NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATERIALS SCIENCE

RAFAEL FILIPE GOMES DE ALMEIDA LOUÇÃO

BsC in Micro and Nanotechnologies Engineering

BAND PASS NOISE-SHAPING DYNAMIC ELEMENT MATCHING FOR VCM BASED SAR-ASSISTED PIPELINE ADCS

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon

September, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
MATERIALS SCIENCE

BAND PASS NOISE-SHAPING DYNAMIC ELEMENT MATCHING FOR VCM BASED SAR-ASSISTED PIPELINE ADCS

RAFAEL FILIPE GOMES DE ALMEIDA LOUÇÃO

BsC in Micro and Nanotechnologies Engineering

Adviser: Nuno Filipe Silva Veríssimo Paulino
Associate Professor (with Habilitation), NOVA University Lisbon

Co-adviser: João Carlos Palma Goes
Full Professor, NOVA University Lisbon

MASTER IN MICRO AND NANOTECHNOLOGIES ENGINEERING

NOVA University Lisbon
September, 2023

Band Pass Noise-Shaping Dynamic Element Matching for VCM based SAR-Assisted Pipeline ADCs

Copyright © Rafael Filipe Gomes de Almeida Loução, NOVA School of Science and Technology, NOVA University Lisbon.
The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.0.2) [1].

To my mom and brothers

Acknowledgements

Firstly I want to acknowledge that without the partnership between the material science department and the electronic engineering department this project wouldn't have been possible. I am deeply grateful for all the remarkable efforts coming from professors Nuno Paulino, João Goes, Luís Oliveira and João Oliveira that truly helped shape the path I am currently treading. Thank you for all the amazing content procured and given in classes as well as the attentiveness shown whenever hardship rose.

I would like to give a special thank you to my advisers, professor Nuno Paulino and professor João Goes for the guidance given to me throughout the development of this project. I learned a great deal with them, both at academical and personal levels either from the endless email chains or the many zoom meetings, know that I truly appreciate every conversation.

I want to thank the very welcoming team of silicon engineers, from analog to layout to digital, that I interacted with during my internship at AMD, Ireland, especially Lim Peng and Vincent Callaghan who facilitated the integration. Even though the work developed in the internship was separate from that of this dissertation, all the soapboxes, intern meetings and overall gatherings really helped make Ireland a warmer place to write in.

My family above all takes an enormous part in this journey. For every soup that my mother made when I was sick, every late night pick-up after parties, every life lesson and long conversations to all the spoiled "mimo" that we shared, thank you mom I love you, you are and will always be a great part of my heart. I appreciate all the "gooferies" that me and my brothers always had and will continue to have for more years to come. Thank you Ricardo for all the lessons that someday I hope to pass on to my kids. Thank you Rodrigo for coming to my room to do nothing but mess with each other and talk about mindless things. Thank you so much for making home feel like home.

My dearest friends: João, you are one of the most inspirational people I have ever met, thank you for allowing me to learn so much from you every day, the Happiness Enjoyment and fun You bring whenever you are around is out of this world. Ana, thank you for being my partner in crime for all these years and the next to come, you are truly one of a kind. Thank you André for being there allowing me to vent off about most of my difficult moments, giving me advice when I needed it. Thank you Eva for putting up with me every day on end while strolling to Lidl to get lunch.

Thank you all.

”

“Non nobis solum nati sumus.
(Not for ourselves alone are we born)”

— Marcus Tullius Cicero, De Officiis
(Statesman, lawyer, writer, orator, philospher)

Abstract

As the drive for more powerful and capable Analog to Digital Converters (ADCs) increases, so does the need for error mitigation techniques. Components' mismatch significantly degrades the performance of ADCs by introducing non-linearities that cause high energy spurs in its output spectra. Dynamic Element Matching (DEM) techniques allow for the mitigation and filtering of mismatch error's spectral influence, known as noise-shaping. Although these techniques are most commonly used in $\Sigma\Delta$ ADCs, given that these operate with relatively high Oversampling Ratios (OSRs) and in the low frequency regime, this work will extend a Data Weight Averaging (DWA) DEM technique to the Successive Approximation Register (SAR) architecture enhanced by the pipeline topology for a band-pass operation and a low Oversampling Ratio (OSR). This extension allows for the utilization of filter transfer functions that are embedded in the DEM technique as general as $H(z) = 1 \pm Z^{-a}$ affecting only the mismatch error. The DEM method, accompanied by an output digital filter, results in the significant increase of the Signal to Noise and Distortion Ratio (SNDR) as well as the Spurious Free Dynamic Range (SFDR) of the converter. This work provides the mathematical demonstration and conceptual explanation, the methods to analyze, and the digital implementation required to realize the referred DEM technique.

Keywords: Mismatch, DEM, DWA, SAR, Noise-Shaping, Band-Pass, OSR, SNDR, SFDR

Resumo

À medida que a procura para ADCs mais precisos e capazes aumenta, a necessidade para técnicas de mitigação de erros é maior. A discrepância entre componentes num conversor, (*mismatch*), deteriora o desempenho significativamente ao introduzir não-linearidades que causam espúrias de alta energia no espetro de saída. Técnicas de DEM permitem a atenuação ou filtragem dos efeitos de *mismatch* no espetro de saída dos ADCs referida como *noise-shaping*. Apesar deste tipo de técnicas ser comumente utilizadas em conversores $\Sigma\Delta$, porque DEM beneficia das altas taxas de amostragem e operações a baixas frequências destes conversores, o trabalho apresentado estende uma técnica DEM conhecida como DWA a ADCs de topologia SAR assistida por uma estrutura *pipeline* que possibilitará uma operação de DEM com baixos OSRs e em modos de operação passa-banda. Esta extensão permite a incorporação de filtros gerados pelo algoritmo de DEM de forma geral $H(z) = 1 \pm Z^{-a}$. Esta implementação acompanhada por um filtro digital resulta na melhoria apreciável da SNDR bem como da SFDR do conversor. Este trabalho desenvolve a demonstração matemática para a implementação de DEM, realiza uma explicação conceptual para o fenómeno, os tipos de métodos para análise bem como a implementação digital que é capaz de gerar este algoritmo.

Palavras-chave: Mismatch, DEM, DWA, SAR, Noise-Shaping, Passa-Banda, OSR, SNDR, SFDR

Contents

List of Figures

List of Tables

Acronyms

ADC Analog to Digital Converter *(pp. ix–xi, xv–xvii, 1–6, 8–19, 22–26, 28–30, 35)*
ADCs Analog to Digital Converters *(pp. v, vi)*

C-DAC Capacitive Digital to Analog Converter *(pp. ix, x, xii, xv–xvii, 1, 3, 7, 9–15, 19–21, 24–26, 28–30, 34, 37, 39–41, 43)*

DAC Digital to Analog Converter *(p. 18)*
DEM Dynamic Element Matching *(pp. v, vi, 2, 4, 9, 11, 12, 18, 21, 29, 30)*
DWA Data Weight Averaging *(pp. v, vi, 18, 30)*

ENOB Effective Number of Bits *(pp. 1, 5, 8–12, 14, 16, 30)*

FFT Fast Fourier Transform *(p. 4)*

LSB Least Most Significant Bit *(pp. xv, 7, 14, 15, 35)*

MC Monte-Carlo *(pp. ix–xi, 5, 6, 8–12, 14, 16, 17, 25, 26, 28, 29)*
MSB Most Significant Bit *(pp. ix–xii, xv, 4, 7, 13–17, 19, 34, 36–38)*
MSB-1 Second Most Significant Bit *(pp. 7, 13, 15, 20)*
MSB-2 Third Most Significant Bit *(p. 13)*
MSB-j $j+1^{th}$ Most Significant Bit *(p. xv)*

NP-Hard Non-deterministic Polinomia-time Hard *(p. 28)*

OSR Oversampling Ratio *(pp. v, vi, 29, 30)*
OSRs Oversampling Ratios *(pp. v, vi)*

SAR Successive Approximation Register *(pp. v, vi, ix, x, xii, xvi, xvii, 1–5, 8, 18, 19, 29, 30, 35–37)*
SFDR Spurious Free Dynamic Range *(pp. v, vi, 2, 5, 8–12, 14, 16, 17, 25, 26, 29, 30)*
SNDR Signal to Noise and Distortion Ratio *(pp. v, vi, ix, x, 2, 5, 6, 8–12, 14, 16, 17, 25, 26, 28, 29)*

VCM Common Mode Voltage *(pp. ix, 2, 3)*

Symbols

a	the multiplicative factor in frequency contraction *(pp. v, vi, 22–24, 26, 28, 29)*
α	the level of MSB permutation *(p. 15)*
AME	the accumulated mismatch error defined in appendix D *(pp. 21, 23)*
$AME_{gnd\,p/n}$	the accumulated mismatch error from capacitors connected to gnd *(p. 42)*
$AME_{V_{Ref}\,p/n}$	the accumulated mismatch error from capacitors connected to V_{Ref} *(p. 42)*
ASC	the accumulated selected capacitors defined in appendix D *(pp. 23, 47)*
$ASC_{gnd\,p/n}$	the accumulated selected capacitors connected to gnd *(pp. 40, 42)*
$ASC_{VCM\,p/n}$	the accumulated selected capacitors disconnected from VCM *(pp. 40, 42)*
$ASC_{V_{Ref}\,p/n}$	the accumulated selected capacitors connected to V_{Ref} *(pp. 40, 41)*
$bit(i,j)$	the value of $b_{n_{bits}-j_i}$ written to facilitate the mathematical demonstration in appendix D *(p. 18)*
$b_{n_{bits}-j_i}$	the value of the bit calculated in conversion step j of sample i *(pp. xv, 13)*
$BW_{OutFilter}$	the bandwidth of the output digital filter *(pp. 10, 25)*
C_{bin}	the unitary capacitor whose function is to create a binary ratio in the ADC *(pp. xvii, 7, 13)*
$C_{k_{p/n}}$	the capacitor of index k from the either of the halves of C-DAC *(pp. xvii, 4, 20, 39)*
C_{k_n}	the capacitor of index k from the negative half of C-DAC *(pp. 48, 49)*
C_{k_p}	the capacitor of index k from the positive half of C-DAC *(pp. 15, 48, 49)*
C_{LSB}	the binary weighted capacitor that relates to the operation of the LSB *(pp. 7, 13)*
C_{MSB}	the binary weighted capacitor that relates to the operation of the MSB *(pp. xv, 7, 13)*
C_{MSB-j}	the binary weighted capacitor that relates to the operation of the MSB-j *(p. 13)*
$C_{MSB_{sec}}$	the sum of all capacitors in either half of the C-DAC excluding the C_{MSB} *(p. 13)*
C_u	the unitary capacitor *(pp. 1, 6)*
Δ_{f/F_S}	a small frequency deviation *(p. 24)*
d_n	the value of the negative decision signal/pointer *(pp. xvi, 19, 20, 39, 41)*
$d_n(i,j)$	the value of the negative decision signal/pointer at conversion step j in sample i *(pp. 18, 19)*
d_p	the value of the positive decision signal/pointer *(pp. xvi, 19, 20, 39, 41)*
$d_p(i,j)$	the value of the positive decision signal/pointer at conversion step j in sample i *(pp. 18, 19)*
$d_{p/n}$	the value of either the positive or negative decision signal/pointer *(pp. 42, 48, 49)*

$d_{p/n}(1,0)$	the value of either the positive or negative decision signal/pointer at conversion step 0, a boundary condition, in sample 1 *(p. 19)*
$d_{p/n}(i,j)$	the value of either the positive or negative decision signal/pointer at conversion step j in sample i *(pp. 42, 48, 49)*
$(e_k)_{p/n}$	the deviation of $(w_k)_{p/n}$ to $(w_{mean})_{p/n}$ *(p. 42)*
$\frac{f}{F_S}$	the ratio between the frequency in a spectrum analysis and the sampling rate, F_S, of the an ADC *(pp. 22–25, 28)*
f_{in}	the input frequency of the ADC *(pp. xvi, 22)*
$\frac{f_{in}}{F_S}$	the ratio between the input frequency of the ADC, f_{in}, and it's sampling rate, F_S *(pp. 13–18, 22–25, 28)*
F_S	the sampling rate of the ADC *(pp. xvi, 22)*
gnd	the negative terminal voltage in an ADC, in this case ground *(pp. xv, 13, 19, 20, 34, 40, 42, 44, 46, 49, 50)*
$H(z)$	the mismatch filter transfer function *(pp. v, vi, 22–24, 26, 27, 29)*
$\frac{KT}{C}$	the thermal noise produced by the switche's resistance and capacitor *(p. 1)*
μ_C	the capacitor's production mean capacitance *(pp. 6, 39)*
N	the total number of capacitors in a given half of a SAR stage C-DAC, typically described as $2^{n_{bits}}$ *(pp. 3, 34, 44, 48, 49)*
n_{bits}	the number of bits or resolution of a stage *(pp. xv, xvi, 3, 6–8, 11, 13–15, 18, 19, 21, 22, 24, 28, 34, 42, 43, 48)*
SC	the selected capacitors by the decision signals d_p or d_n *(p. 40)*
σ	the standard deviation of a general process *(p. 14)*
σ_C	the capacitor's production standard deviation *(p. 6)*
$\Sigma\Delta$	Sigma-Delta Converters *(pp. v, vi, 1, 2, 18)*
$SS(i)$	the value of the switching sequence in sample i *(pp. 27, 28, 46–48)*
TME	the total mismatch error defined in appendix D *(pp. 21, 23–25, 28)*
$TME(z)$	the z-transform of the total mismatch error defined in appendix D *(pp. 23, 24, 28)*
VCM	the common mode voltage in an ADC *(pp. xv, 13, 40, 46, 49, 50)*
V_{in_g}	the value of sample m *(p. 18)*

V_{in_i}	the value of sample i *(p. 18)*
$v_{p/n}$	the value of positive and negative input nodes of the comparator in a SAR stage *(p. 34)*
V_{Ref}	the reference voltage in an ADC *(pp. xv, 13, 19, 20, 34, 40, 42, 44, 46, 48–50)*
$(w_{bin})_{p/n}$	the weight capacitor C_{bin}, on either halves of the C-DAC *(p. 44)*
$(w_k)_{p/n}$	the weight of $C_{k_{p/n}}$ *(pp. xvi, 20, 41)*
$(w_{mean})_{p/n}$	the mean weight of capacitors on either half of the C-DAC, excluding C_{bin} *(pp. xvi, 44)*
y_{DAC}	the real value of the C-DAC residue voltage *(pp. 43, 44)*
$y_{DAC\,Ideal}$	the ideal value of the C-DAC residue voltage *(p. 43)*
$y_{DAC\,Mismatch}$	the mismatched value of the C-DAC residue voltage *(pp. 43, 44)*

1 Introduction

ADCs are at the center of wireless communications systems, these are components that can translate a continuous and infinitely varying signal prone to errors know as analog signal, into a time discrete with quantized levels digital signal resistant to errors.

This quantization can be done with several degrees of resolution (bits) at diverse speeds utilizing different architectures. Some of these architectures are most efficient when the analog signal is composed of high frequency waves, like the flash or pipeline ADCs, and others perform best when the frequency domain of the input signal is relatively low. Some of these architectures comprise the $\Sigma\Delta$ or the SAR ADCs [2–4].

Typically when the ADC is optimized to be able to perform under high speeds, the quantizer is unable to output a long digital word compared with those that are optimized to operate at low frequencies [5]. In order to obtain an ADC that can perform both at high speeds and in the precision domain an expense in power and area, respectively, is eventually necessary. An hybridization of two or more architectures can also be successful at increasing the speed of a converter while only slightly decreasing its resolution. One of the most common hybridizations in the industry is the pipeline structure with SAR stages [5–7]. A SAR assisted pipeline ADC is able to take advantage of the low power consumption of the SAR ADCs and the speed of the pipeline architecture. By splitting the slow but relatively high resolution of a regular SAR ADC into several stages (sub-SAR-ADCs), the global ADC is able to quantize the output as fast as the, now smaller and faster, slowest sub-SAR-ADC stage maintaining the initial resolution. Overall, the resolution is maintained and the speed increased, it comes however with added power consumption due to the pipeline structure.

