *Virtuoso* from *CADENCE®*. The corners are different than the corners previously used on MATLAB® HLM. On these corners, resistors vary from -16% to +17%, the capacitors vary from -15% to +15%, the temperature from $-45°C$ to $125°C$, with a -5% to +5% supply voltage variation.

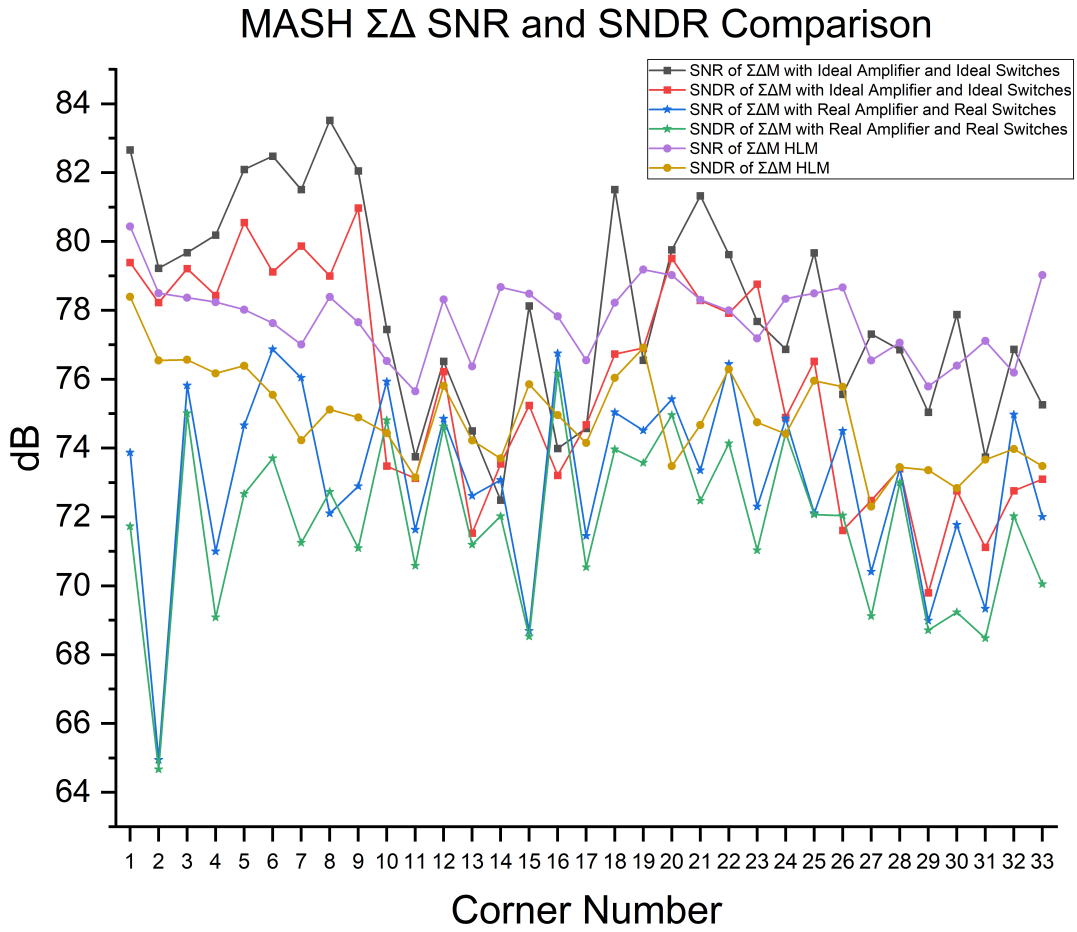

Figure 4.5: MASH SNR and SNDR comparison between the ΣΔM HLM, ΣΔM with real amplifiers and switches and a ΣΔM with ideal ones

To perform an electrical simulation analysis, simulations were done for a ΣΔM using

real amplifiers and real switches in one scenario, and ideal amplifiers and ideal switches in another. This approach was adopted to address discrepancies observed when comparing the HLM with ΣΔM real components. It's important to mention that in the latter, the capacitance values were reduced to account for the parasitic capacitances associated with the real switches.

In Figure 4.5, a notable contrast in SNDR is evident when comparing the ΣΔM with real components to the ΣΔM with ideal amplifiers and switches. This difference is thought to come from the switches and the amplifiers. The HLM can reasonably simulate the behavior of the ΣΔM with the ideal amplifiers and switches.

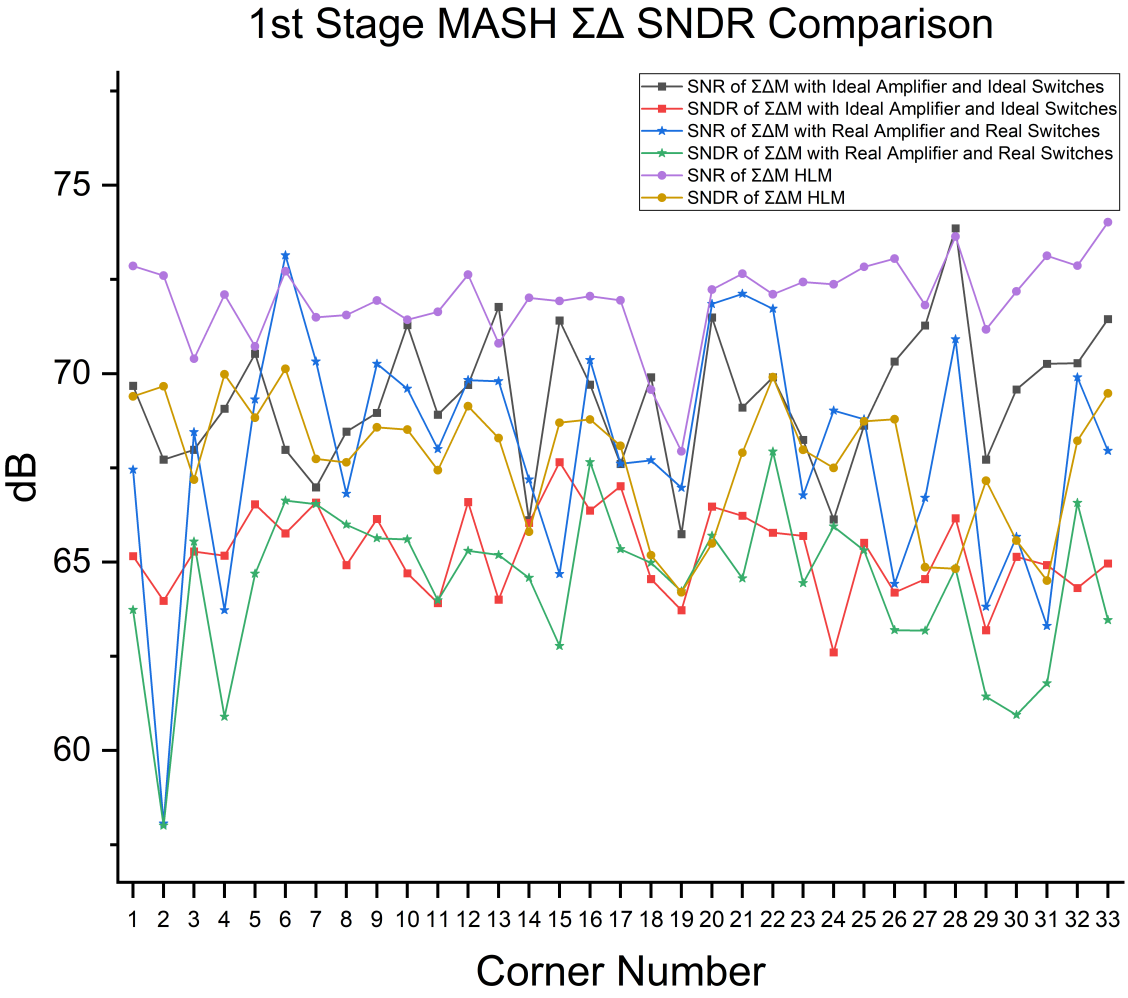

Figure 4.6: 1st Stage SNR and SNDR comparison between the ΣΔM HLM, ΣΔM with real amplifiers and switches and a ΣΔM with ideal ones

The Figure 4.6 shows that the HLM model has a higher SNDR when compared to the MASH ΣΔM SNDR in Figure 4.5. This shows that, even though the model seems accurate, when the DCL might not be as perfect as in electrical simulations or a more pessimistic approach is being taken in relation to noise. It also shows, that without taking into account certain problematic corners like for example 2, the ΣΔM with real amplifier

and real switches should be a lot more similar than ΣΔM with ideal ones. It seems the problem lies on the 2^{nd} stage and should be investigated further.