Because SAR assisted pipeline ADCs rely heavily on the functioning of the native SAR ADCs, some of the major issues that this hybrid architecture possess derives from the latter. Although the introduction of the Capacitive Digital to Analog Converter (C-DAC) into the SAR implementation has major advantages pertaining to the power consumption, it also poses a direct trade-off between the noise that the switch and the capacitor system creates, the $\frac{KT}{C}$ noise, and the area of the circuit itself [6, 8]. Considering the conventional C-DAC implementation for a 5-bit resolution application, 32 unitary capacitors, C_u, are needed if single ended, and 64, if a fully differential ADC is desired. The number of capacitors increases exponentially with the resolution and poses a substantial bandwidth limitation if power dissipation is required to a minimum. If high bandwidth, high resolution, and low power are required, the total input capacitance of the circuit must be kept as low as possible, this will evidently create more $\frac{KT}{C}$ noise since in order to achieve low input capacitance, C_u must be kept at a minimum value. The increase in $\frac{KT}{C}$ noise will then degrade the effective number of bits of the ADC, Effective Number of Bits (ENOB), making it less efficient in quantizing an analog signal.

Furthermore, although large C_u values are conformably fabricated using today's fabrication techniques, small values are not [9, 10]. This is because by fixing both the dielectric material and the distance between the two electrodes, any variation of the capacitance is solely dependent on the area of the electrodes. For large C_u values the area is required to be high, hence the

processes allows for an excellent matching between the first capacitor and the second capacitor of equal nominal capacitance. This effect occurs because errors of fabrication are in absolute forms and not in perceptual ones, favouring large area capacitors over low area capacitors. But as discussed above, for specific applications the capacitors are required to be small to increase the bandwidth and decrease power consumption. Therefore, the then small absolute errors of fabrication can now substantially impact the capacitance value. Consequently, the first capacitor and the second capacitor of equal nominal value can be very different to each other, this is referred to as the capacitor mismatch [9, 10].

Since the relevant SAR derived ADCs quantize the input with respect to ratios of capacitance (by the principle of charge conservation) a large mismatch between two capacitors results in the wrong ratio and consequently in an error in quantization. Theses errors are then seen as spurs throughout the output spectrum severely affecting the SNDR and the SFDR, the two main performance metrics in ADCs [2, 11].

Given that ADCs are at the forefront of wireless communications, where spectral purity is required to maintain channels uncorrupted, controlling spurious sources is paramount in these systems. Given that process variations will always occur, eliminating the mismatch of two elements is impossible, hence, the growing efforts in augmenting the ADC's overall performance are focused on mitigating the effects of the mismatch, at the cost of circuit complexity, rather than pursuing a complete elimination of capacitor differences [12–16].

The work presented is focused on this field of converters, and its goal is to present an extension of the DEM techniques typically used for low bandwidth high resolution ADCs, the $\Sigma\Delta$, to reduce mismatch errors, into the previously discussed hybrid architecture, SAR assisted pipeline ADCs. Utilizing DEM techniques in high resolution ADCs is generally expensive in terms of digital implementation and depends exponentially with the number of bits [17, 18]. By utilizing a pipelined structure, the exponential effect is smaller, while still performing the DEM, this reduces the digital complexity of the circuit by separating the resolution onto several stages. This work will then provide a detailed explanation of the basics of DEM as well as a mathematical description of the effect for DEM applied to SAR stages. To corroborate the techniques demonstrated a script containing a high level analysis of the SAR Pipeline as well as of the DEM techniques will be used. The specific architecture of the SAR stage used to demonstrate the techniques is the VCM based SAR structure.

2 SAR ADC Analysis

The present chapter aims to introduce the mathematical basis of a SAR architecture, as well as introduce the effects that the mismatch between capacitors have on the spectrum of the ADC.

2.1 VCM-based SAR Stage Functionality

Figure 2.1 depicts a diagram for the 3-bit VCM-based stage to be analyzed. A SAR stage in a pipeline differs from the regular SAR ADC in the sense that for the stage to output n_{bits}, a C-DAC of n_{bits} ($2^{n_{bits}+1}$ capacitors) is necessary whereas for a SAR ADC the same C-DAC allows for an extra quantization of the signal resulting in a resolution of $n_{bits}+1$. The SAR ADC, at the last conversion step, can utilize the voltages at the input of the comparator to determine if the difference is positive or negative, calculating the value of the $(n_{bits}+1)^{th}$ bit. However, the SAR stage needs to output the residue of the quantization to the next stage, only calculating n_{bits} bits. In the case of figure 2.1 if it were a fully realized ADC, the diagram would represent a 4-bit SAR ADC instead of the 3-bit stage implementation. At the last stage of the pipeline, no further residue is needed therefore the resolution of the last SAR sub-ADC is the same as of a SAR ADC. Nevertheless the equations that describe both the SAR stage and the SAR ADC are the same.

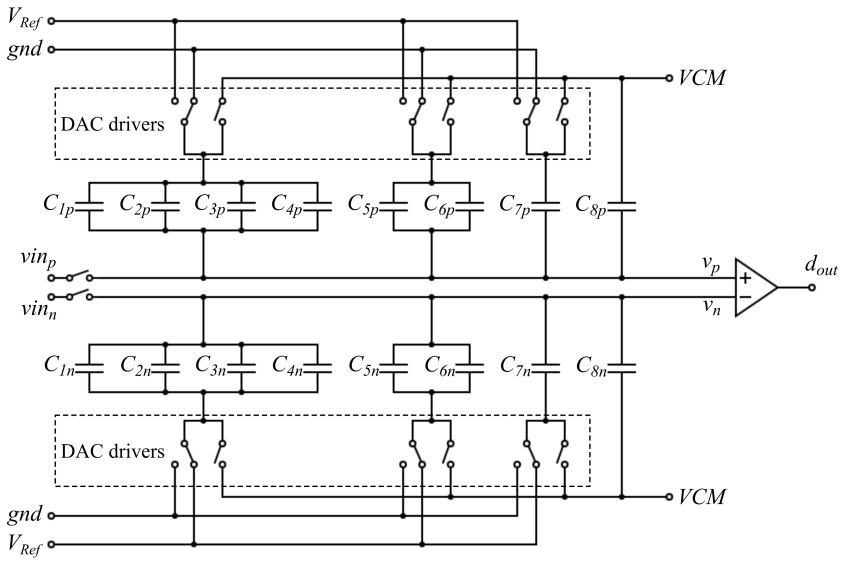


Figure 2.1: A 3-bit VCM-based SAR ADC/Stage with capacitor discrimination.

In addition to the sampling step and the residue calculation, when necessary, the SAR architecture realizes as many steps as the number of bits it can resolve, these steps can be described by the set of equations presented and derived in appendix A. Equations, (A.1c), (A.2c), (A.3c) and (A.4c) allow the conclusion that it is possible to derive the functionality of the SAR stage depicted in figure 2.1 by analysing only the final residue calculated by the C-DAC, equation (A.4c). Therefore each step of the conversion performed by the stage can be derived using only the residue voltage equation at the final step. To generalize, equation (2.1) defines the residue voltage for the case of a SAR stage with n_{bits}, where $N = 2^{n_{bits}}$:

$$V_{Residue} = (vin_p - vin_n) - VCM\left[\frac{\sum_{k=1}^{N-1} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{\sum_{k=1}^{N-1} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right]$$
$$+ V_{Ref}\left[\frac{1}{\sum_{k=1}^{N} C_{k_p}} \sum_{j=1}^{n_{bits}} \overline{b_{n_{bits}-j}} \left(\sum_{k=N+1-2^{n_{bits}-j+1}}^{N-2^{n_{bits}-j}} C_{k_p}\right) - \frac{1}{\sum_{k=1}^{N} C_{k_n}} \sum_{j=0}^{n_{bits}} b_{n_{bits}-j} \left(\sum_{k=N+1-2^{n_{bits}-j+1}}^{N-2^{n_{bits}-j}} C_{k_n}\right)\right]$$
$$+ gnd\left[\frac{1}{\sum_{k=1}^{N} C_{k_p}} \sum_{j=0}^{n_{bits}} b_{n_{bits}-j} \left(\sum_{k=N+1-2^{n_{bits}-j+1}}^{N-2^{n_{bits}-j}} C_{k_p}\right) - \frac{1}{\sum_{k=1}^{N} C_{k_n}} \sum_{j=0}^{n_{bits}} \overline{b_{n_{bits}-j}} \left(\sum_{k=N+1-2^{n_{bits}-j+1}}^{N-2^{n_{bits}-j}} C_{k_n}\right)\right] \tag{2.1}$$

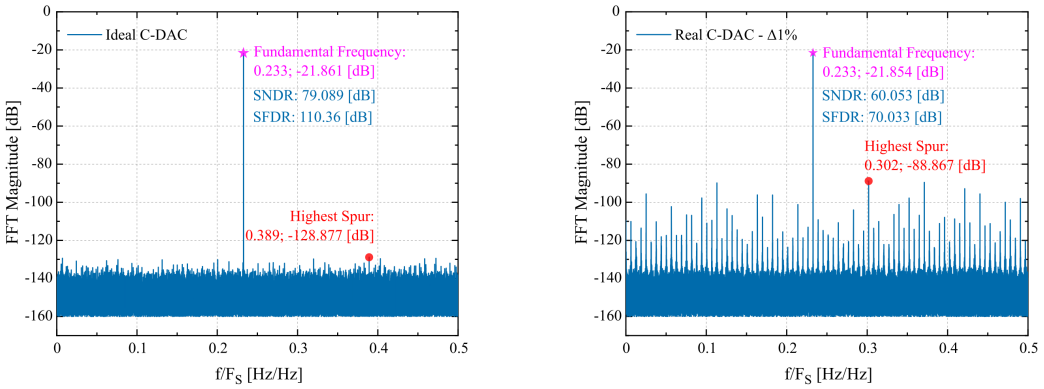
If there were no errors affecting the value of the capacitors $C_{k_{p/n}}$, this is, no mismatch between capacitors, equation (2.1) would collapse to equation (2.2):

$$V_{Residue} = (vin_p - vin_n) + \left(V_{Ref} - gnd\right) \sum_{j=1}^{n_{bits}} (\overline{b_{n_{bits}-j}} - b_{n_{bits}-j}) 2^{-j} \tag{2.2}$$

2.2 Mismatch Effects

As equation (2.1) suggests, the realization of purely mathematical analysis will produce very complex equations that although from a theoretical point of view might hold much of the information necessary, from a practical point of view, it requires extensive analytical resources to put forth. Therefore, to test the effects that capacitor mismatch has on the performance of the global ADC, a script developed in *MatLab* is going to be used. The script was developed to encase a high level model of a generic SAR assisted pipeline ADC, with possible engagement in dynamic element matching techniques across different stages. The simulations are ran for a pipeline of three stages of overall resolution of 13 bits with digital correction (industry relevant). To effectively test the effects of capacitor mismatch and of dynamic element matching techniques to be onward described, only the first stage (the MSBs containing stage) will be target of dynamic element matching techniques on capacitors with relative errors of 1%. As seen in [9, 10], the capacitor errors is in most of the cases less than 1%, by setting the relative error of the capacitors to this value, it will be possible to observe the improvements that certain DEM algorithms bring into the possible worst case scenarios. The remainder of the stages will be set as ideal so to observe the isolated effect that these DEM techniques have on a stage. The pipeline structures that are going to be used will be [366] or [555], with each of the numbers describing the resolution in bits of each of the stages. These allow for realistic implementations of the DEM while illustrating the most relevant effects of DEM. Figure 2.2a depicts the output magnitude frequency spectra, Fast Fourier Transform (FFT), of a 3 stage pipeline SAR ADC of 5 bit resolution per stage, [555], with no errors across all stages. Figure 2.2b represents the FFT with 1% relative errors introduced in the first stage's capacitors, where it is possible to see that these errors translate into spurious tones throughout the spectrum.

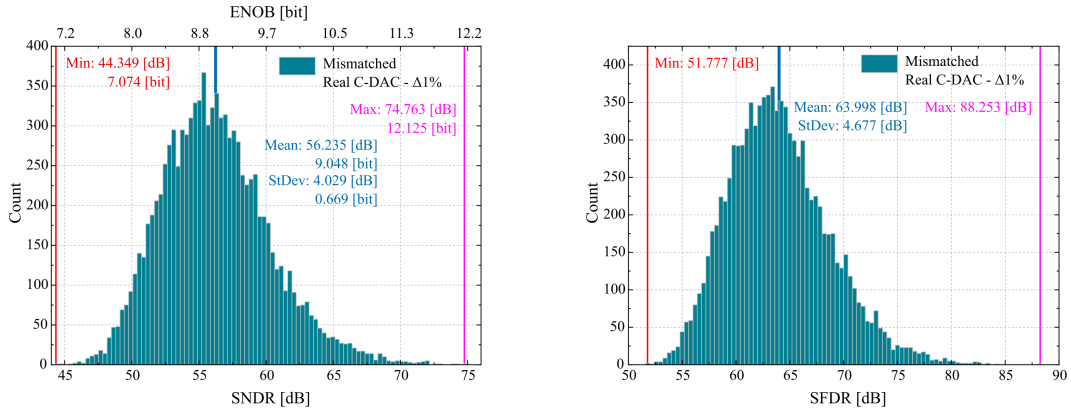
Due to the random nature of the mismatch errors, there will be some runs that will provide better results than others. To realize a full and concrete comparison between two different



(a) Ideal ADC **(b)** Mismatch Test

Figure 2.2: Output spectra of the pipeline SAR ADC's in a) with ideal capacitors, and in b) with mismatch applied to them.

settings it is necessary to control for this variability so that the results are not randomly biased. To control for the randomization across multiple runs, a Monte-Carlo (MC) analysis is performed, this will be able to confidently produce results that fall under the best and worst performance for that particular setting simultaneously. For comparisons between different settings or algorithms, the same seed of random variability will be used, this will make the errors of the capacitors, in a given MC run, the same across any test. The results of these simulations will provide a histogram that holds information about the distribution of the SNDR/ENOB and SFDR of the ADC under that particular setting. The output spectra for the worst scenario encountered in the MC analysis will be shown when it is relevant for the discussion. Figures 2.3 and 2.4 illustrate the results of such analysis performed for [555] pipeline SAR ADC that has 1% error in the first stage, with no mismatch effect mitigation techniques.



(a) SNDR and ENOB of the ADC **(b)** SNDR of the ADC

Figure 2.3: MC analysis of the mismatch effect in the output spectra of the ADC.

From a statistical point of view, the equations previously mentioned, specifically equation (2.1), asserts that to produce the ratios for the quantization of the input, a sum of unit capacitors is essentially performed. This effect has consequences, namely if the capacitors are described

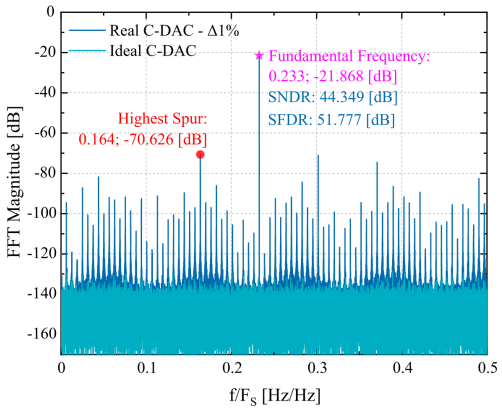


Figure 2.4: Comparison of the output spectra of the ADC with mismatch in dark blue, and the ideal in light blue. This case represents the run out of 10000 MC runs that rendered the lowest SNDR.

as identically and independently distributed variables, that have a mean equal to their nominal value, $\mu_C = C_u$, and a standard deviation, σ_C. From an academical point of view the indefinite sum of capacitances, correspondent to $n_{bits} \to \infty$, will produce the same result as if the capacitors would not have any errors at all. To quantize this result, equation (2.3c) refers to the standard deviation of the relative error of the sum of capacitors with respect to their mean.

$$E\left[\sum_{k=1}^{N} C_k\right] = N\mu_C \tag{2.3a}$$

$$\sigma\left(\sum_{k=1}^{N} C_k\right) = \sqrt{N}\sigma_C \tag{2.3b}$$

$$\begin{aligned}\sigma\left(\frac{\sum_{k=1}^{N} C_k - N\mu_C}{N\mu_C}\right) &= \sqrt{E\left[\left(\frac{\sum_{k=1}^{N} C_k - N\mu_C}{N\mu_C} - E\left[\frac{\sum_{k=1}^{N} C_k - N\mu_C}{N\mu_C}\right]\right)^2\right]} = \\ &= \sqrt{E\left[\left(\frac{\sum_{k=1}^{N} C_k - E\left[\sum_{k=1}^{N} C_k\right]}{N\mu_C}\right)^2\right]} = \frac{\sqrt{E\left[\left(\sum_{k=1}^{N} C_k - E\left[\sum_{k=1}^{N} C_k\right]\right)^2\right]}}{N\mu_C} = \\ &= \frac{1}{N\mu_C}\sigma\left(\sum_{k=1}^{N} C_k\right) = \frac{1}{\sqrt{N}}\frac{\sigma_C}{\mu_C}\end{aligned} \tag{2.3c}$$

Equation (2.3c) corroborates the statement presented before. As the sum of the capacitors gets greater, their average is less spread out, noted by the fact that $\lim_{N\to\infty} \frac{1}{\sqrt{N}}\frac{\sigma_C}{\mu_C} = 0$, as such, the sum of N mismatched capacitances, as N approaches ∞, is the same as the sum of N ideal capacitances. This means that for larger resolution stages, the same relative error in the capacitors will produce less adverse effects overall, however the area of these stages will be exponentially higher as an added bit of resolution increases the number of capacitors needed by a factor of 2.

3 Sampling Frequency Dependent Dynamic Element Matching Algorithms

3.1 Dithering Effect

From the capacitors point of view it is clear that not all permutations of a single array lead to the same error, and that these are dependent on the previous comparison. Supposing an array of capacitors, of length $2^{n_{bits}}$, the first $2^{n_{bits}-1}$ capacitors are reserved for the operation of the MSB, C_{MSB}, after these, the first $2^{n_{bits}-2}$ capacitors represent the Second Most Significant Bit (MSB-1) (C_{MSB}), and the same for the rest of MSBs until the last 2 capacitors are left. The last one is a capacitor that performs the binary weight during a conversion, C_{bin}, and the previous one refers to the operation of the LSB, C_{LSB}. The MSB capacitor, the capacitor that results from the parallel connection of the first $2^{n_{bits}-1}$ unit capacitors, can have different values depending on the permutation, which results in different ratios in accordance with equation (2.1). Different permutations can result in different errors, but not with exact certainty. Supposing the permutations seen in table 3.1 for the positive half of the C-DAC it is possible to see that although all permutations are different, not all produce different errors.

Table 3.1: Different permutations of unit capacitors according to their position in a C-DAC of a 3 bit stage.

		Capacitors							
		C_{MSB}				C_{MSB-1}		C_{LSB}	C_{bin}
Permutations	P1	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
	P2	C_2	C_3	C_4	C_1	C_6	C_5	C_7	C_8
	P3	C_3	C_4	C_2	C_1	C_8	C_7	C_6	C_5
	P4	C_3	C_2	C_3	C_4	C_8	C_5	C_7	C_6
	P5	C_5	C_6	C_7	C_8	C_3	C_4	C_1	C_2

The permutations shown are all different, however if the input is the same then the residue voltage is the same across permutations P1 and P2. Furthermore the error referring to the MSB capacitor is the same in permutations P1, P2, P3 and P4. Only for permutation P5 is the error different across all capacitors. Although for the particular scenario that the input is the same the permutations shown can have the same errors, if considered otherwise, then each of the permutations would produce different errors. This particular example relates to the fact that dynamic element matching and unit component permutation is highly dependent on the input and is very difficult, although possible as it will be explained in later chapters, to foresee the outcome that a particular permutation of capacitors has on the quantization of several inputs. The effects will therefore be tested recurring to the script described earlier.

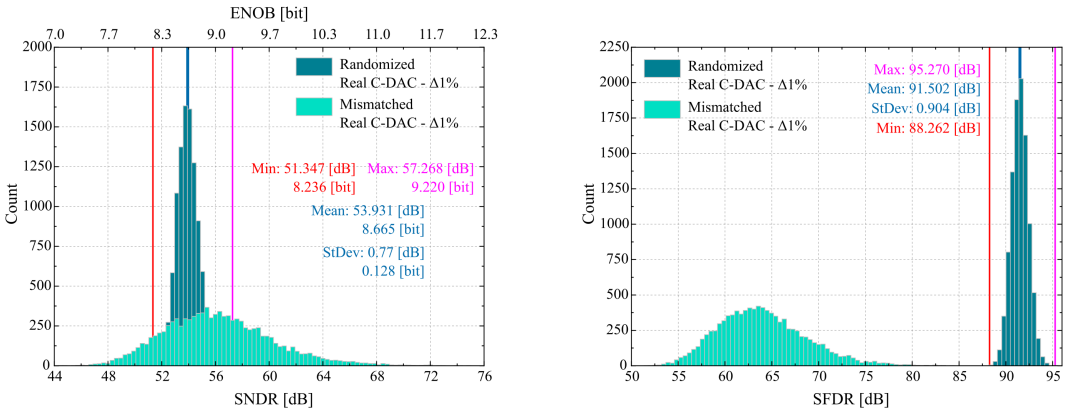
3.1.1 Random Permutation

The simplest type of permutations that can be tested are those that are independent of the quantization signal, and that are dependent only on the sample number. Every time that one conversion is performed the permutation changes, or follows a pattern. It is necessary to take into account that the goal is to decrease the spurious energy, over all frequencies, or around the fundamental frequency, with the least reposition of energy in the noise floor, this aims to increase the SFDR while maintaining or increasing the SNDR. One permutation that can accomplish this spectral effect is a random permutation of capacitors [19–21]. In an array of capacitors, considering the positive and the negative arrays together, the random permutation allows the unitary capacitors to be "ordered" randomly every conversion cycle. In a SAR stage of n_{bits} where $2^{n_{bits}+1}$ capacitors are present, there are $\left(2^{n_{bits}+1}\right)!$ possible permutations, where not every possible permutation conduces to different errors and are necessarily dependent on the conversion signal as described earlier.

This type of permutation can be based on a random number generator such that chooses, at the sampling rate, one of the $\left(2^{n_{bits}+1}\right)!$ possible permutations to perform the conversion of a particular sample. The error of these randomly selected permutations is not only "randomized" in accordance with the pseudo-random sequence that the generator outputs, but they also contain the modulation of the input signal, since it has been already established that the error of the same permutation is highly dependent on the input. This kind of permutation is equivalent to establishing a dithering effect on the quantization of the input, that can be optimized to contain a smaller or greater pseudo-random period, so that different amounts of errors are permuted which increases or decreases spurious effects but has the reverse effect on the noise level.

From a general point of view a patterned or otherwise unpatterned permutation can be seen as an average of the errors given by each of the permutations [22]. When no swap is performed, only one permutation is established and the error results in a limited amount of spurs further modulated by the different inputs. Randomly permuting the capacitors allows for the spurs to be of greater number, but the energy of those spurs is much lower, resulting in an almost random noise [19]. This effect can be seen not only in the spectrum of the output of an ADC that performs this kind of permutation but also in MC analysis. Figures 3.1 and 3.2 depicts this effect.

It is possible to see that with the permutation, the consistency of results with respect to SNDR, and consequently ENOB, is much greater, which allows a better circuit yield, if the area necessary for this implementation is disconsidered. The better consistency in results inevitably means that the cases where the errors have a small effect in the input quantization, and therefore the performance is greater, are lost with the introduction of the random permutation. However, the same applies to the worst case scenarios, with the random selection of the capacitors, the cases where the errors were severely impacting the performance of the ADC are effectively removed to approximate the average. Furthermore, comparing with the case where no permutation is applied, figure 2.3, a large increase in the SFDR of the ADC is now noticeable,



(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.1: MC analysis of the randomized and mismatched C-DAC effect in the output spectra of the ADC.

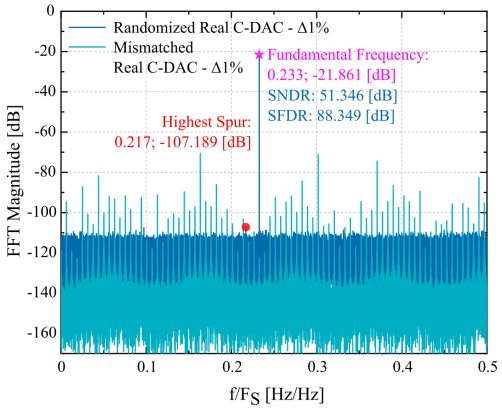
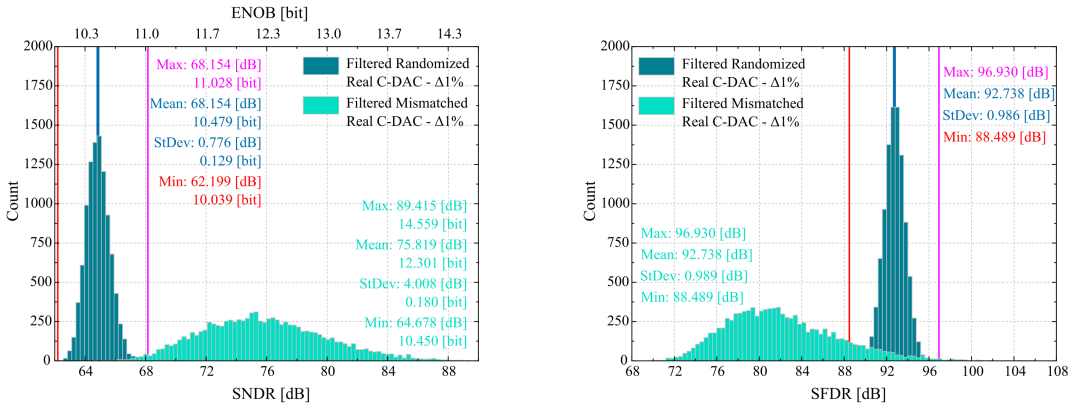


Figure 3.2: Comparison of the output spectra of the ADC with randomization in dark blue, and the mismatch in light blue. This case represents the run out of 10000 MC runs that rendered the lowest SNDR.

this increase is much greater that the eventual loss in the SNDR domain. The spectra depicted in figure 3.2 shows the comparison between the C-DAC with no DEM, onward referred to as mismatched C-DAC, and the randomized C-DAC, with random capacitor selection, and it is noticeable the increase of the noise floor but without any spurs.

A digital filter can be introduced to boost performance as seen in figures 3.3 and 3.4. It is possible to see that, with the digital filter, the SNDR of both the randomized and mismatched ADCs increase, however this increase is most notable in the mismatched ADC. Since the existence of spurs is what contributes to the degradation of the SNDR and SFDR, the elimination of a great portion of those spurs, boosts the SNDR of the ADC surpassing that of the randomized ADC. The SFDR also has this effect where the mismatched ADC is more notably affected by the filtering, since the filter reduces most of the spurs present. Given that the random permutation averages all errors across the whole spectrum, the influence of the before highest energy spurs is still seen after the randomization in the band pass. Although not particularly useful for the random permutation, the implementation of digital filtering is necessary as some DEM

techniques allow only for the mitigation or cancellation of capacitor mismatch around the fundamental frequency, these permutations will be discussed in the next section 3.2 and chapter 4. The output digital filter was implemented using a butterworth filter of order 6 with bandwidth $BW_{OutFilter} = 0.04$, from the signal processing toolbox of *MatLab*.



(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.3: MC analysis of the randomized and mismatched C-DAC effect in the filtered output spectra of the ADC.

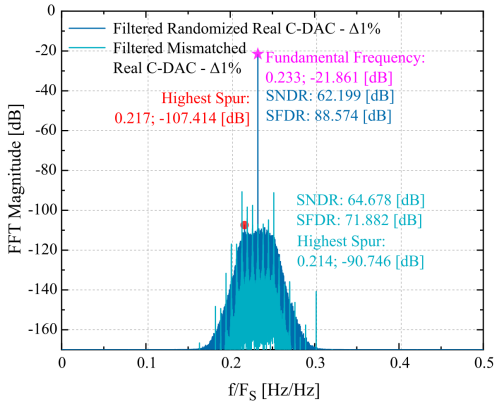


Figure 3.4: Comparison of the filtered output spectra of the ADC with randomization in dark blue, and the mismatch in light blue. This case represents the run out of 10000 MC runs that rendered the lowest SNDR.

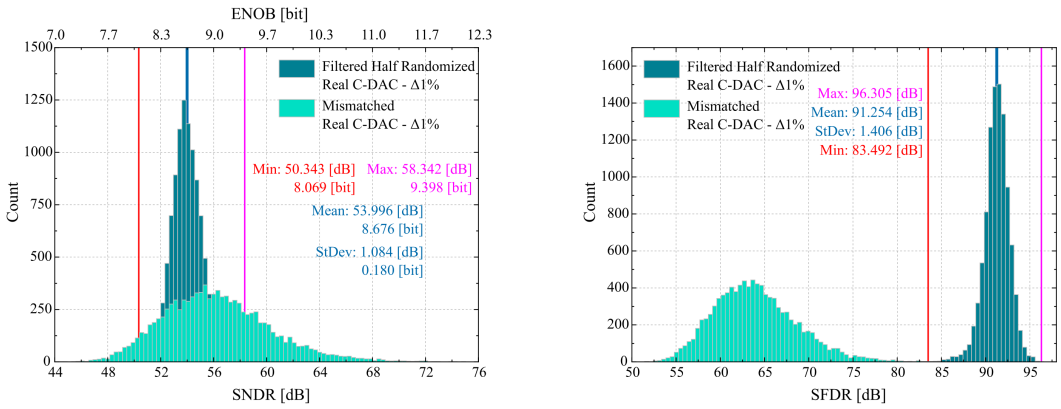
3.1.2 Half Random Permutation

The random permutation can be altered by looking at the equations derived earlier. Due to the differential nature of equation (2.1), ideally the negative half of the C-DAC would produce the same gain coefficients as the positive half. By performing the random selection under all possible capacitors crossing over the positive and negative arrays, the errors are being averaged in such a way that the positive and the negative ratios tend not only to the ideal value, but also to equal each other. However, since the ideal values for both positive and negative ratios are the same, the effects of obtaining the ideal value can be done only by making the positive and

negative sides converge to the ideal ratios independently. This effect can be generated by applying the random permutation to the positive and negative arrays separately. This means that the spectral properties of the output will be very similar to those of the crossed-over randomization, but is achieved with less possible permutations, $2\,(2^{n_{bits}})!$ instead of $\left(2^{n_{bits}+1}\right)!$. Besides requiring less permutations, since there is no crossover between the capacitor of the positive and negative sides, the digital circuit that implements this configuration can be much simpler.

It is very important to keep in mind that although the half random permutation allows for less complexity and less permutations to be chosen, even if the number of capacitors is low, for example for a 3 bit stage, the number of possible permutations is very high, 40320, and has a factorial growth rate for every added capacitor and an exponentially higher factorial growth rate for every added bit (4 bit stage would have approximately 21e12 possible permutations). In practice, these implementations would be unfeasible for high resolution stages and are only limited to very low resolution stages.

Nonetheless, the results depicted in figure 3.5 corroborates the statement earlier regarding the optimizable nature of the dithering effect, fewer permutations leads to a similar noise level with the possibility in increase of energy in the spurious tones. Less permutations means that the average of the errors is less consistent among different MC runs, resulting in a larger spread of performance, seen in the increase, although marginally, of the standard deviation in both SFDR and SNDR histogram plots. The average of these metrics is however greater. Furthermore the same results reached while applying the digital filter to the output of the fully randomized ADC are replicated for this permutation, seen in figures 3.7 and 3.8.



(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.5: MC analysis of the half randomized and mismatched C-DAC effect in the output spectra of the ADC.