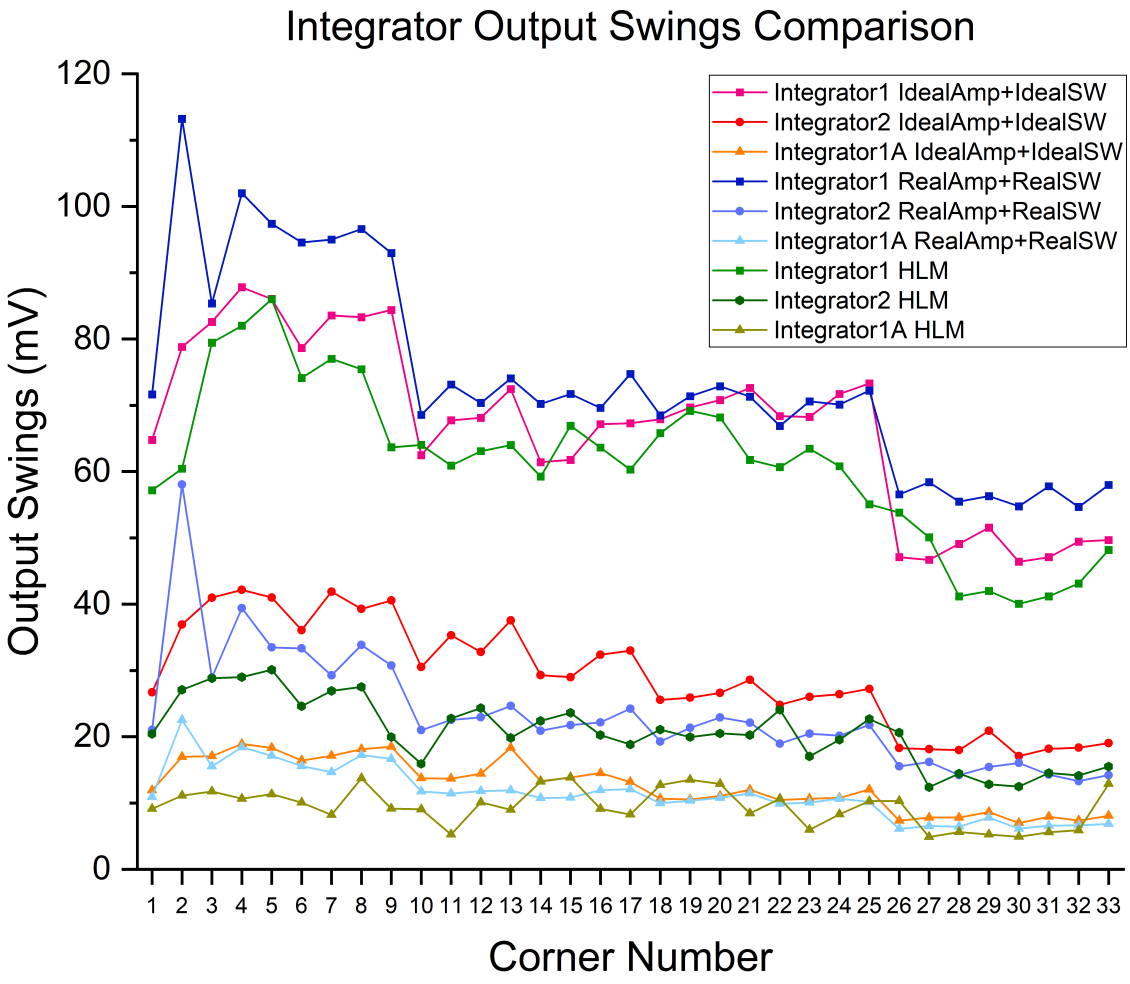

Figure 4.7: Output Swings comparison between the ΣΔM HLM, ΣΔM with real amplifiers and switches and a ΣΔM with ideal ones

The output swings on Figure 4.7 are crucial to verify if the implementation was successful. It is possible to observe that for the corner 2 there is a steep increase on the output swing. This could justify the loss of SNDR present in that corner on the $1st$ stage and consequently on the MASH ΣΔM. It is also confirmed the similarity of output swings of all integrators of the ΣΔM with ideal amplifiers and ideal switches.

In contrast to the current ΣΔM, the optimization did not yield apparent benefits, possibly due to two factors: first, the use of original switches in the integrators instead of the bootstrapped ones; and second, the impact of non-linear characteristics in the amplifiers on this ΣΔM which could have a more pronounced influence than on the current ΣΔM. To circumvent the latter, this non-linear behaviour could try to be modulated on HLM. Taking this into account, rectifying these issues and undertaking further optimization efforts could potentially improve the performance of the ΣΔM.

5

Conclusion and Future Work

This thesis is focused on optimizing R, C components values for the 2-1 MASH ΣΔM that uses passive integrators. This type of architecture is known to be sensitive to analog circuit imperfections, but with the use of GA, it was possible to mitigate it theoretically. In the high-level model, it was possible to obtain a ΣΔM with an improved performance to the ΣΔM with the calibration [3]. In this work, it is illustrated how it was possible to do so.

Chapter 3 provides a brief description of all the properties of the initial GA and all the steps needed to do the adaptation and optimize it. The biggest challenge of this thesis was the time constraint since GA takes a long time to produce meaningful results since a higher number of corners, generations and chromosomes had to be used, resulting in a constant search to improve the code's time efficiency. It is important to emphasize that while it was achieved a C++ code is slightly faster to MATLAB®, the C++ code has the potential for significantly faster execution depending on the technical expertise applied to its enhancement.

In Chapter 4, it is presented the resulting chromosomes from using GA optimization. The SNDR and other parameters are measured for each chromosome. Electrical simulations are also shown, and the analysis of a chosen winner chromosome is presented but certain corners still need to be improved upon. These electrical simulations also show some disparity between a ΣΔM with real components and the HLM however, it is also shown, that this could be attributed to the non-ideal characteristics of gain blocks.

This thesis shows that genetic algorithm is a powerful tool to optimize the performance of ΣΔM however it is a time consuming technique if wanting to optimize the results to the maximum. To minimize the time consumed, an efficient code should be implemented with a particular focus on the generation of random numbers, techniques as shown in this work should be implemented to reduce the optimization time and a hardware as fast as possible should be used to expedite this process.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] B. Nowacki. "Design of Sigma-Delta Modulators for Analog-to-Digital Conversion Intensively Using Passive Circuits". PhD thesis. Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, 2016. URL: http://hdl.handle.net/10362/18507 (cit. on pp. v, vi, 1, 5, 7, 8, 12, 34).

[3] I. F. C. N. de Brito. "Study of a Calibration Scheme for a Sigma-Delta Modulator Using Passive Integrators". Master's Thesis. NOVA University Lisbon, 2023 (cit. on pp. v, vi, 7, 8, 13, 15, 18–20, 22–26, 30, 39, 44, 45).

[4] C.-C. Lu and T.-S. Lee. "A 10-bit 60-MS/s Low-Power CMOS Pipelined Analog-to-Digital Converter". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 54.8 (2007), pp. 658–662. DOI: 10.1109/TCSII.2007.899449 (cit. on p. 1).

[5] S. Lee, A. P. Chandrakasan, and H.-S. Lee. "A 1 GS/s 10b 18.9 mW Time-Interleaved SAR ADC With Background Timing Skew Calibration". In: *IEEE Journal of Solid-State Circuits* 49.12 (2014), pp. 2846–2856. DOI: 10.1109/JSSC.2014.2362851 (cit. on p. 1).

[6] I. N. De Brito et al. "A Calibration Scheme for a Sigma-Delta Modulator Using Passive Integrators". In: *2023 18th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*. 2023, pp. 257–260. DOI: 10.1109/PRIME58259.2023.10161912 (cit. on p. 1).