3.2 Sequential Permutations and Noise shaping

As previously mentioned, the goal of DEM is to reduce the energy of the noise/spurs either over the entirety of the spectrum or locally around the fundamental frequency. The removal of the noise/spurs around the fundamental frequency refers to the noise-shaping characteristic

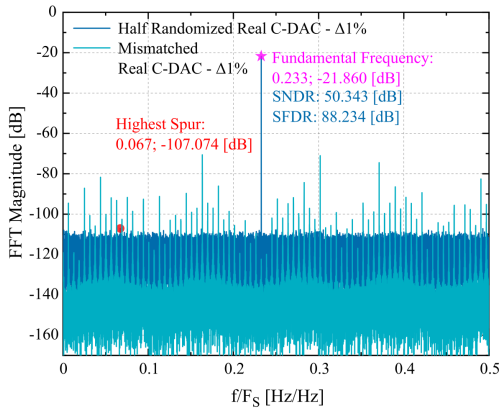
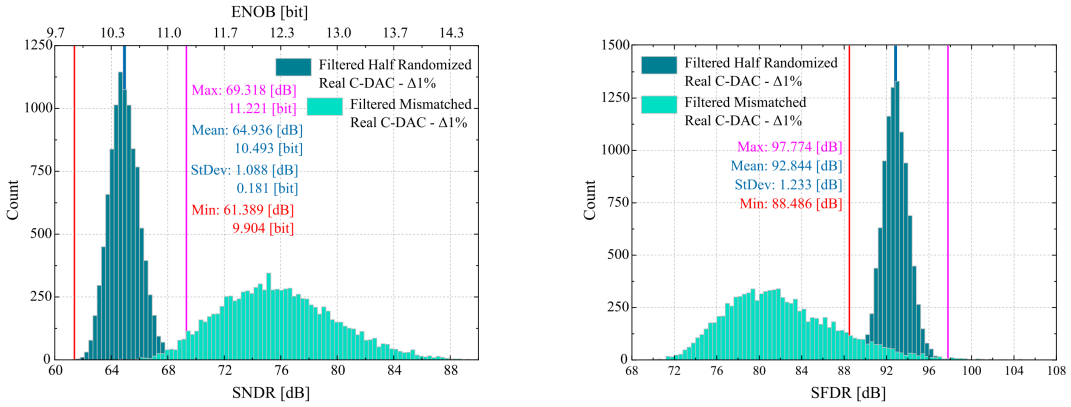


Figure 3.6: Comparison of the output spectra of the ADC with the half randomization in dark blue, and the mismatch in light blue. This case represents the run out of 10000 MC runs that rendered the lowest SNDR.



(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.7: MC analysis of the half randomized and mismatched C-DAC effect in the filtered output spectra of the ADC.

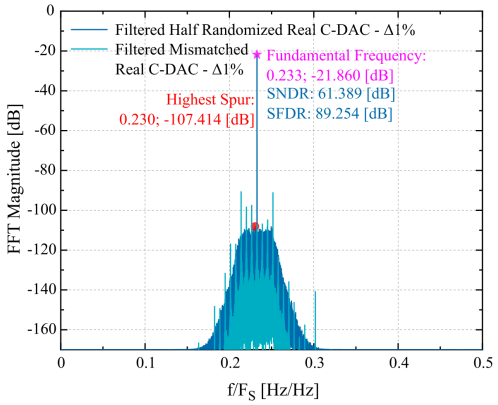


Figure 3.8: Comparison of the filtered output spectra of the ADC with the half randomization in dark blue, and the mismatch in light blue. This case represents the run out of 10000 MC runs that rendered the lowest SNDR.

of the DEM. Generating this effect, recurs to the selection of a given permutation taking into

consideration the single or multiple previous permutations composing an overall repeating pattern or sequence [23–27]. This effect will naturally depend on the ratio $\frac{f_{in}}{F_S}$, given that the sequence of permutations constructed follows independently whether or not the quantized samples have a specific relationship.

3.2.1 MSB Permutation

An example of the described effect, can be demonstrated for a permutation that considers a periodic swap between the MSB capacitors and the remaining capacitors in each of the sides of the C-DAC, while applying a random permutation mentioned in section 3.1.2.

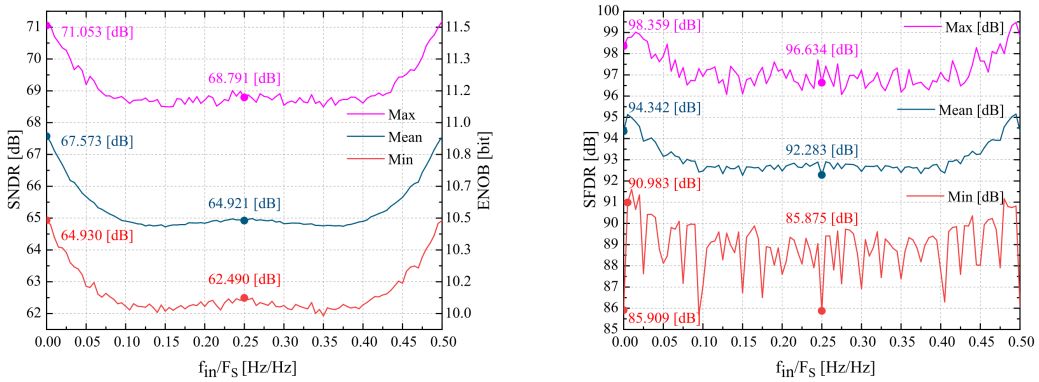
This sequence takes into consideration that the output spectrum is a function of the ratio between the input frequency and the sampling rate, $\frac{f_{in}}{F_S}$. The MSB permutation, is a simple patterned sequence of period two, meaning that the pattern repeats every 2 samples. The first permutation is a random permutation (selecting only one of the halves of the C-DAC) and the second is the swap between the MSB capacitors. This type of permutation, unlike the previous dithering permutations, outputs different spectral properties depending on the ratio $\frac{f_{in}}{F_S}$. At $\frac{f_{in}}{F_S} \approx 0$ the sampled input and consequently output will be consecutively similar, if the ADC operates in these conditions then a permutation can be engineered to explicitly cancel the mismatch of the capacitors [26, 28].

In a given conversion, a permutation of capacitors can be described, in both halves simultaneously by the binary weighted capacitors: the C_{MSB} composed of the parallel of $2^{n_{bits}-1}$ capacitors, the C_{MSB-j} composed of the parallel of $2^{n_{bits}-j}$ and the binary capacitor of the same weight of the C_{LSB}, C_{bin}. The MSB permutation addresses the fact that the single most consequential error that can ever occur impacting the overall linearity of the ADC is at the MSB level. By realizing the swap between the C_{MSB} and the other $2^{n_{bits}-1}$ capacitors, $C_{MSB_{sec}} = C_{bin} + \sum_{j=1}^{n_{bits}-1} C_{MSB-j}$, the error referring to the MSB ratio is eliminated. This is because performing permutations over samples, as described by section 3.1 is, from the practical point of view, an average of the errors. Taking for simplicity a 3 bit stage regarding only ratios V_{Ref} of equation (2.1) from the positive half of the C-DAC, this effect can be mathematically described, by equation (B.1) in appendix B. If $\frac{f_{in}}{F_S} = 0$ then $b_{n_{bits}-j_i} = b_{n_{bits}-j_{i+1}}$, the input samples are the same, meaning that the average in equation (B.1) collapses to the ideal value for the ratio of MSB seen in equation (B.2) in appendix B.

Equation (B.2) implies that the residue voltage of the ADC on the MSB step is completely eliminated. Furthermore, due to equation (2.3c) it is possible to argue that the errors of the MSB-1 and Third Most Significant Bit (MSB-2) are reduced since the corresponding ratios are performed using a larger sum of capacitors. Although this example uses the fact that the entire output code (input sample) is to some extent the same, the cancellation of the MSB error can have its conditions relaxed. If only the MSB is the same across 2 samples, meaning $\frac{f_{in}}{F_S} \approx 0$, the cancellation still occurs, since to go from equation (B.1) to (B.2), on the MSB ratio, only $b_{2_i} = b_{2_{i+1}}$ is necessary, however the reduction of the remaining ratios is not performed. The equations (B.1) and (B.2) shown can be derived for the negative half and for the gnd and VCM

ratios, as well as for the n_{bits} general case.

Conventionally, a simple analysis of the spectrum at a particular input frequency would not render the result described, it is only with the sweep of the performance though several ratios $\frac{f_{in}}{F_S} \in \left[0, \frac{1}{2}\right]$ that this effect is noticeable. By allowing the frequency to vary, calculating the SNDR and SFDR over a band near the fundamental frequency, it is possible to observe the error cancellation at low frequencies in figure 3.9. Due to simulation timing constraints, as the frequency was swept the number of MC runs performed was drastically decreased to 400. Whenever the frequency sweep is utilized to generate results the MC runs will be 400 so to reduce measurement times. The intent of these simulations is not to show conformity to industry standards (3σ), but to corroborate the theoretical explanation put forward to predict the results.



(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.9: Performance of an ADC that undergoes MSB swap, on the first stage of resolution 5 bits as the input frequency changes. The data were obtained by running 400 MC runs for each frequency swept.

The permutation of capacitors relies not only on the described algorithm, but also on the randomization of the initial permutation to reduce the spurious tones whenever the cancellation of the MSB and the reduction of the LSBs is not possible.

Figure 3.9 depicts the effects of the permutation, as expected when $\frac{f_{in}}{F_S}$ approaches the ideal value $0; \frac{1}{2}$, the performance in both SNDR and SFDR increases, due to the cancellation of the MSB. It is also noticeable an increase in performance at $\frac{f_{in}}{F_S} = 0.25$, although smaller than the previous mentioned increase. At this ratio, the effect of MSB swap is not seen, however the dithering effect is. Remembering that the dithering effect can be achieved by a utilizing different number of permutations, the less permutations that are used the better for the noise level. In this case, the condition $\frac{f_{in}}{F_S} = 0.25$ means that the input of the ADC is the same every 2 samples, and every two samples the permutation that occurs is a random one. Since the random permutation sees the same sample in a period of two, then the average will be more effective resulting in a lower noise floor and hence a greater SNDR and SFDR. A comparison between the spectral properties for both frequencies, $\frac{f_{in}}{F_S} = 0$ and $\frac{f_{in}}{F_S} = 0.25$ is depicted in figure 3.10. It is possible to see that at $\frac{f_{in}}{F_S} = 0.25$ the ADC has similar properties to those shown when the C-DAC is randomized, at $\frac{f_{in}}{F_S} = 0$ it is possible to see a decrease of the noise floor near the fundamental frequency.

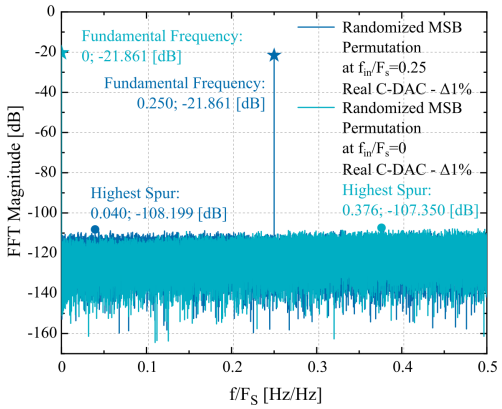


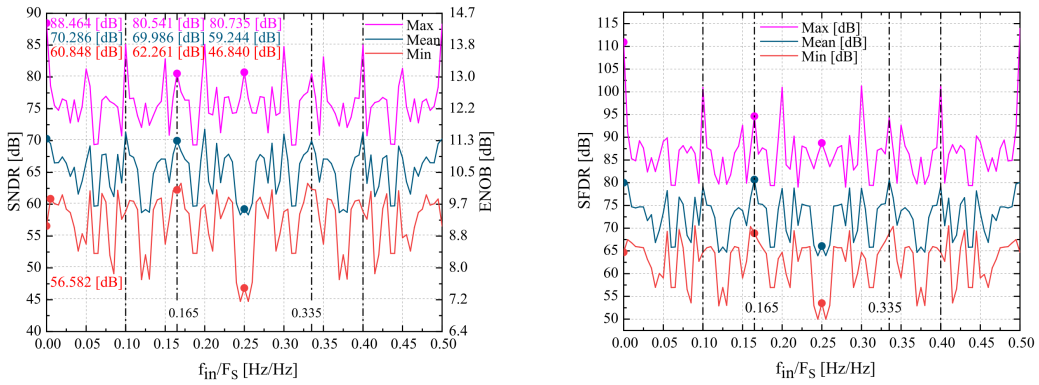
Figure 3.10: Output spectra comparison of an ADC that undergoes the randomized MSB permutation on a 5 bit stage for input frequencies of $f_{in}/F_S = 0$, in light blue, and $f_{in}/F_S = 0.25$, in dark blue.

3.2.2 Full Resolution Level MSB Permutation

The previous example refers to the cancellation of the MSB error, however, it is possible to obtain the cancellation of the first α bits, mentioned in [26, 28] as the α-level MSB swap. To have this effect the first α bits must stay constant, for at least 2^α samples, and the permutation has to swap all the capacitors such that at the end of the 2^α samples all of the unit capacitors have been at least once apart of $C_{MSB-\alpha+1}$. This ensures that the numerator of the average of the ratios is exactly an integer number of $\sum_{k=1}^{n_{bits}} C_{k_p}$, meaning that the error of $MSB - \alpha + 1$ is cancelled. The permutation for a 3-bit stage in which all of the mismatch errors are eliminated follows table C.1[29] in appendix C.

One must note that although table C.1 depicts only one of the halves of the C-DAC, the same applies for the other and that the algorithm only works if both halves are permuted the same way as the other while capacitors from the negative and positive halves cannot crossover to the other half. Through the tabular definition of the algorithm, it is possible to see that the MSB capacitor is composed of all the unit capacitors in a pair of samples, and as seen previously this effect results in the cancellation of the errors regarding the MSB (equation (B.2)). Likewise, the MSB-1 capacitor is composed of all unit capacitors considering 4 consecutive samples and the LSB capacitor is composed of all unit capacitors considering 8 consecutive samples. This particular implementation of the algorithm results in the cancellation of all mismatch errors throughout 8 samples, however if the intent of the designer is to permute only the first 2 bits instead of all 3, then the permutation could follow the order described in the table C.1 up to Vin_4. Thereafter, in Vin_5, the permutation would refresh and would either be an exact copy of the permutation seen in Vin_1 or would be a randomly generated one. The randomness of the first permutation would in turn reduce spurs left from the unsuccessful cancellation of the mismatch errors due to $\frac{f_{in}}{F_S}$ shifts. To realize whichever level MSB permutation a block diagram shown in figure C.1, appendix C, could be conceptually used with the help of n_{bits} clocks that could be generated from counters utilizing the already existing SAR clock. The Switching blocks, whenever triggered by the correspondent clock, swaps the two outputs performing the permutation. These blocks take inspiration from [17, 18, 25], which uses a similar structure to

implement a tree structure algorithm, however their usage in this diagram is purely conceptual given that the inner circuitry has not been developed. The rising edge of the clocks that drive the Switching blocks are depicted in table C.2, in appendix C. As introduced before, the ideal frequency for these permutations is zero, $\frac{f_{in}}{F_S} = 0$, however, it is possible to see that this particular frequency is not unique for the cancellation of the MSB error. The MSB capacitor is also composed of all unit capacitors if the $Vin_1 = Vin_4$, which corresponds to a frequency of $\frac{f_{in}}{F_S} = \frac{1}{6}$, a repetition of samples every 3 inputs. If a random permutation is not chosen after Vin_8, then it is also possible to conclude that all other errors are eliminated and that the performance of the ADC increases near $\frac{f_{in}}{F_S} = \frac{1}{6}$. The same cannot be said considering a repetition of samples every two inputs, at this frequency, $\frac{f_{in}}{F_S} = 0.25$ the MSB capacitor does not change unit elements, meaning that the mismatch error is preserved and the performance of the ADC is deteriorated, although the remainder binary weighted capacitors do change, they only do so until they are constituted by the same 4 unit elements, constituting a decrease in the mismatch but not a full elimination of the error. Table C.1 is a good representation of the algorithm given that it allows the reader the perception of what particular frequencies will affect the performance of the ADC positively or otherwise. The results of the permutation can be seen in figure 3.11, where no dithering effect is performed, corroborating the comments above mentioned, for a stage of 3 bits.



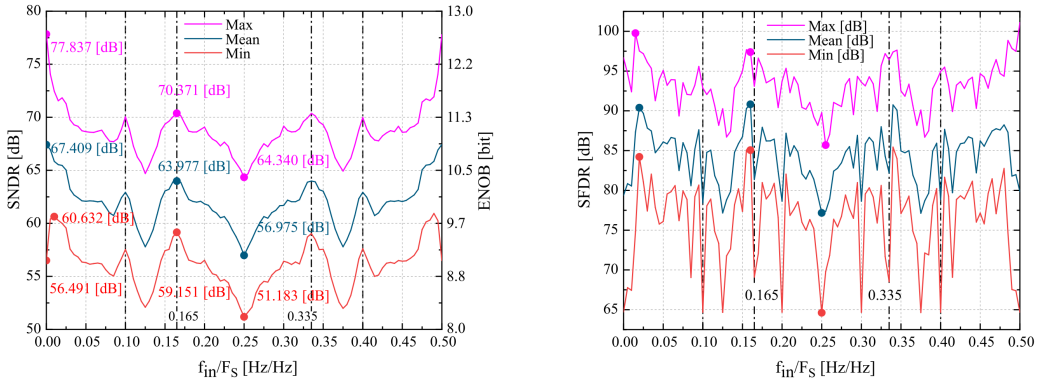
(a) SNDR and ENOB of the ADC

(b) SFDR of the ADC

Figure 3.11: Performance of an ADC that undergoes non-dithered level 3 MSB swap, on the first stage of resolution 3 bits as the input frequency changes. The data were obtained by running 400 MC runs for each frequency swept.