[7] Y. Lim and M. P. Flynn. "A 1 mW 71.5 dB SNDR 50 MS/s 13 bit Fully Differential Ring Amplifier Based SAR-Assisted Pipeline ADC". In: *IEEE Journal of Solid-State Circuits* 50.12 (2015), pp. 2901–2911. DOI: 10.1109/JSSC.2015.2463094 (cit. on p. 3).

[8] C. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21. DOI: 10.1109/JRPROC.1949.232969 (cit. on p. 3).

[9] R. Schreier, G. C. Temes, et al. *Understanding delta-sigma data converters*. Vol. 74. IEEE press Piscataway, NJ, 2005 (cit. on pp. 4, 9).

[10] J. Wagner, P. Vogelmann, and M. Ortmanns. "On the Signal Filtering Property of CT Incremental Sigma–Delta ADCs". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 66.11 (2019), pp. 1780–1784. DOI: 10.1109/TCSII.2019.2891885 (cit. on p. 4).

[11] P. Benabes and R. Kielbasa. "Passive sigma-delta converter design". In: *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276)*. Vol. 1. 2002, 469–474 vol.1. DOI: 10.1109/IMTC.2002.1006887 (cit. on p. 4).

[12] P. Benabes, M. Keramat, and R. Kielbasa. "A methodology for designing continuous-time sigma-delta modulators". In: *Proceedings European Design and Test Conference. ED & TC 97*. 1997, pp. 46–50. DOI: 10.1109/EDTC.1997.582328 (cit. on p. 5).

[13] C.-H. Lin and M. Ismail. "Synthesis and analysis of high-order cascaded continuous-time ΣΔ modulators". In: *ICECS'99. Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357)*. Vol. 3. 1999, 1693–1696 vol.3. DOI: 10.1109/ICECS.1999.814501 (cit. on p. 5).

[14] B. Del Signore et al. "A monolithic 2-b delta-sigma A/D converter". In: *IEEE Journal of Solid-State Circuits* 25.6 (1990), pp. 1311–1317. DOI: 10.1109/4.62174 (cit. on p. 5).

[15] P. Heydari. "Characterizing the effects of the PLL jitter due to substrate noise in discrete-time delta-sigma modulators". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 52.6 (2005), pp. 1073–1085. DOI: 10.1109/TCSI.2005.849118 (cit. on p. 5).

[16] K. Sarangam and N. B. Rao. "A second-order Switched Capacitor Passive Sigma Delta Modulator with Bootstrapped switches in FinFET Technology". In: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. 2021, pp. 1–6. DOI: 10.1109/ICECCE52056.2021.9514234 (cit. on p. 5).

[17] J. L. A. de Melo, N. Paulino, and J. Goes. "Continuous-Time Delta-Sigma Modulators Based on Passive RC Integrators". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.11 (2018), pp. 3662–3674. DOI: 10.1109/TCSI.2018.2855649 (cit. on p. 6).

[18] A. Hussain et al. "Active–Passive ΔΣ Modulator for High-Resolution and Low-Power Applications". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.1 (2017), pp. 364–374. DOI: 10.1109/TVLSI.2016.2580712 (cit. on p. 6).

[19] R. Schreier and G. C. Temes. *Delta-Sigma Data Converters: Theory, Design, and Simulation*. IEEE Press, 1996 (cit. on p. 6).

[20] H. Wang et al. "A 0.59-mW 78.7-dB SNDR 2-MHz Bandwidth Active-RC Delta-Sigma Modulator With Relaxed and Reduced Amplifiers". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.3 (2021), pp. 1114–1122. DOI: 10.1109/TCSI.2020.3044075 (cit. on p. 6).

[21] D.-Y. Yoon, S. Ho, and H.-S. Lee. "15.1 An 85dB-DR 74.6dB-SNDR 50MHZ-BW CT MASH ΔΣ modulator in 28nm CMOS". In: *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*. 2015, pp. 1–3. DOI: 10.1109/ISSCC.2015.7063031 (cit. on p. 6).

[22] B. Nowacki, N. Paulino, and J. Goes. "15.3 A 1V 77dB-DR 72dB-SNDR 10MHz-BW 2-1 MASH CT ΔΣM". In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. 2016, pp. 274–275. DOI: 10.1109/ISSCC.2016.7418013 (cit. on p. 6).

[23] B. Nowacki, N. Paulino, and J. Goes. "A Third-Order MASH ΣΔ Modulator Using Passive Integrators". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.11 (2017), pp. 2871–2883. DOI: 10.1109/TCSI.2017.2704164 (cit. on pp. 7, 8).

[24] S. Rabii and B. A. Wooley. "A 1.8-V digital-audio sigma-delta modulator in 0.8-μm CMOS". In: *IEEE Journal of Solid-State Circuits* 32.6 (1997), pp. 783–796 (cit. on p. 9).

[25] B. Murmann. *ADC Performance Survey 1997-2023*. [Online]. Available: https://github.com/bmurmann/ADC-survey (cit. on pp. 10, 11).

[26] J. M. de la Rosa et al. "Next-Generation Delta-Sigma Converters: Trends and Perspectives". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 5.4 (2015), pp. 484–499. DOI: 10.1109/JETCAS.2015.2502164 (cit. on p. 10).

A

Appendixes

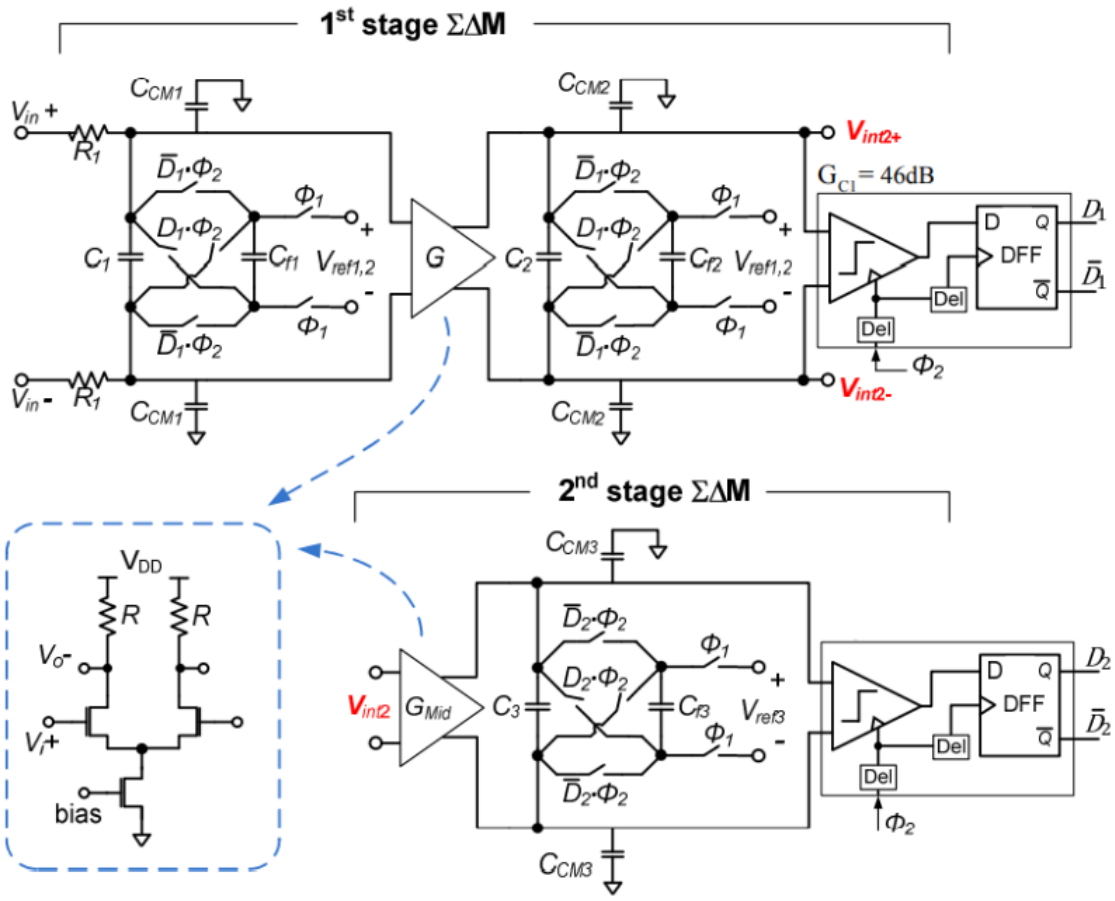

Figure A.1: A more detailed schematic of the MASH 2-1 ΣΔM, from [2]

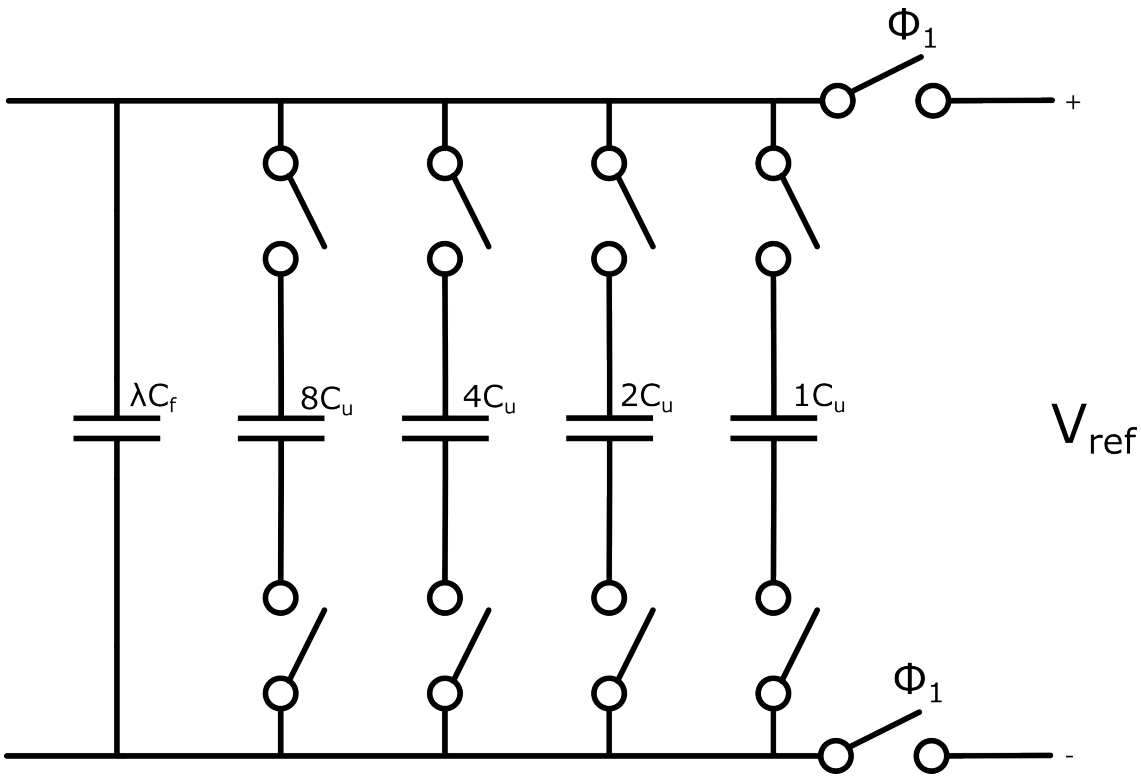

Figure A.2: Schematic of a 4-bit feedback capacitor calibration which would replace C_f in Figure 2.5. λ will be the minimum value of the feedback capacitor interval.

Table A.1: Performance metrics for current ΣΔM with V_{ref} calibration and without error in calibration values

Current ΣΔM w/ V_{ref} calibration w/o calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.17	1.17	1.17	5682	79.1	27.1	34.3	74.4
Min	0.89	0.88	0.89	2699	53.3	14.6	18.5	68.2
Max	1.32	1.32	1.33	10192	112.2	71.1	77.4	81.2
Std.Dev	0.068	0.068	0.068	1521	9.9	7.7	7.8	1.7

Table A.2: Performance metrics for current ΣΔM with V_{ref} calibration and with maximum error in calibration values

Current ΣΔM w/ V_{ref} calibration w/ calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	5711	78.8	27.0	34.2	74.3
Min	0.88	0.89	0.89	2207	52.5	14.8	18.5	65.9
Max	1.33	1.32	1.32	11018	112.5	71.4	73.1	80.2
Std.Dev	0.071	0.071	0.071	1583	10.0	7.4	7.7	1.8

Table A.3: Performance metrics for current ΣΔM with V_{ref} calibration and with maximum negative calibration errors

Current ΣΔM w/ V_{ref} calibration w/ Maximum Negative calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.20	1.20	1.20	5688	80.3	27.3	35.1	74.5
Min	0.94	0.94	0.94	2787	57.5	15.2	19.6	68.5
Max	1.35	1.35	1.35	10228	112.3	67.6	89.2	80.0
Std.Dev	0.057	0.057	0.057	1508	9.1	7.4	7.3	1.7

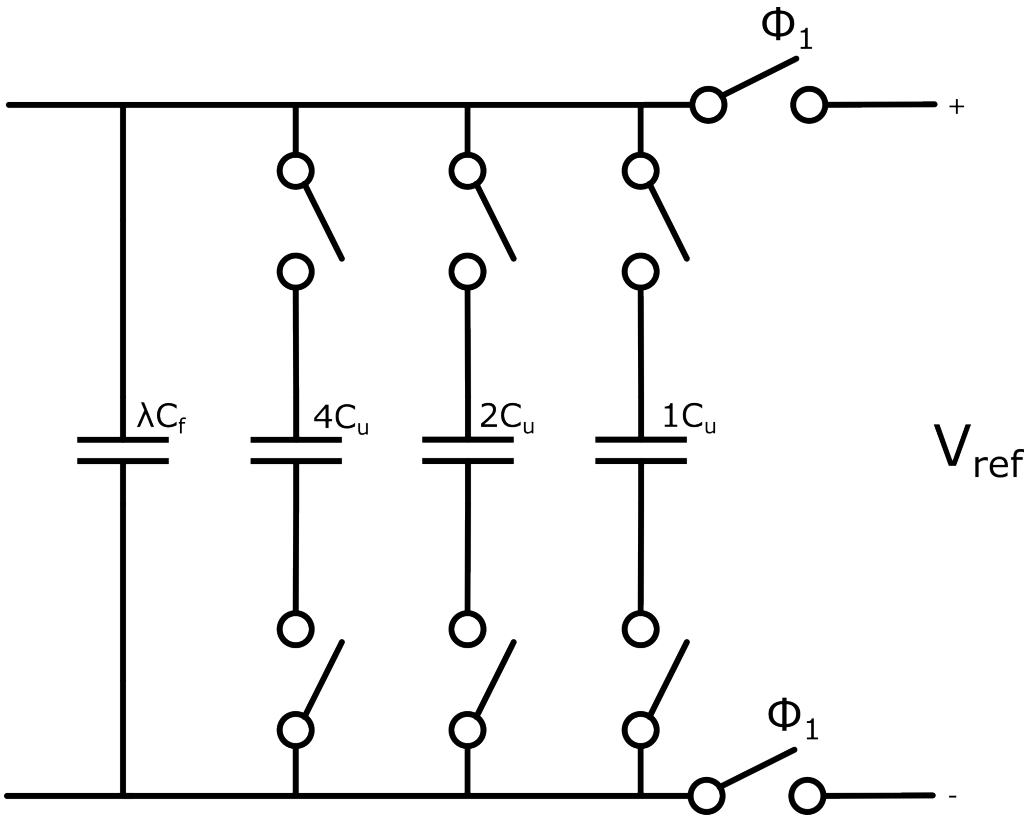

Figure A.3: Schematic of a 3-bit feedback capacitor calibration which would replace C_f in Figure 2.5. λ will be the minimum value of the feedback capacitor interval.