In figure 3.11 it is possible to see the increase of the performance at specific frequencies, including but not limited to those mentioned with the help of table C.1, the other frequencies such as $\frac{f_{in}}{F_S} = \frac{1}{10}; \frac{1}{20}$, rely heavily on the non randomization of the permutation, more so than $\frac{f_{in}}{F_S} = \frac{1}{6}$, given that they require several repetitions of the pattern in table C.1 with the exact same initial permutation to produce the cancellation. If a dither effect would be used, the frequencies that require a greater repetition of the pattern presented on table C.1, would result in a much lower performance. Figure 3.12 depicts the performance of the ADC when a random permutation is chosen at the beginning of the pattern. It is possible to see the intense decrease of performance on the frequencies that rely heavily on several repetitions of the pattern in table

C.2.



(a) SNDR of the ADC

(b) SFDR of the ADC

Figure 3.12: Performance of an ADC that undergoes dithered/randomized level 3 MSB swap, on the first stage of resolution 3 bits as the input frequency changes. The data were obtained by running 400 MC runs for each frequency swept.

A comparison between the spectral properties for both frequencies, $\frac{f_{in}}{F_S} = 0$ and $\frac{f_{in}}{F_S} = 0.25$ is depicted in figure 3.13. It is possible to see that at $\frac{f_{in}}{F_S} = 0.25$ the ADC has a high noise floor near the fundamental frequency, whereas at $\frac{f_{in}}{F_S} = 0$, the noise floor near the fundamental frequency is lower. Therefore, when filtering, $\frac{f_{in}}{F_S} = 0$ has a better overall performance than when $\frac{f_{in}}{F_S} = 0.25$.

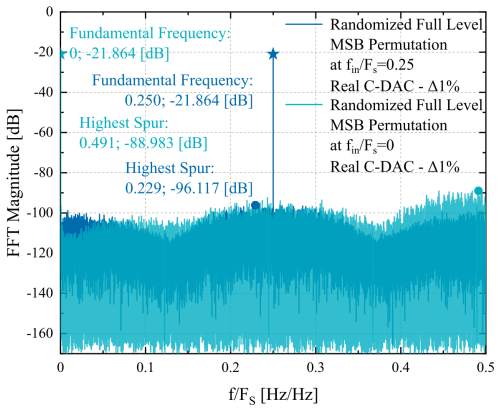


Figure 3.13: Output spectra comparison of an ADC that undergoes the randomized Full resolution level MSB permutation on a 3 bit stage for input frequencies of $f_{in}/F_S = 0$, in light blue, and $f_{in}/F_S = 0.25$, in dark blue.

4 Signal Dependent Data Weight Averaging

From the conceptual point of view, the permutations shown in the previous chapter demonstrates the effects that DEM can have in the cancellation of the mismatch of capacitors. It is possible to establish that these effects are highly dependent on the equations of the converter as well as the input frequency. The reason is, that different ratios $\frac{f_{in}}{F_S}$ produce different patterns of samples such that $V_{in_i} = V_{in_g}$ for $i \neq g$. However, DEM also allows for the implicit realization of filter transfer functions using permutations of the unit capacitors. While signal independent algorithms produce shaping that is dependent on $\frac{f_{in}}{F_S}$, algorithms that take the multiple previous converted samples into account, allow for the transfer function that is generated to act on the mismatch independent of $\frac{f_{in}}{F_S}$. These techniques are conventionally applied to $\Sigma\Delta$ converters with two main implementations, the tree-structure implementation [17, 18, 25], a similar architecture to the one mentioned in section 3.2.2, and the vectorial approach [30–33]. These ways to obtain noise shaping through the permutation of unit elements, generating filters within the Digital to Analog Converter (DAC), are typically used in DACs that are inputted the complete digital word. In a SAR stage/ADC the digital word is not produced in a single step nor is it inputted into the DAC in a single step, it is a progressive calculation of the digital word. This means that the direct implementation of the tree-structure or vector approaches will render unfruitful when trying to extend their usage to SAR topologies, making adaptations to the algorithms necessary. Hence, only recently have been surging implementations of noise shaping into SAR ADCs [34]. A recent study, [35], regarding SAR ADCs was able to implement a vectorial approach to DWA, previously used in $\Sigma\Delta$ modulators [36] and produce a transfer function applied solely to the mismatch of the unit capacitors, however their implementation applies only to single ended ADCs and provides only a low pass operation, effectively eliminating the low frequency components of the mismatch error. This chapter will present an extension of the algorithm for a differential approach and for an arbitrary operation frequency of the ADC, not only reserved for low frequencies, but also high frequencies.

4.1 Algorithm Analysis

The algorithm follows a linear selection of the capacitors, such that it progressively selects the capacitors taking into consideration the previously calculated bit. Since no capacitor is needed in the first bit determination, seen in equation (A.1c), the algorithm starts with no capacitors selected, and only when it is needed to calculate the second bit does the feedback from the output of the SAR comparator realizes the capacitor selection through the DAC drivers, later described in appendix E, (A.2c). For each of the halves of an n_{bits} bit SAR stage, $2^{n_{bits}}$ capacitors are present but only $2^{n_{bits}} - 1$ will be permuted, the remainder will be only needed for the binary weight of the ratio. In the implementation seen in [35] 2 pointers, or decision signals, are needed to describe the algorithm, in this differential approach also only 2 pointers will be needed, a decision signal $d_p(i, j)$ that reacts to $bit(i, j) = 0$ and $d_n(i, j)$ that reacts when $bit(i, j) = 1$, where (i, j) represents the sample number and step tuple. Both decision signals

control both the positive and negative capacitive arrays of the C-DAC, but only do so when they react. The algorithm for a 3 bit SAR stage then follows the following explanation. The decision signals $d_p(i,j)$ and $d_n(i,j)$ initially start with values $2^{n_{bits}}-1$, $d_{p/n}(1,0)=2^{n_{bits}}-1$ where $j=0$ represents the sampling step (the boundary condition). Supposing that over 3 conversions of the input signal, the complete digital word output would follow table 4.1, the algorithm can be described.

Table 4.1: Output per step of a 3 bit SAR stage over 3 full conversions.

		Output i		
		1	2	3
Step	$bit(i,1)=b_{2_i}$	0	0	1
	$bit(i,2)=b_{1_i}$	1	0	0
	$bit(i,3)=b_{0_i}$	0	1	1

To facilitate in the mathematical description that will be presented in further sections, $bit(i,1)$ represents the first bit that is calculated by the ADC, b_{2_i}, the second bit, b_{1_i} will take the form of $bit(i,2)$ and the third bit, b_{0_i} will take the form of $bit(i,0)$. Hence, the first step determines the MSB that in turn selects the first 4 capacitors. Initially, in the positive half of the C-DAC, the decision signal has the value of 7 as illustrated by figure 4.1.

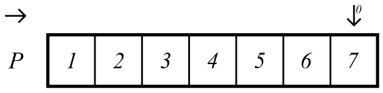


Figure 4.1: Array diagram of the capacitors from the positive half of the C-DAC in a 3 bit resolution SAR stage.

In figure 4.1 the arrow on top of "P" represents the orientation of the cyclic selection, from left to right, and the arrow pointing to the number 7 represents the value of the decision signal at step 0. Following table 4.1, since the selection is cyclic and the number of capacitors that should be selected is 4, then the decision signal updates its value to 4 (at step 1), and the selected capacitors are $(C_1, C_2, C_3, C_4)_p$ of weights $(w_1, w_2, w_3, w_4)_p$, respectively.

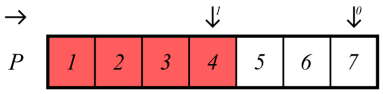


Figure 4.2: First selection of positive capacitors for sample 1: Sample 1, step 1.

These capacitors connect to V_{Ref}, however on the negative half of the C-DAC, there are also capacitors that need to connect to gnd, the algorithm, as explained earlier, allows the reacted decision signal to select capacitors from both the positive and negative halves. On the negative half, to be connected to gnd, d_p selects capacitors $(C_7, C_6, C_5, C_4)_n$ of weights $(w_7, w_6, w_5, w_4)_n$, respectively and the negative decision signal, d_n, does not change its value, as illustrated by figure 4.3.

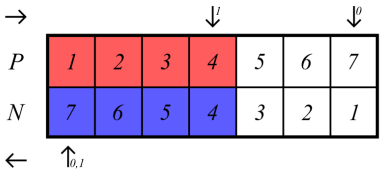


Figure 4.3: First selection of capacitors for sample 1: Sample 1, step 1.

In figure 4.3 the upper and lower sections represent the positive and negative halves of the C-DAC, with each of the number representing the index of capacitor $C_{k_{p/n}}$ of weight $(w_k)_{p/n}$. Red colored capacitors are connected to V_{Ref} and blue colored capacitors are connected to gnd. In the next step, the MSB-1 is determined and 2 other pairs of capacitors must be selected. $bit(1,2) = 1$ therefore the negative decision signal is updated. Because the selection of capacitors follows the same cyclical pattern in both the positive and negative halves, d_p remains the same value while d_n is updated from 7 to 2, selecting capacitors $(C_1, C_2)_n$ on the negative half to connect to V_{Ref}, as well as capacitors $(C_7, C_6)_p$, on the positive half to connect to gnd, illustrated by figure 4.4.

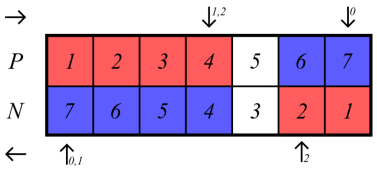


Figure 4.4: Second selection of capacitors for sample 1: Sample 1, step 2.

The last step of the conversion determines the third bit, that allows for the selection of the last pair of capacitors. On table 4.1 the last bit, $bit(1,3) = 0$ therefore the positive decision signal is updated while, the negative decision signal is not. The last capacitors selected are $(C_5)_p$ to V_{Ref} and $(C_2)_n$ to gnd as illustrated in figure 4.5.

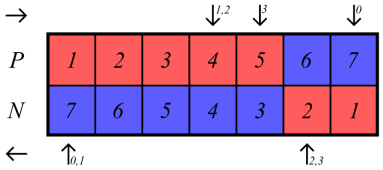


Figure 4.5: Third selection of capacitors for sample 1: Sample 1, step 3.

The next sample conversion follows the second row of table 4.1 whose digital word is $(0,0,1)$ and the algorithm follows the illustration in figure 4.6.

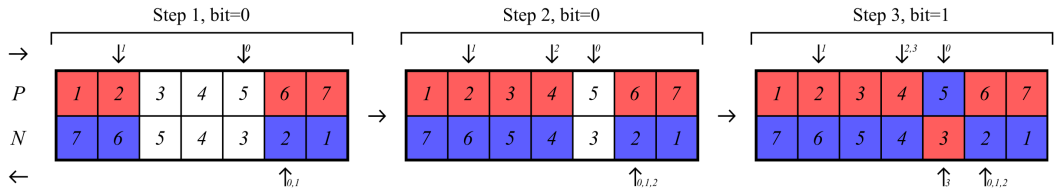


Figure 4.6: Progressive selection of capacitors for sample 2.

In step 1 of the second sample, the decision signal starts where the last conversion finished so that $d_{p/n}(i,0) = d_{p/n}(i-1, n_{bits})$. Furthermore, $d_p(1,3) = 5$ and is updated from $d_p(2,0) = d_p(1,3) = 5$ to $d_p(2,2) = 2$, this is because after reaching index 7 the next index is 1, due to the cyclic nature of the algorithm. In appendix D a mathematical representation of this effect is described through an overflow signal, equivalent to the carry term in digital addition blocks. For completion, the last conversion outputs the digital word $(1,0,1)$ and the algorithm follows the illustration in figure 4.6.

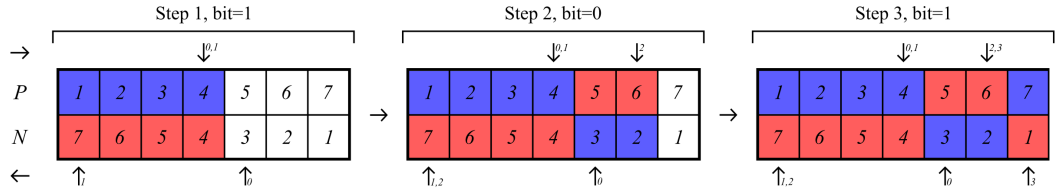


Figure 4.7: Progressive selection of capacitors for sample 3.

By describing the selected capacitors as the combination of an ideal and mismatched parts, as set forth by equations (4.1), then, utilizing the description given previously, it is possible to derive the mathematical equations that govern this DEM algorithm, seen in appendix D.1.

$$(w_k)_{p/n} = (w_{mean})_{p/n} + (e_k)_{p/n} \tag{4.1a}$$

$$(w_{mean})_{p/n} = \frac{1}{N-1}\sum_{k=1}^{N-1}(w_k)_{p/n}, \quad N = 2^{n_{bits}} \tag{4.1b}$$

The final equation pertaining to the mismatch term of the C-DAC residue voltage sees the contribution of a term AME, that represents the accumulated mismatch error across a full conversion, seen in equation (D.19c). Utilizing this description of the mismatch voltage its frequency response can be analyzed by performing the z-transform, seen in equation (4.2).

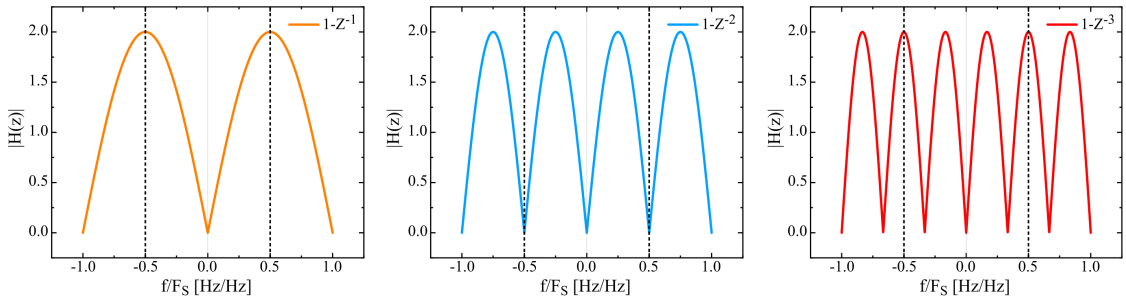
$$\begin{aligned}
&\mathcal{Z}\{y_{DAC\,Mismatch}(i)\} = Y_{DAC\,Mismatch}(z) = \\
&\underbrace{\frac{1-Z^{-1}}{N}}_{\text{Explicit Filtering}} \left[\begin{array}{l} VCM\left[\dfrac{AME_{V_{Ref}\,p}(z) + AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z) + AME_{gnd\,n}(z) -}{(w_{mean})_n}\right] \\ \qquad - V_{Ref}\left[\dfrac{AME_{V_{Ref}\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z)}{(w_{mean})_n}\right] \\ \qquad - gnd\left[\dfrac{AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{gnd\,n}(z)}{(w_{mean})_n}\right] \end{array} \right] \Rightarrow \\
&Y_{DAC\,Mismatch}(z) = \underbrace{\left[1-Z^{-1}\right]}_{H(z)} \frac{TME(z)}{N}
\end{aligned} \tag{4.2}$$

Equation (4.2) demonstrates that the algorithm in fact performs an explicit filtering on the mismatch effects, the total mismatch error, TME. This filter affects TME independently of

the frequency at which the ADC is functioning, given that the algorithm depends not on a predetermined pattern, but on a selection that is dependent on the conversion steps. In other words, the algorithm is dependent on the input and therefore dependent on the ratio $\frac{f_{in}}{F_S}$ which allows the algorithm to filter the mismatch errors whether or not the ADC is working at different $\frac{f_{in}}{F_S}$.