Table A.4: Performance metrics for current ΣΔM with V_{ref} calibration and with maximum positive calibration errors

Current ΣΔM w/ V_{ref} calibration w/ Maximum Postive calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.20	1.20	1.20	6548	75.9	26.4	32.9	75.6
Min	0.94	0.94	0.94	3421	55.6	15.0	19.8	67.6
Max	1.35	1.35	1.35	11209	106.8	65.7	90.2	80.7
Std.Dev	0.066	0.066	0.066	1618	8.5	6.8	7.0	1.7

Table A.5: Performance metrics for Chromosome2 with V_{ref} calibration and without error in calibration values

Chromosome2 w/ V_{ref} calibration w/ calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	5709	60.4	23.6	10.6	76.7
Min	0.94	0.94	0.93	2779	43.1	13.8	6.8	71.6
Max	1.30	1.30	1.30	10026	79.6	53.0	22.4	83.2
Std.Dev	0.056	0.056	0.056	1500	5.6	5.3	1.7	1.6

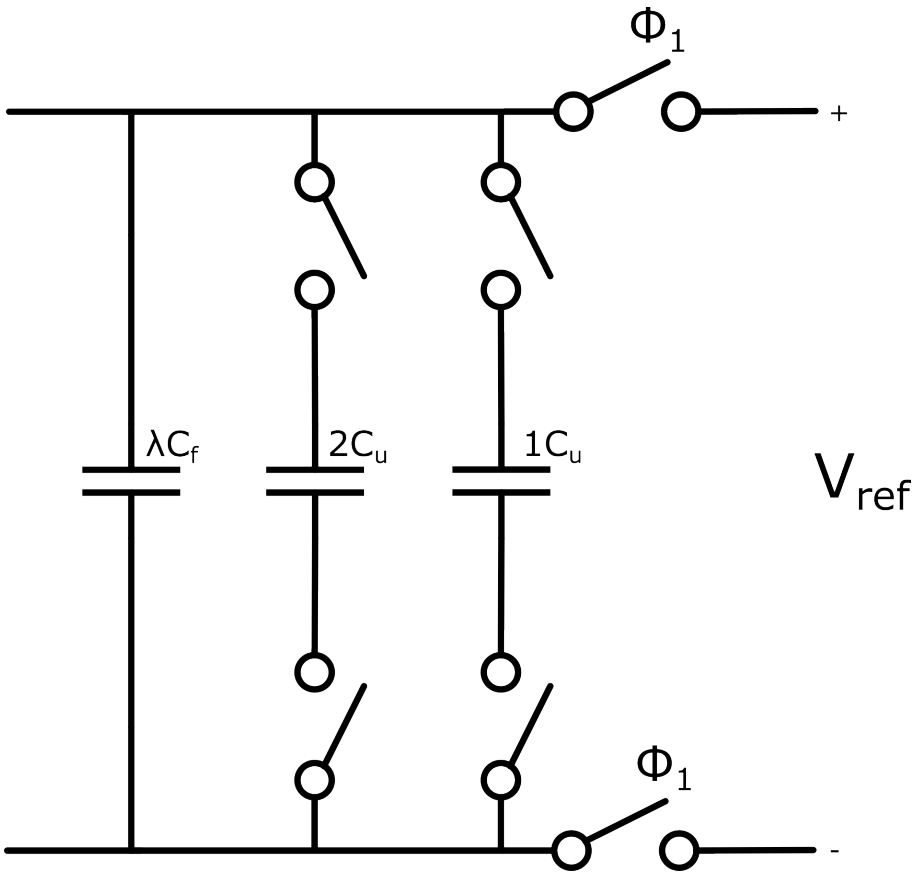

Figure A.4: Schematic of a 2-bit feedback capacitor calibration which would replace C_f in Figure 2.5. λ will be the minimum value of the feedback capacitor interval.

Table A.6: Performance metrics for Chromosome2 with V_{ref} calibration and with maximum error in calibration values

Chromosome2 w/ V_{ref} calibration w/ calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	5708	60.6	23.7	10.6	75.7
Min	0.94	0.94	0.94	2345	43.8	13.9	6.9	68.5
Max	1.30	1.30	1.31	11225	79.1	54.7	23.7	83.0
Std.Dev	0.057	0.057	0.057	1563	5.8	5.3	1.7	1.9

Table A.7: Performance metrics for Chromosome2 with V_{ref} calibration and with maximum negative calibration errors

Chromosome2 w/ V_{ref} calibration w/ Maximum Negative calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	5711	60.6	23.8	13.1	75.4
Min	0.95	0.95	0.95	2843	44.6	14.2	6.7	69.8
Max	1.30	1.30	1.30	10183	84.1	57.4	37.4	82.3
Std.Dev	0.055	0.055	0.055	1488	5.6	5.3	3.1	1.7

Table A.8: Performance metrics for Chromosome2 with V_{ref} calibration and with maximum positive calibration errors

Chromosome2 w/ V_{ref} calibration w/ Maximum Positive calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.15	1.15	1.15	6560	57.3	23.4	12.8	75.7
Min	0.95	0.95	0.95	3472	43.2	13.9	6.8	66.1
Max	1.30	1.30	1.30	11149	74.6	52.9	35.9	81.2
Std.Dev	0.064	0.064	0.064	1595	5.3	4.8	3.2	2.0

Table A.9: Performance metrics for Chromosome7 with V_{ref} calibration and without error in calibration values

Chromosome7 w/ V_{ref} calibration w/o calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	6621	57.4	23.8	12.9	75.4
Min	0.95	0.95	0.95	3185	41.5	13.9	8.2	68.8
Max	1.30	1.30	1.30	11934	78.5	54.5	24.8	80.9
Std.Dev	0.055	0.055	0.055	1762	5.6	5.7	2.1	1.6

Table A.10: Performance metrics for Chromosome7 with V_{ref} calibration and with maximum error in calibration values

Chromosome7 w/ V_{ref} calibration w/ calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	6648	57.4	23.8	12.9	75.4
Min	0.94	0.94	0.94	2697	41.2	14.2	8.3	68.6
Max	1.30	1.30	1.31	13171	76.2	57.0	26.5	81.6
Std.Dev	0.057	0.057	0.057	1842	5.8	5.6	2.1	1.6

Table A.11: Performance metrics for Chromosome7 with V_{ref} calibration and with maximum negative calibration errors

Chromosome7 w/ V_{ref} calibration w/ Maximum Negative calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.16	1.16	1.16	6651	57.3	23.9	14.8	75.0
Min	0.95	0.95	0.95	3278	41.7	13.7	8.1	68.9
Max	1.30	1.30	1.30	11934	77.4	59.5	38.2	81.0
Std.Dev	0.055	0.055	0.055	1754	5.6	5.5	2.9	1.5

Table A.12: Performance metrics for Chromosome7 with V_{ref} calibration and with maximum positive calibration errors

Chromosome7 w/ V_{ref} calibration w/ Maximum Positive calibration error								
	b_1/b_{1nom}	b_2/b_{2nom}	b_{1A}/b_{1Anom}	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	1.15	1.15	1.15	7652	54.3	23.2	14.3	75.9
Min	0.94	0.94	0.94	4013	40.0	14.1	8.2	67.3
Max	1.30	1.30	1.30	13081	71.9	54.9	29.7	80.4
Std.Dev	0.064	0.064	0.064	1884	5.3	5.0	2.9	1.6

Table A.13: Comparing the Top 5 Chromosomes sorted by Min. SNDR for each calibration

Calibration	Avg SNDR(dB)	Min. SNDR(dB)	Max. OS1(mV)	Max. OS2(mV)
C_f calib.	74.3	72.5	68.6	33.9
	74.9	72.2	76.0	47.2
	74.6	72.1	71.4	51.3
	74.4	72.0	69.7	58.5
	74.7	72.0	73.9	59.0
V_{ref} calib.	73.6	71.3	69.3	59.4
	74.0	71.3	58.9	51.0
	73.7	71.2	65.2	60.0
	73.5	71.2	72.0	45.9
	73.5	71.0	66.1	61.9

Table A.14: Relation between feedback coefficient with calculated feedback capacitor variations in Table 4.2

	Using C_f variation obtained from Corner5&9		Using C_f variation obtained from Corner13&17	
Parameter Analysis	Corner5&9	Corner13&17	Corner5&9	Corner13&17
$\frac{\Delta R}{R}(\%)$	-18	-18	-18	-18
$\frac{\Delta C_f}{C_f}(\%)$	+16	+16	+28	+28
$\frac{\Delta V_{ref}}{V_{ref}}(\%)$	+5	-5	+5	-5
$\frac{\Delta b}{b}(\%)$	0	-10	+11	0