4.2 Frequency Domain Contraction

Despite filtering the mismatch regardless of the $\frac{f_{in}}{F_S}$ at which the ADC operates, the algorithm, as it stands, produces a filter that has a high-pass configuration, filtering low frequency components and allowing high frequency components to persevere. This is evident by the zero of the filter's transfer function $\left|1-Z^{-1}\right|=0 \Rightarrow \frac{f_{in}}{F_S}=0$, this result reveals that the most efficient frequency operation of the ADC, in terms of mismatch elimination is 0, or from an academical point of view: $f_{in} \in \mathbb{R}\backslash\{0\} \wedge F_S \to \infty \vee f_{in}=0 \wedge F_S \in \mathbb{R}\backslash\{0\}$. However, given the repetition profile of $H(z)$ in the frequency domain $\left(\frac{f}{F_S} \in \mathbb{R}\right)$ as well as the results mentioned in section 3.2, the frequency at which the filtering is most effective can be altered. As with the bit independent permutations, offsets can be added to the sequence, or algorithms in this case, and effectively contract the frequency axis to allow more zeros from unseen repetitions come to the usable spectrum $\frac{f}{F_S} \in \left[0;\frac{1}{2}\right]$. The magnitude of the filter, $|H(z)|=\left|1-Z^{-1}\right|=2\sqrt{\sin^2\left(\frac{\pi f}{F_S}\right)}$ has zeros of algebraic multiplicity 2 at frequencies $\frac{f}{F_S}=0+m, \forall m \in \mathbb{Z}$ and poles of algebraic multiplicity 2 at frequencies $\frac{f}{F_S}=\frac{1}{2}+m, \forall m \in \mathbb{Z}$, this means that the pair of zeros that are within the usable spectra $\frac{f}{F_S} \in \left[0;\frac{1}{2}\right]$ are only at $\frac{f}{F_S}=0$. However, if the transfer function would have become $H(z)=1-Z^{-a}$, whose magnitude is $|H(z)|=\left|1-Z^{-1}\right|=2\sqrt{\sin^2\left(\frac{a\pi f}{F_S}\right)}$, then the number of zeros within the usable frequency range is greater as a increases at frequencies $\frac{f}{F_S}=\frac{m}{a}, \forall m \in \mathbb{Z}$. Figure 4.8 shows the intuition behind this effect.



(a) $a=1$ **(b)** $a=2$ **(c)** $a=3$

Figure 4.8: Magnitude comparison between filter transfer functions with varying a.

To obtain a filter that has this behaviour, it is necessary to slightly modify the algorithm by incorporating an offset to it. Previously, at the beginning of each conversion the algorithm refers to the value of both decision signals from the last step of the previous conversion , this referral leads to the persistence of terms containing index $(i-1, n_{bits})$ or simply $(i-1)$. This is

most notably evidenced in equations (D.15) to (D.17), in appendix D.1, which describes how the mismatch term of the accumulated selected capacitors, ASC, can be described only by the initial and final terms of the AME in a given step of the conversion. If the referencing mechanism at the beginning of the algorithm were to reference the last step of a conversions prior, then the persistent index in all equations would be $(i-a)$ instead of $(i-1)$. Where before the mismatch term could be described by equation (D.19c), it could now be described by equation (D.21) in appendix D.2. Therefore the mismatch term would be filtered by the more general discussed filter of $H(z) = 1 - z^{-a}$, evidenced in equation (4.3).

$$
\begin{aligned}
&\mathcal{Z}\{y_{DAC\,Mismatch}(i)\} = Y_{DAC\,Mismatch}(z) = \\
&\underbrace{\frac{1-Z^{-a}}{N}}_{\text{Explicit Filtering}} \left[\begin{array}{l} VCM\left[\dfrac{AME_{V_{Ref}\,p}(z) + AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z) + AME_{gnd\,n}(z) -}{(w_{mean})_n}\right] \\ \qquad - V_{Ref}\left[\dfrac{AME_{V_{Ref}\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z)}{(w_{mean})_n}\right] \\ \qquad - gnd\left[\dfrac{AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{gnd\,n}(z)}{(w_{mean})_n}\right] \end{array} \right] \Rightarrow \\
&Y_{DAC\,Mismatch}(z) = \underbrace{\left[1 - Z^{-a}\right]}_{H(z)} \frac{TME(z)}{N}
\end{aligned}
\tag{4.3}
$$

While there were alterations in the algorithm in between conversions, seen by equations in appendix D.2, the mechanism that governs the value of the decision signals and the respective capacitor selection within a given conversion must maintain the same. Determining the value of a is of preference of the designer however, it should be noted that to obtain a band pass operation of the ADC, $\frac{f_{in}}{F_S} \neq \{0; \frac{1}{2}\}$ the value of a should be equal or greater than 3, $a \geq 3$. As previously mentioned incorporating an offset of a between conversions is equivalent to contracting the frequency axis, symbolized by the argument of the sine function in $|H(z)| = |1 - Z^{-a}| = 2\sqrt{\sin^2\left(\frac{a\pi f}{F_S}\right)}$. As the value of a increases, so does the number of crests present in the interval of $\frac{f}{F_S} \in \left[0; \frac{1}{2}\right]$, in total, there will be $\frac{a}{2}$ crests. This means that the spacing between 2 consecutive crests will be smaller, and so will the range of frequencies that the mismatch is appreciably eliminated, as seen in figure 4.9. For this reason $a = 3$ should be used whenever a band-pass utilization of the ADC is desired.

Another effect necessary to take notice is that although the mismatch is filtered, the mismatch is not ideally random, nor can it be affected by the dithering mechanisms detailed in earlier sections under the penalty of not causing filter in the first place. $TME(z)$ depends mainly on the fact that the capacitors, and their mismatched capacitance, are cycled through, giving rise to a limited number of different values of $TME(i)$, hence this cyclicity opens the possibility for discrete spurs to be present in the usable frequency range. To better understand the frequency response of $TME(z)$ the following example is useful: Supposing that a ratio of

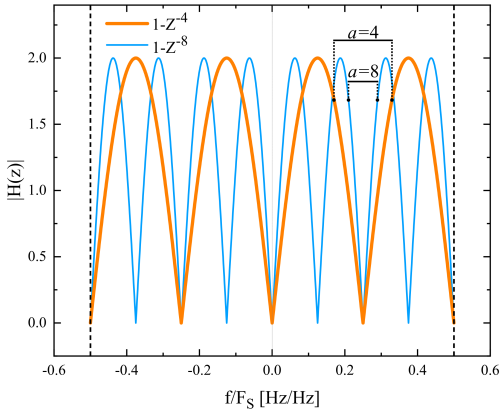
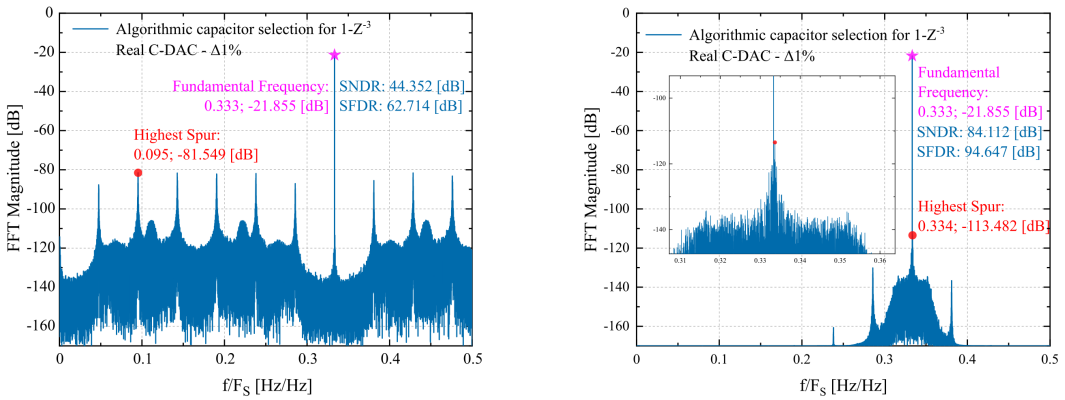


Figure 4.9: Bandwidth comparison between filter transfer functions of different values of a

$\frac{f_{in}}{F_S} = \frac{1}{3}$ is used and that the value of $a = 3$, then at this setting only 3 samples will be converted, and utilizing the algorithm offset of 3, samples $3i-2$, $3i-1$, $3i$ will have the decision signals $d_{p/n}(3i-2,0)$, $d_{p/n}(3i-1,0)$, $d_{p/n}(3i,0)$ refer to last conversion step of the 3 conversions prior ,$d_{p/n}(3i-5,n_{bits})$, $d_{p/n}(3i-4,n_{bits})$, $d_{p/n}(3i-3,n_{bits})$, consequently of the same sample. This means that the number of selected capacitors, from either the positive or negative halves of the C-DAC, to connect to either V_{Ref} or gnd every 3 samples will be the same. Given that an array of capacitors is $2^{n_{bits}}-1$ long, then over $(a=3)\cdot(2^{n_{bits}}-1)$ samples each different sample will appear $2^{n_{bits}}-1$ times, and the decision signals, $d_{p/n}\left((a=3)i\cdot(2^{n_{bits}}-1)-2,0\right)$, $d_{p/n}\left((a=3)i\cdot(2^{n_{bits}}-1)-1,0\right)$, $d_{p/n}\left((a=3)i\cdot(2^{n_{bits}}-1),0\right)$, will become the same as the initial conditions, meaning that only $2^{n_{bits}}-1$ total values for $TME(i)$ will become available for each sample. In total, $(a=3)\cdot(2^{n_{bits}}-1)$ different deviations of the ideal value for the C-DAC voltage will be found. Because these errors are encountered in an cyclical fashion as the ADC operates, in the frequency domain, these errors will appear as $(a=3)\cdot(2^{n_{bits}}-1)$ equally spaced odd-type spurs between $\frac{f}{F_S}\in[0;1]$, at frequencies $\left(\frac{f}{F_S}\right)_s = \frac{s}{a\cdot(2^{n_{bits}}-1)}$, $s\in\mathbf{Z}$, including the zeros of the filter transfer function $H(z)=1-Z^{-3}$ where the fundamental frequency is. Figure 4.10 illustrates the output frequency spectra for a simulated ADC, with the first stage of 3 bits being the one that undergoes the algorithmic selection of capacitors with mismatch of 1%, while the remaining stages of 6 bit resolution each are treated as ideal, with no mismatch.

It is possible to observe both the predicted spurs as well as the filtering effect of $TME(z)$ caused by transfer function $H(z)=1-Z^{-3}$. The filtering only affects the mismatch term $TME(z)$, leaving the ideal term untouched as predicted by equations (D.19a) (4.3). It is important to note that although the filtering effect is independent of the input frequency, the frequency response of $TME(i)$ is not, given that different errors will be encountered if the sampling follows regular patterns such as when $\frac{f_{in}}{F_S}=\frac{1}{3}$, or not when for example $\frac{f_{in}}{F_S}=\frac{1}{3}+\Delta_{f/F_S}$. For real world applications, the frequency input $\frac{f_{in}}{F_S}$ can vary, and if it does then the $(a=3)\cdot(2^{n_{bits}}-1)$ error cycle is broken, and the number of possible values for the $TME(i)$ increases, and with it so does the number of spurs. Although the number of possible errors is deterministic in nature, to know the number itself is not important, but rather the effect that it has on the frequency

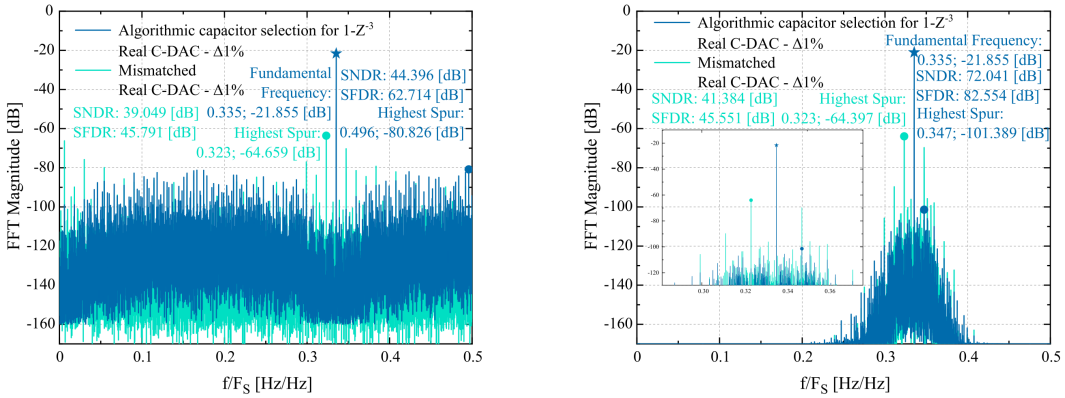


(a) Unfiltered output spectra.

(b) Filtered output spectra.

Figure 4.10: Output spectra of an ADC that undergoes the algorithm, with $a = 3$, on the first stage of resolution 3 bits. This case represents the run out of 10000 MC runs that rendered the lowest SNDR, with an input frequency of 1/3.

response of the system. As explained, the number of spurs increases however, because there are more values of TME that the C-DAC can cycle through the intensity of the spurs decreases appreciably. Figure 4.11 demonstrates this effect and compares it to a C-DAC with no capacitor permutation technique.

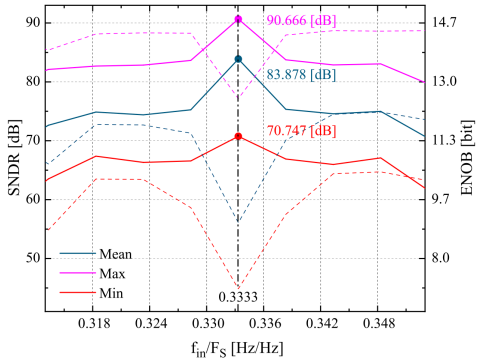


(a) Unfiltered output spectra.

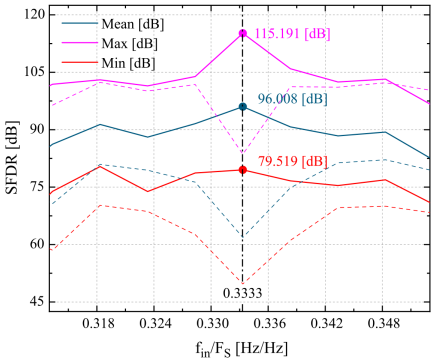
(b) Filtered output spectra.

Figure 4.11: Output spectra of an ADC that undergoes the algorithm, with $a = 3$, on the first stage of resolution 3 bits. This case represents the run out of 10000 MC runs that rendered the lowest SNDR, with an input frequency of approximately 1/3.

A further deviation of the input frequency will make the ADC not as tuned in with the filtering that occurs through the cyclic selection of capacitors meaning that the performance of the ADC is prone to decrease. Figure 4.12 demonstrates the SNDR and SFDR of the ADC whose output is filtered near the fundamental frequency with a 6th order band pass digital filter of $BW_{OutFilter} = 0.04$ centered at $\frac{f}{F_S} = \frac{1}{3}$ as $\frac{f_{in}}{F_S}$ changes. It is possible to see that near $\frac{f_{in}}{F_S} = \frac{1}{3}$, there is an increase of performance due to the decrease in the mismatch effect, and as the input frequency strays away from the zero of the transfer function, the performance of the ADC as a whole decreases.



(a) SNDR of the ADC.

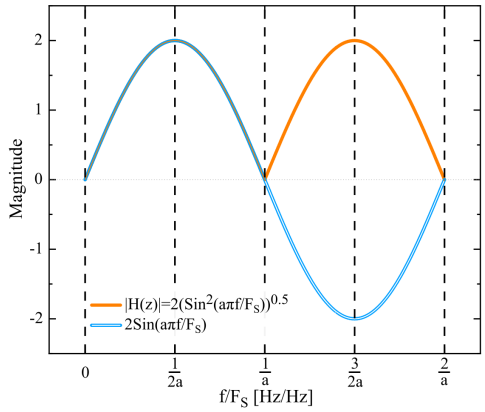


(b) SFDR of the ADC.