Table A.15: Relation between feedback coefficient with calculated feedback capacitor variations in Table 4.1

	Using C_f variation obtained from Corner2&6		Using C_f variation obtained from Corner10&14	
Parameter Analysis	Corner2&6	Corner10&14	Corner2&6	Corner10&14
$\frac{\Delta R}{R}(\%)$	+18	+18	+18	+18
$\frac{\Delta C_f}{C_f}(\%)$	-19	-19	-11	-11
$\frac{\Delta V_{ref}}{V_{ref}}(\%)$	+5	-5	+5	-5
$\frac{\Delta b}{b}(\%)$	0	-10	+11	0

Table A.16: Analysis of Current ΣΔM [3] Using Corners from Table 3.1

	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	5841	75.7	27.1	34.7	75.2
Min	2699	49.6	12.4	16.7	70.8
Max	10192	108.8	68.7	68.7	81.3

Table A.17: Analysis of Chromosome2 Using Corners from Table 3.1

	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	5879	57.6	22.7	9.4	76.3
Min	2763	41.4	11.4	4.6	73.2
Max	10183	75.1	48.7	20.0	80.3

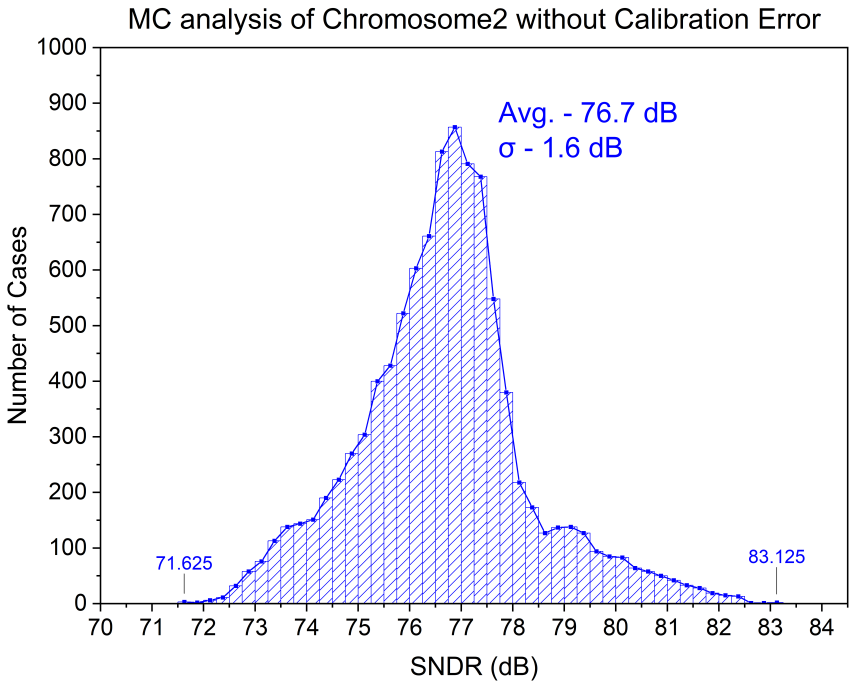

Figure A.5: Monte Carlo Analysis of Chromosome2 without calibration error

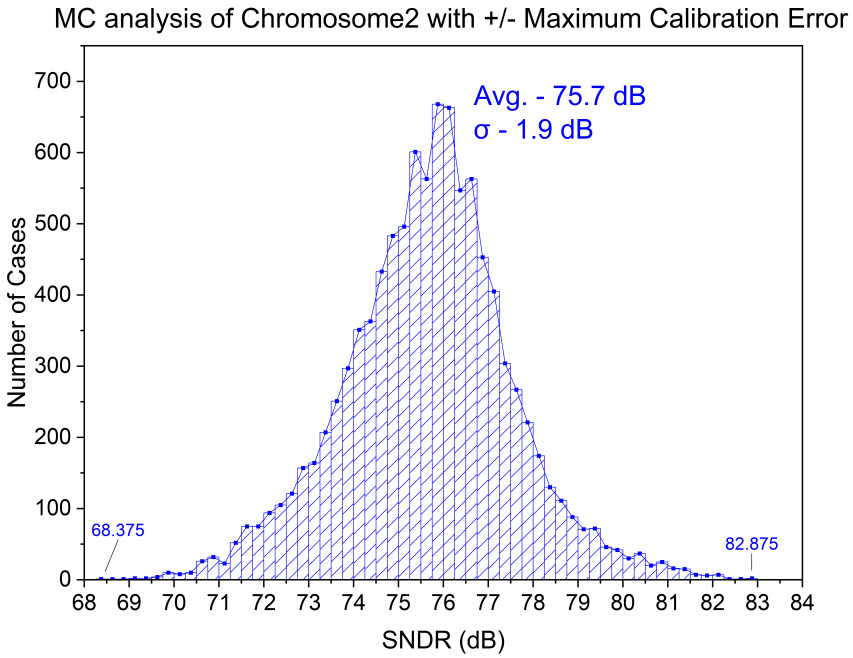

Figure A.6: Monte Carlo Analysis of Chromosome2 with maximum calibration error

Table A.18: Analysis of Chromosome7 Using Corners from Table 3.1

	Max D_{out}	OS1(mV)	OS2(mV)	OS1A(mV)	SNDR(dB)
Avg	6852	54.2	22.9	11.6	75.0
Min	3185	38.6	11.3	6.1	72.8
Max	11934	73.3	54.3	20.3	77.4

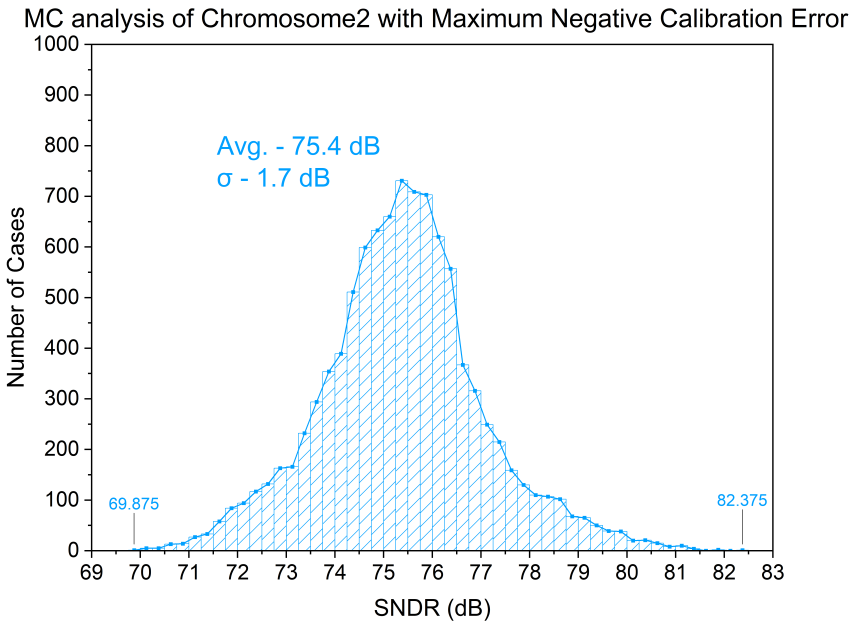

Figure A.7: Monte Carlo Analysis of Chromosome2 while subtracting maximum calibration error

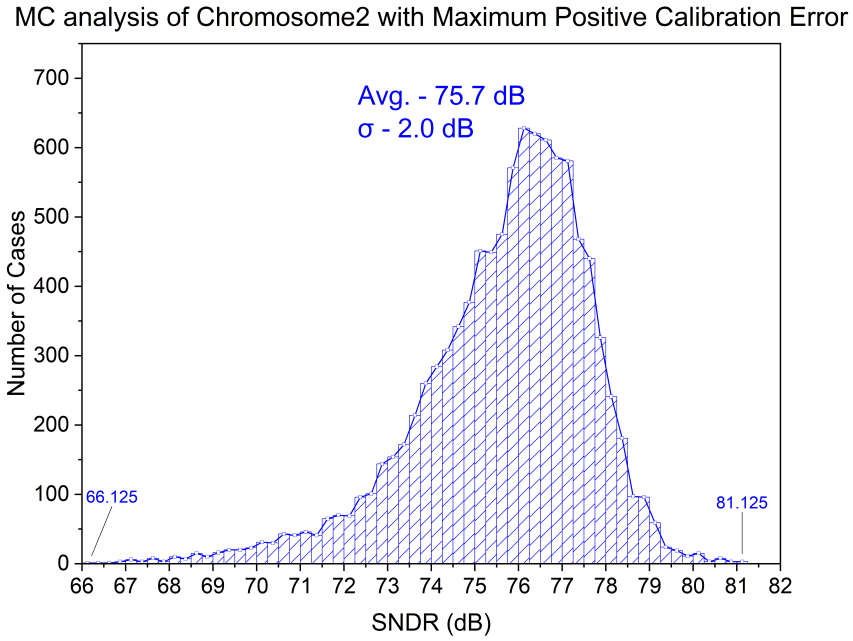

Figure A.8: Monte Carlo Analysis of Chromosome2 while adding maximum calibration error