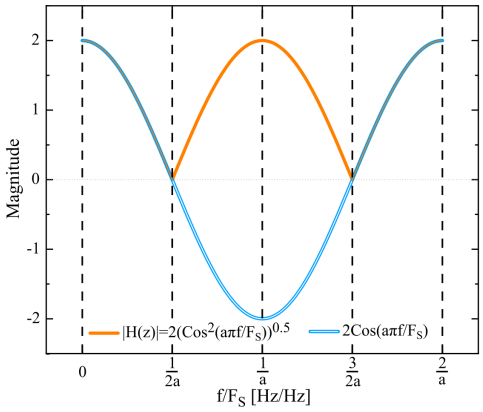
Figure 4.12: Performance of an ADC that undergoes the algorithm, with $a = 3$, on the first stage of resolution 3 bits as the input frequency changes. The dashed lines shows the results for the ADC without the algorithm. The data were obtained by running 400 MC runs for each frequency swept.

4.3 Switching Sequences to obtain other Transfer functions

Mathematically, it has been proved that the algorithm described can in fact produce an explicit filter affecting only the mismatch errors of the C-DAC, however, the designed filter has the factor a to optimize. For the given application, a band-pass operating ADC, the best possible value of a is $a = 3$ given that it is the value that allows for the highest bandwidth. However, given the family of the transfer functions, $H(z) = 1 - Z^{-a}$, and remembering that the mathematical representation of the magnitude has a sinusoidal argument, changing it to a cosine derived magnitude, $1 + Z^{-a}$, would prove a selection of $a = 2$ successful at implementing a band-pass operation ADC with the added benefit of having larger bandwidth than with $a = 3$. Figure 4.13a shows the sinusoidal shape of the magnitude of the current transfer function, and 4.13b shows the cosine shaped magnitude of the desired transfer function for the mismatch filter.



(a) Odd Type filtering $1 - Z^{-a}$.

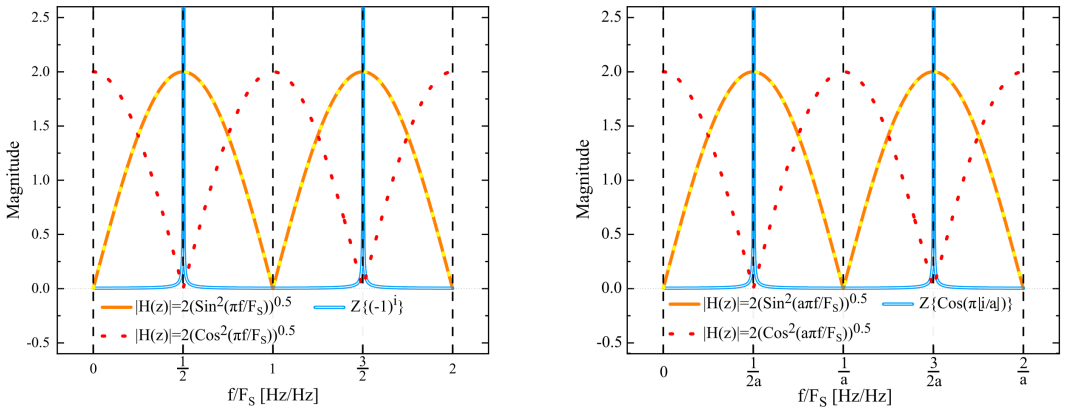


(b) Even Type filtering $1 + Z^{-a}$.

Figure 4.13: Difference between a transfer function of a sine derived nature, odd, and a cosine derived nature, even.

To transition from a sine wave to it's cosine counter part an offset of $\frac{\pi}{2}$ must be generated. This offset can be thought of as the convolution of the current transfer function with a function

of equally spaced, repeating impulses in the frequency domain at the maxima of the current transfer function. The result will be that the maxima of the previous transfer function will become the minima of the new transfer function. For the case were the filter transfer function is $H(z) = 1 - Z^{-1}$, then the convolution of the spectrum of $\mathcal{Z}\{(-1)^i\} = \frac{z}{z+1}$, depicted in figure 4.14a, will result in $H(z) = 1 + Z^{-1}$.



(a) $a = 1$ example. **(b)** General value of a.

Figure 4.14: Graphical representation of the spectral convolution between the filter transfer function and the function of equally spaced and repeating impulses.

From the practical point of view, performing these convolutions in the frequency regime, means the multiplication in the time domain expressed in equation (4.4a).

$$TME(i) \cdot (-1)^i \tag{4.4a}$$

$$TME(i) \cdot SS(i) \tag{4.4b}$$

Equation (4.4b) is a generalization of equation (4.4a), where $SS(i)$ defines switching sequence. This multiplication can be seen as the orientation of the decision signal's progression. Previously only one orientation was regarded, a progression in which it increased the value of the decision signal, now translating to $SS(i) = 1$, if rather $SS(i) = -1$, then the progression of the decision signal's value is to decrease. To illustrate the algorithm, a diagram, in figure 4.15 representative of a stage of 3 bits, similar to that of figure 4.7 is used to represent the progression of the decision signals as well as the selected capacitors responding to the outputs presented in table 4.1, earlier. In this demonstration the switching sequence will take values -1, 1 and -1.

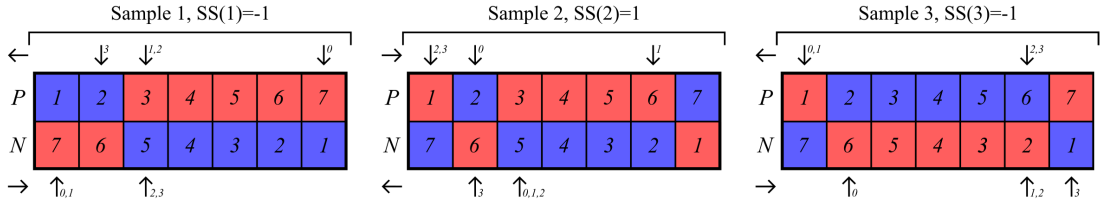


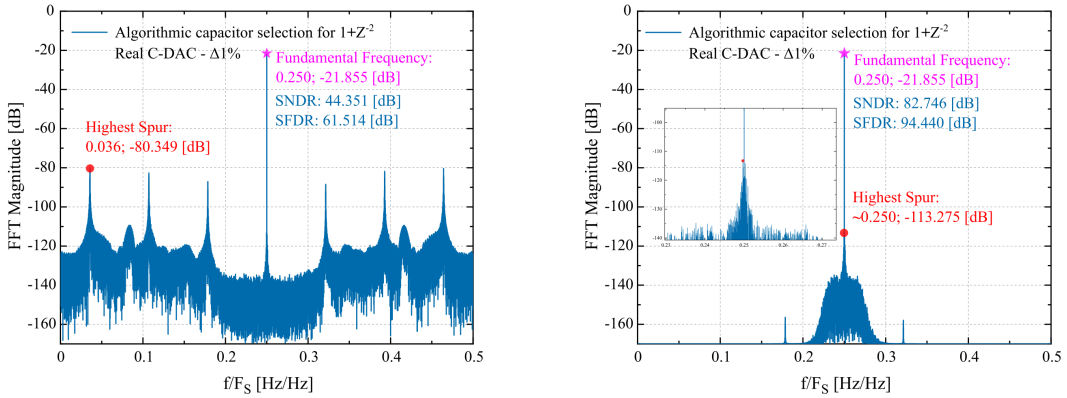
Figure 4.15: Progressive selection of capacitors over the course of 3 samples.

The changes in the algorithm conduces to the changes in the equations of appendix D.1 to equations of appendix D.3, specifically, from equation (D.3) to equation (D.24), for the decision signal with $SS(i) = -1$, and equation (D.5) to equation (D.25) for the overflow term.

The equations set in appendix D.3 prove that the change in the algorithm does in fact perform the convolution theorized earlier, seen in equation (4.5).

$$\begin{aligned}\mathcal{Z}\{y_{DAC\,Mismatch}(i)\} &= \frac{1}{N}\mathcal{Z}\{SS(i)\cdot TME(i)\} \Rightarrow \\ &\Rightarrow \frac{1}{N}\mathcal{Z}\{SS(i)\} \circledast \mathcal{Z}\{TME(i)\} \Rightarrow \\ &\Rightarrow \frac{TME(z)}{N}\left[\mathcal{Z}\{SS(i)\} \circledast \left[1 - Z^{-a}\right]\right]\end{aligned} \tag{4.5}$$

This alteration allows the utilization of a lower a. It should be noted that although the demonstration had the purpose of utilizing $SS(i) = \cos\left(\pi \left\lfloor \frac{i}{a} \right\rfloor\right)$, since a general approach was used, $SS(i)$ can be an arbitrary binary sequence, that if designed correctly can create new transfer functions from $1 - Z^{-a}$. The design of binary sequences with custom characteristics can become a Non-deterministic Polinomia-time Hard (NP-Hard) problem [37, 38], therefore the presented work will not pursue better transfer functions than those already obtained. Since in equation (4.5) the convolution does not affect the term $TME(z)$, then its characteristics can be derived in the same way as without the convolution. Because $a = 2$ then the C-DAC will have $(a = 2)\cdot\left(2^{n_{bits}=3} - 1\right) = 14$ values of $TME(i)$ causing $(a = 2)\cdot\left(2^{n_{bits}=3} - 1\right) = 14$ equally spaced spurs between $\frac{f}{F_S} \in [0;1]$, at frequencies $\left(\frac{f}{F_S}\right)_s = \frac{s}{a\cdot(2^{n_{bits}}-1)}$, $s \in \mathbf{Z}$, including the spur that coincides with the input frequency at $\frac{f_{in}}{F_S} = \frac{1}{4}$. Figure 4.16 shows the resultant output frequency spectra for a simulated ADC with the first stage of 3 bits, being the one that undergoes the even-type (with $a = 2$) algorithmic selection of capacitors. The mismatch of the capacitors are 1% while the remaining stages are treated as ideal, with no mismatch.



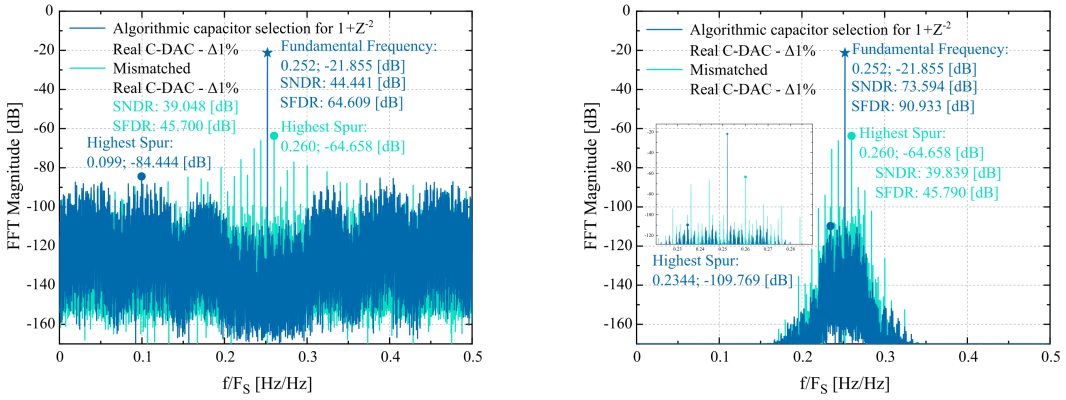
(a) Unfiltered output spectra.

(b) Filtered output spectra.

Figure 4.16: Output spectra of an ADC that undergoes the Even-type algorithm, with $a = 2$, on the first stage of resolution 3 bits. This case represents the run out of 10000 MC runs that rendered the lowest SNDR, with an input frequency of 1/2.

As with the case of the odd-type algorithm, if the frequency deviates from the ideal, the pronounced spurs fade as more values of $TME(i)$ are cycled through giving the spectrum seen

in figure 4.17.

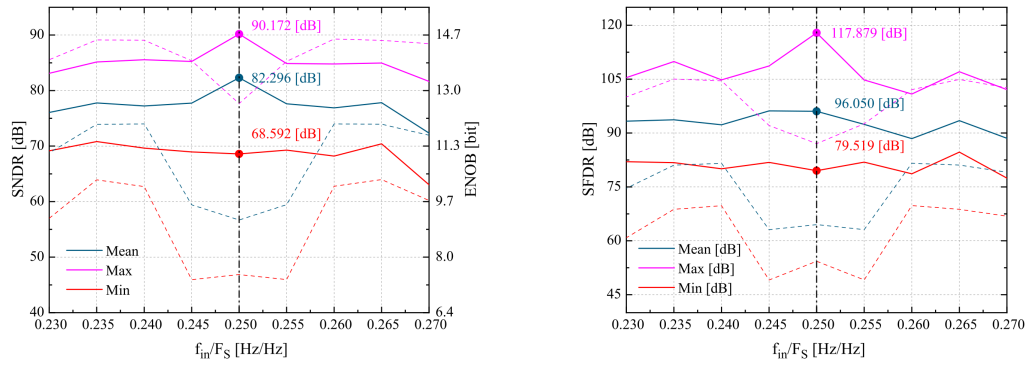


(a) Unfiltered output spectra.

(b) Filtered output spectra.

Figure 4.17: Output spectra of an ADC that undergoes the even-type algorithm, with $a = 2$, on the first stage of resolution 3 bits. This case represents the run out of 10000 MC runs that rendered the lowest SNDR, with an input frequency of approximately $1/4$.

If the frequency deviation is too great, then the performance of the ADC decreases significantly. To visualize this effect, the same frequency sweep performed for the testing of the Odd-type algorithm is conducted for the Even-type algorithm rendering the results seen in figure 4.18.



(a) SNDR of the ADC.

(b) SFDR of the ADC.

Figure 4.18: Performance of an ADC that undergoes the Even-type algorithm, with $a = 2$, on the first stage of resolution 3 bits as the input frequency changes. The dashed lines shows the results for the ADC without the algorithm. The data were obtained by running 400 MC runs for each frequency swept.

With the alterations to the algorithm mentioned, and its implementation in the SAR architecture, it is possible to obtain a DEM algorithm that allows for the mismatch error frequency components to be filtered out by the general filter of $H(z) = 1 \pm Z^{-a}$. In both the Even and Odd algorithms the performance is kept high in a interval of 0.03 around the zero of the filter transfer function, which corresponds to a relatively low OSR of 16.7, [33]. This DEM algorithm has a relatively simple digital implementation introduced in appendix E that coupled with the modular structure of the pipeline can produce the relevant increase of performance in figures 4.12a or 4.18 in comparison with the mismatched C-DAC seen in figures 4.11 and 4.17.

5 Final Observations and Conclusions

In sum, the presented work had the aim of introducing some dynamic element matching techniques, as well as expanding the algorithmic selection of capacitors, utilizing the basis of data weight averaging, into SAR ADCs not only in a low-pass operation but a band-pass operation that effectively filters mismatch derived spurious tones in the output of the ADC.

The work starts by introducing the general equations for the sub-SAR ADC, in chapter 2, as well as showing the adverse effect that capacitor mismatch can have in the output frequency spectra of the combined ADC in section 2.2.

The sampling frequency dependent DEM algorithms shown thereafter, in chapter 3, allow for the acquaintance of DEM fundamentals that introduce the topic of dynamic element matching in general. The most simple case, where the capacitors are randomly selected performing a dithering effect on the error produced in the C-DAC voltage. That in turn augments the spurious free dynamic range with an observable increase of the noise floor, in section 3.1. Followed by more complex permutations that allows for the explicit cancellation of the unique or several errors referring to different rations, in section 3.2.

Utilizing the fundamentals of DEM as well as the measurements developed in section 3.2, an extension to the DWA algorithm, a vectorial approach to signal dependent DEM typically used in $\Sigma\Delta$ converters, was performed to encompass its implementation onto the SAR ADC. Sections 4.1, 4.2 and 4.3 explained the algorithm as well as the mathematical equations to obtain the general mismatch error filter of $H(z) = 1 \pm Z^{-a}$. The algorithm applied to SAR ADCs presents excellent results in the mitigation of the mismatch error derived spurs, by filtering the output spectra near the signal frequency, achieving levels of SFDR of 117.9 [dB] and ENOB of 14.8 [bit], with capacitor mismatch errors of 1%.

In both even and odd implementations of the algorithm, the performance is kept high in 0.03 interval around the zeros of the mismatch filter transfer function corresponding to the relatively low OSR of 16.7. The digital implementation of the algorithm explained in appendix E, is able to take advantage of the modular structure of the pipeline ADC, given that its complexity only grows multiplicatively with the number of stages whose resolution can be set low to allow for the most efficient area possible.