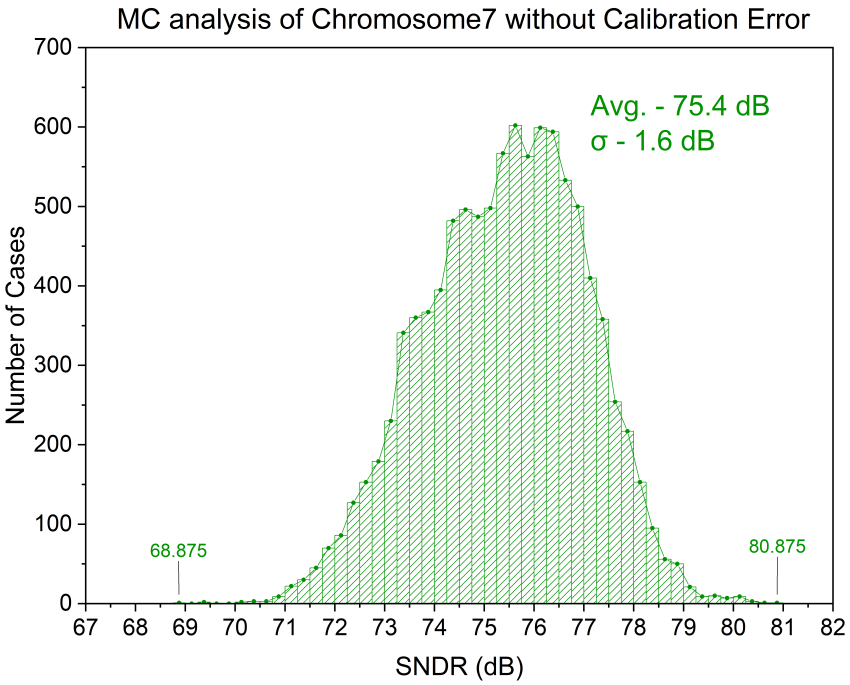

Figure A.9: Monte Carlo Analysis of Chromosome7 without calibration error

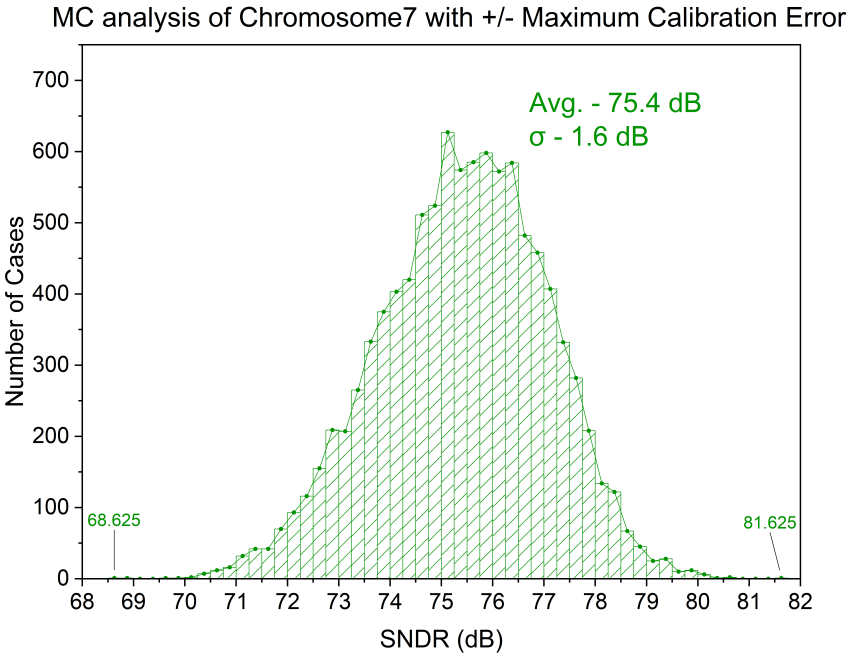

Figure A.10: Monte Carlo Analysis of Chromosome7 with maximum calibration error

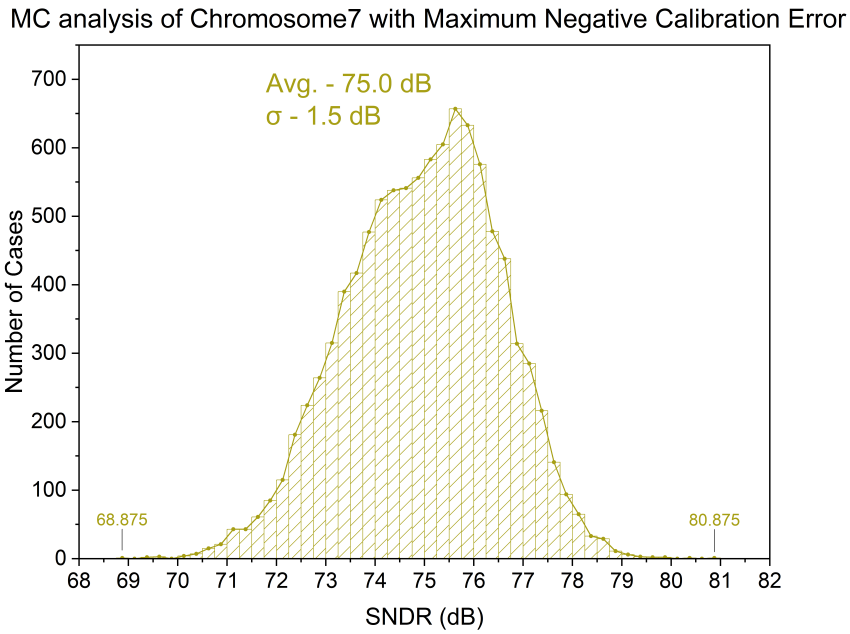

Figure A.11: Monte Carlo Analysis of Chromosome7 while subtracting maximum calibration error

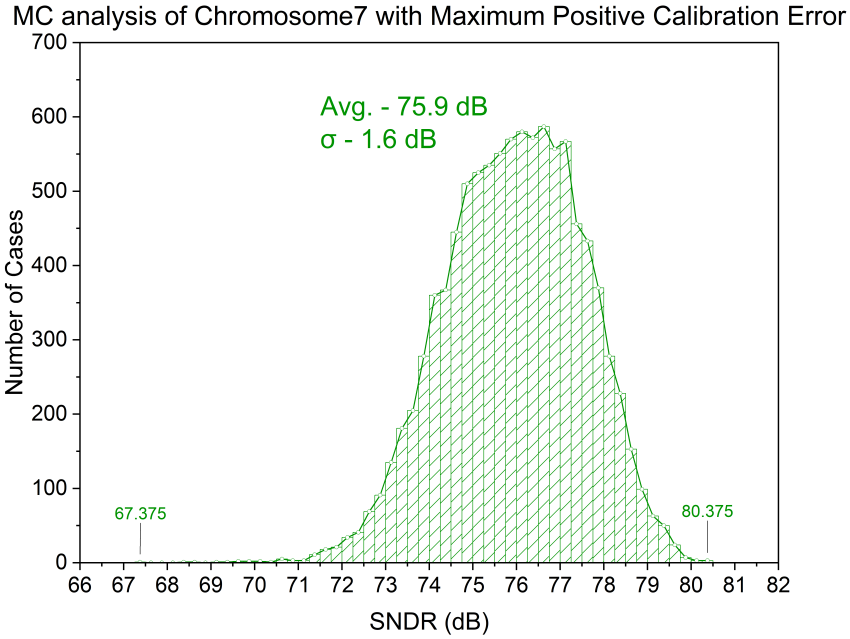

Figure A.12: Monte Carlo Analysis of Chromosome7 while adding maximum calibration error

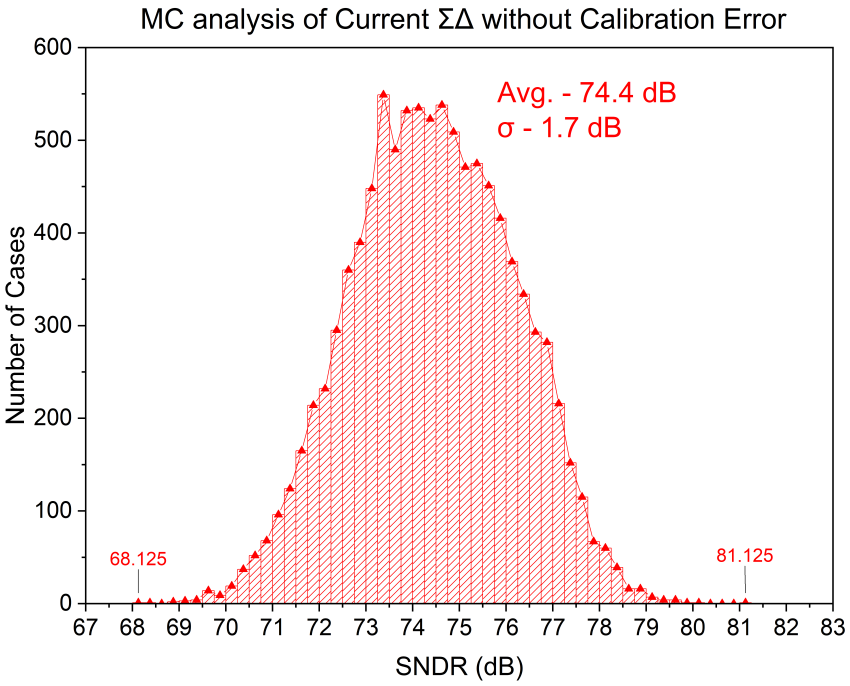

Figure A.13: Monte Carlo Analysis of Current ΣΔM [3] with no calibration error

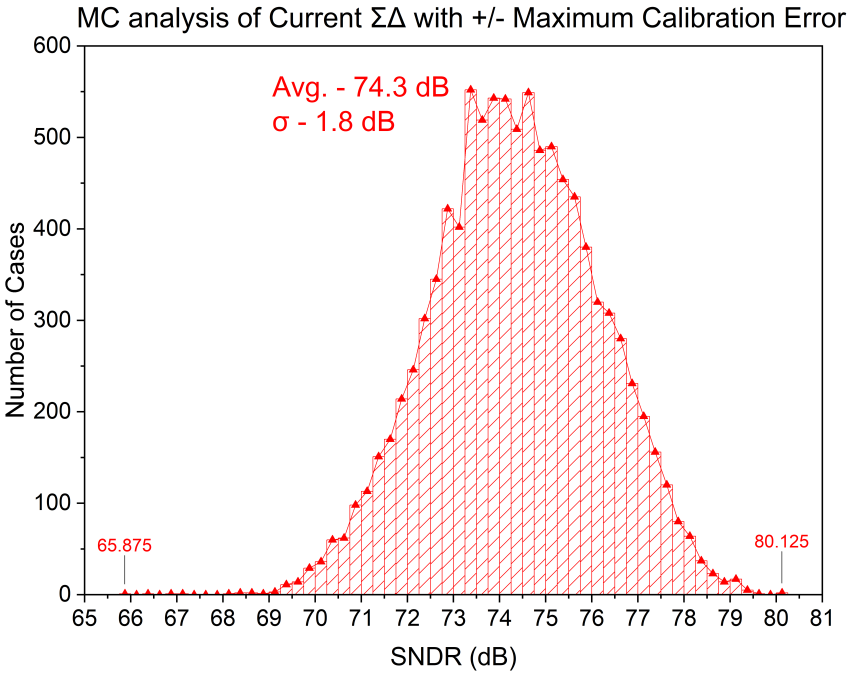

Figure A.14: Monte Carlo Analysis of Current ΣΔM [3] with maximum calibration error

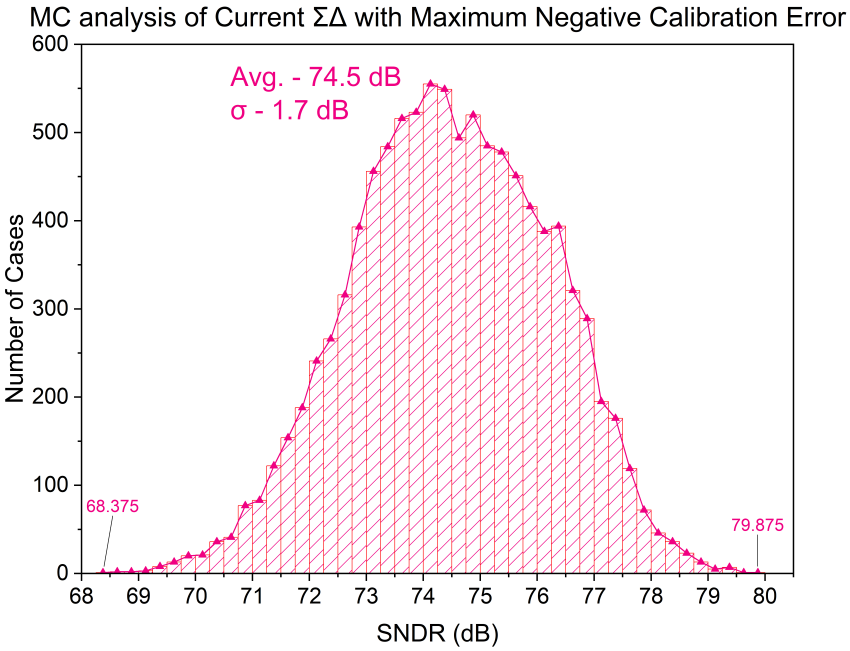

Figure A.15: Monte Carlo Analysis of Current ΣΔM [3] while subtracting maximum calibration error

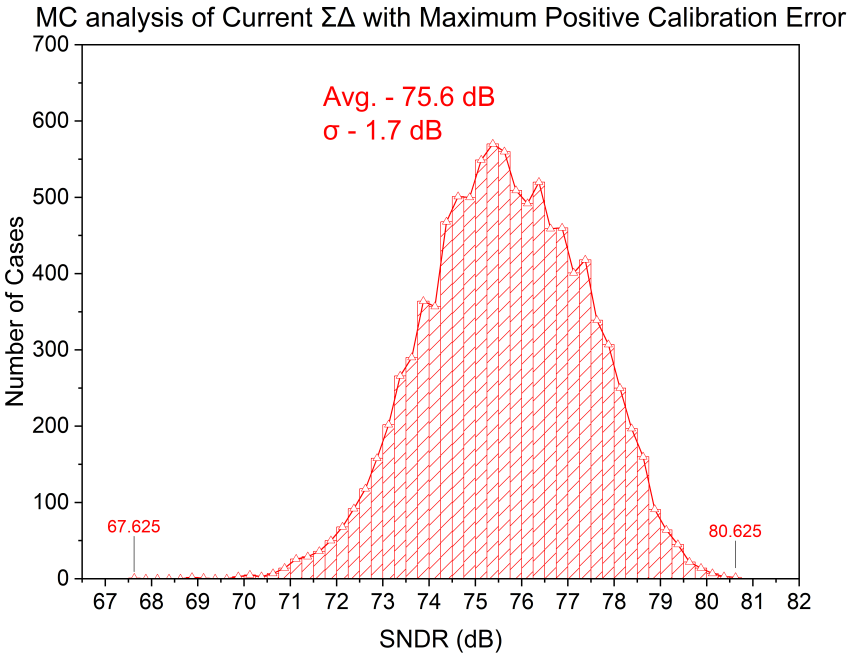

Figure A.16: Monte Carlo Analysis of Current ΣΔM [3] while adding maximum calibration error