Future work should present a study concerning the selection of a pipeline structure, that takes advantage of the last DEM technique shown and its digital implementation. The study should evaluate the necessity of utilizing the DEM technique on more than one stage, in terms of spurious mitigation, and both advantages and disadvantages, in terms of area used for digital implementation, while considering a real ADC where all stages contain capacitors with different mismatch errors.

Overall, the aim of this work was successfully achieved given that an implementation of the DWA algorithm for the SAR ADC was adequately performed, allowing a general operation of the ADC, including a band-pass operation.

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. Murmann. "The race for the extra decibel: A brief review of current ADC performance trajectories". In: *IEEE Solid-State Circuits Magazine* 7.3 (2015), pp. 58–66 (cit. on pp. 1, 2).

[3] D. A. Johns and K. Martin. *Analog integrated circuit design*. John Wiley & Sons, 2008 (cit. on p. 1).

[4] B. Razavi. *Design of analog CMOS integrated circuits*. 2005 (cit. on p. 1).

[5] B. Murmann. *ADC Performance Survey 1997-2023*. [Online]. Available: https://github.com/bmurmann/ADC-survey (cit. on p. 1).

[6] B. Razavi. "A tale of two ADCs: Pipelined versus SAR". In: *IEEE Solid-State Circuits Magazine* 7.3 (2015), pp. 38–46 (cit. on p. 1).

[7] J. Lagos et al. "A single-channel, 600-MS/s, 12-b, ringamp-based pipelined ADC in 28-nm CMOS". In: *IEEE Journal of Solid-State Circuits* 54.2 (2018), pp. 403–416 (cit. on p. 1).

[8] T. Iizuka, H. Xu, and A. A. Abidi. "A tutorial on systematic design of CMOS A/D converters: Illustrated by a 10 b, 500 MS/s SAR ADC with 2 GHz RBW". In: *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)*. IEEE. 2021, pp. 381–386 (cit. on p. 1).

[9] Y.-C. Kwon and O.-K. Kwon. "A precision mismatch measurement technique for integrated capacitor array using a switched capacitor amplifier". In: *IEEE transactions on semiconductor manufacturing* 26.2 (2013), pp. 226–232 (cit. on pp. 1, 2, 4).

[10] H. Omran et al. "Direct mismatch characterization of femtofarad capacitors". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 63.2 (2015), pp. 151–155 (cit. on pp. 1, 2, 4).

[11] I. Galton. "Why dynamic-element-matching DACs work". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 57.2 (2010), pp. 69–74 (cit. on p. 2).

[12] K. D. Choo et al. "Energy-efficient motion-triggered IoT CMOS image sensor with capacitor array-assisted charge-injection SAR ADC". In: *IEEE Journal of Solid-State Circuits* 54.11 (2019), pp. 2921–2931 (cit. on p. 2).

[13] Y. Tsukamoto et al. "High power efficient and scalable noise-shaping SAR ADC for IoT sensors". In: *2016 IEEE International Meeting for Future of Electron Devices, Kansai (IMFEDK)*. IEEE. 2016, pp. 1–2 (cit. on p. 2).

[14] K. Seong et al. "Time-interleaved SAR ADC with background timing-skew calibration for UWB wireless communication in IoT systems". In: *Sensors* 20.8 (2020), p. 2430 (cit. on p. 2).

[15] Y.-H. Hwang et al. “A 0.6-to-1V 10k-to-100kHz BW 11.7 b-ENOB Noise-Shaping SAR ADC for IoT sensor applications in 28-nm CMOS”. In: *2018 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE. 2018, pp. 247–248 (cit. on p. 2).

[16] J.-E. Park, Y.-H. Hwang, and D.-K. Jeong. “A 0.4-to-1 V Voltage Scalable ΔΣ ADC With Two-Step Hybrid Integrator for IoT Sensor Applications in 65-nm LP CMOS”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.12 (2017), pp. 1417–1421 (cit. on p. 2).

[17] E. Fogelman et al. “A 3.3-V single-poly CMOS audio ADC delta-sigma modulator with 98-dB peak SINAD and 105-dB peak SFDR”. In: *IEEE Journal of solid-state circuits* 35.3 (2000), pp. 297–307 (cit. on pp. 2, 15, 18).

[18] I. Galton. “Spectral shaping of circuit errors in digital-to-analog converters”. In: *IEEE Transactions on Circuits and Systems II: Analog and digital signal processing* 44.10 (1997), pp. 808–817 (cit. on pp. 2, 15, 18).

[19] E. Fogleman and I. Galton. “A dynamic element matching technique for reduced-distortion multibit quantization in delta-sigma ADCs”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 48.2 (2001), pp. 158–170 (cit. on p. 8).

[20] M. Bolatkale et al. “A 28-nm 6-GHz 2-bit continuous-time ΔΣ ADC with- 101-dBc THD and 120-MHz bandwidth using blind digital DAC error correction”. In: *IEEE Journal of Solid-State Circuits* 57.12 (2022), pp. 3768–3780 (cit. on p. 8).

[21] K. Lee, Y. Yoon, and N. Sun. “A 10MHz-BW, 5.6 mW, 70dB SNDR ΔΣ ADC using VCO-based integrators with intrinsic DEM”. In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2013, pp. 2006–2009 (cit. on p. 8).

[22] B. Olleta et al. “A deterministic dynamic element matching approach to ADC testing”. In: *2003 IEEE International Symposium on Circuits and Systems (ISCAS)*. Vol. 5. IEEE. 2003, pp. V–V (cit. on p. 8).

[23] W. Guo and N. Sun. “A 12b-ENOB 61μW noise-shaping SAR ADC with a passive integrator”. In: *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE. 2016, pp. 405–408 (cit. on p. 13).

[24] K. Hasebe et al. “A 100kHz-bandwidth 98.3 dB-SNDR noise-shaping SAR ADC with improved mismatch error shaping and speed-up techniques”. In: *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE. 2022, pp. 56–57 (cit. on p. 13).

[25] H.-Y. Hsieh and L. Lin. “A first-order tree-structured DAC with reduced signal-band noise”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 54.5 (2007), pp. 392–396 (cit. on pp. 13, 15, 18).

[26] Y.-H. Chung and S.-Y. Shih. “A 10-bit 100-MS/s SAR ADC with Capacitor Swapping Technique in 90-nm CMOS”. In: *2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*. IEEE. 2017, pp. 1–4 (cit. on pp. 13, 15).

[27] C.-C. Liu et al. “A 10-bit 50-MS/s SAR ADC With a Monotonic Capacitor Switching Procedure”. In: *IEEE Journal of Solid-State Circuits* 45.4 (2010), pp. 731–740. DOI: 10.1109/JSSC.2010.2042254 (cit. on p. 13).

[28] Y.-H. Chung, M.-H. Wu, and H.-S. Li. “A 12-bit 8.47-fJ/conversion-step capacitor-swapping SAR ADC in 110-nm CMOS”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.1 (2014), pp. 10–18 (cit. on pp. 13, 15).

[29] B. H. Leung and S. Sutarja. “Multibit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39.1 (1992), pp. 35–51 (cit. on p. 15).

[30] N. Sun. “High-order mismatch-shaping in multibit DACs”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 58.6 (2011), pp. 346–350 (cit. on p. 18).

[31] N. Sun. “High-Order Mismatch-Shaped Segmented Multibit ΔΣ DACs With Arbitrary Unit Weights”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 59.2 (2011), pp. 295–304 (cit. on p. 18).

[32] R. Henderson and O. Nys. “Dynamic element matching techniques with arbitrary noise shaping function”. In: *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*. Vol. 1. IEEE. 1996, pp. 293–296 (cit. on p. 18).

[33] V. O’Brien and B. Mullane. “High order mismatch shaping for low oversampling rates”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 67.1 (2019), pp. 42–46 (cit. on pp. 18, 29).

[34] L. Jie et al. “An overview of noise-shaping SAR ADC: From fundamentals to the frontier”. In: *IEEE Open Journal of the Solid-State Circuits Society* 1 (2021), pp. 149–161 (cit. on p. 18).

[35] X. Han et al. “A Vector Pair Based DWA Algorithm for Linearity Enhancement of CDACs in the NS-SAR ADC”. In: *2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE. 2022, pp. 1–5 (cit. on p. 18).

[36] J.-y. Kojima et al. “DWA algorithm for band-pass ΔΣ DAC with ternary unit cells”. In: *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. IEEE. 2018, pp. 1–3 (cit. on p. 18).

[37] D. Mo and M. F. Duarte. “Binary Sequence Set Design for Interferer Rejection in Multi-Branch Modulation”. In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 3769–3778. DOI: 10.1109/TSP.2020.3000288 (cit. on p. 28).

[38] D. Mo and M. F. Duarte. “Design of spectrally shaped binary sequences via randomized convex relaxations”. In: *2015 49th Asilomar Conference on Signals, Systems and Computers*. IEEE. 2015, pp. 164–168 (cit. on p. 28).

A VCM-based SAR Stage Equations

The sampling step forces the connection of all capacitors to the VCM voltage as to sample the input, the quantization of the input into a digital word, is done using the transitions from sampling to step 1, step 1 to step 2, and so on, applying the charge conservation principle (on nodes $v_{p/n}$). These steps can be described by set of equations (A.1), (A.2), (A.3) and (A.4), where the value $N = 2^{n_{bits}}$, in this case $n_{bits} = 3 \Rightarrow N = 8$. To determine the first bit, b_2, the charge conservation principle renders equations:

$$(v_p - VCM)\sum_{k=1}^{N} C_{k_p} = (vin_p - VCM)\sum_{k=1}^{N} C_{k_p} \Rightarrow v_p = vin_p \tag{A.1a}$$

$$(v_n - VCM)\sum_{k=1}^{N} C_{k_n} = (vin_n - VCM)\sum_{k=1}^{N} C_{k_n} \Rightarrow v_n = vin_n \tag{A.1b}$$

$$v_p - v_n > V_{offset} \Rightarrow b_2 \tag{A.1c}$$

Following the determination of b_2, the C-DAC drivers change the connections of the MSB capacitors, if $b_2 = 1$, then on the positive half, they are connected to gnd and on the negative half to V_{Ref}, if $b_2 = 0$, then the reverse happens. Utilizing the charge conservation principle based on the new connections the second bit can be determined:

$$(v_p - VCM)\sum_{k=1}^{N} C_{k_p} = (v_p - VCM)\sum_{k=5}^{8} C_{k_p} + (v_p - V_{Ref}\overline{b_2} - gndb_0)\sum_{k=1}^{4} C_{k_p} \tag{A.2a}$$

$$(v_n - VCM)\sum_{k=1}^{N} C_{k_n} = (v_n - VCM)\sum_{k=5}^{8} C_{k_n} + (v_n - V_{Ref}b_2 - gnd\overline{b_2})\sum_{k=1}^{4} C_{k_n} \tag{A.2b}$$

$$\begin{aligned} v_p - v_n = (vin_p - vin_n) - VCM\left[\frac{\sum_{k=1}^{4} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{\sum_{k=1}^{4} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] \\ + V_{Ref}\left[\overline{b_2}\frac{\sum_{k=1}^{4} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - b_2\frac{\sum_{k=1}^{4} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] \\ + gnd\left[b_2\frac{\sum_{k=1}^{4} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \overline{b_2}\frac{\sum_{k=1}^{4} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] > V_{offset} \Rightarrow b_1 \end{aligned} \tag{A.2c}$$

The third bit can be determined using the value of the second bit b_1, the connections are changed in the same way but now only for the second MSB capacitors, while the other remain the same:

$$\begin{aligned} (v_p - VCM)\sum_{k=1}^{N} C_{k_p} = (v_p - VCM)\sum_{k=7}^{8} C_{k_p} + (v_p - V_{Ref}\overline{b_2} - gndb_2)\sum_{k=1}^{4} C_{k_p} \\ + (v_p - V_{Ref}\overline{b_1} - gndb_1)\sum_{k=5}^{6} C_{k_p} \end{aligned} \tag{A.3a}$$

$$
\begin{aligned}
(v_n - VCM)\sum_{k=1}^{N} C_{k_n} = (v_n - VCM)\sum_{k=7}^{8} C_{k_n} + (v_n - V_{Ref} b_2 - gnd\overline{b_2})\sum_{k=1}^{4} C_{k_n} \\
+ (v_n - V_{Ref} b_1 - gnd\overline{b_1})\sum_{k=5}^{6} C_{k_n}
\end{aligned}
\tag{A.3b}
$$

$$
\begin{aligned}
v_p - v_n =& (vin_p - vin_n) - VCM\left[\frac{\sum_{k=1}^{6} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{\sum_{k=1}^{6} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] \\
&+ V_{Ref}\left[\frac{\overline{b_2}\sum_{k=1}^{4} C_{k_p} + \overline{b_1}\sum_{k=5}^{6} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{b_2\sum_{k=1}^{4} C_{k_n} + b_1\sum_{k=5}^{6} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] \\
&+ gnd\left[\frac{b_0\sum_{k=1}^{4} C_{k_p} + b_1\sum_{k=5}^{6} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{\overline{b_0}\sum_{k=1}^{4} C_{k_n} + \overline{b_1}\sum_{k=5}^{6} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right], \\
& v_p - v_n > V_{offset} \Rightarrow b_0
\end{aligned}
\tag{A.3c}
$$

To determine the residue voltage, the LSB capacitors are changed, in the same way as explained, and using the charge conservation principle the residue voltage can be calculated and then delivered to the next stage:

$$
\begin{aligned}
(v_p - VCM)\sum_{k=1}^{N} C_{k_p} = (v_p - VCM)\sum_{k=8}^{8} C_{k_p} + (v_p - V_{Ref}\overline{b_2} - gndb_2)\sum_{k=1}^{4} C_{k_p} \\
+ (v_p - V_{Ref}\overline{b_1} - gndb_1)\sum_{k=5}^{6} C_{k_p} + (v_p - V_{Ref}\overline{b_0} - gndb_0)\sum_{k=7}^{7} C_{k_p}
\end{aligned}
\tag{A.4a}
$$

$$
\begin{aligned}
(v_n - VCM)\sum_{k=1}^{N} C_{k_n} = (v_n - VCM)\sum_{k=8}^{8} C_{k_n} + (v_n - V_{Ref} b_2 - gnd\overline{b_2})\sum_{k=1}^{4} C_{k_n} \\
+ (v_n - V_{Ref} b_1 - gnd\overline{b_1})\sum_{k=5}^{6} C_{k_n} + (v_n - V_{Ref} b_0 - gnd\overline{b_0})\sum_{k=7}^{7} C_{k_n}
\end{aligned}
\tag{A.4b}
$$

$$
\begin{aligned}
V_{Residue} = v_p - v_n = (vin_p - vin_n) &- VCM\left[\frac{\sum_{k=1}^{7} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} - \frac{\sum_{k=1}^{7} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}\right] \\
&+ V_{Ref}\frac{\overline{b_2}\sum_{k=1}^{4} C_{k_p} + \overline{b_1}\sum_{k=5}^{6} C_{k_p} + \overline{b_0}\sum_{k=7}^{7} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} \\
&- V_{Ref}\frac{b_2\sum_{k=1}^{4} C_{k_n} + b_1\sum_{k=5}^{6} C_{k_n} + b_0\sum_{k=7}^{7} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}} \\
&+ gnd\frac{b_2\sum_{k=1}^{4} C_{k_p} + b_1\sum_{k=5}^{6} C_{k_p} + b_0\sum_{k=7}^{7} C_{k_p}}{\sum_{k=1}^{N} C_{k_p}} \\
&- gnd\frac{\overline{b_2}\sum_{k=1}^{4} C_{k_n} + \overline{b_1}\sum_{k=5}^{6} C_{k_n} + \overline{b_0}\sum_{k=7}^{7} C_{k_n}}{\sum_{k=1}^{N} C_{k_n}}, \\
&\underbrace{v_p - v_n > V_{offset} \Rightarrow b_{-1}}_{\text{Possible in last sub-SAR or solo SAR ADC}}
\end{aligned}
\tag{A.4c}
$$

B MSB Permutation Equations

Equations B.1 and B.1 illustrate the error cancellation of the MSB related ratio for a SAR stage of resolution 3 bits.

$$\begin{aligned}
&\text{Sample i: } V_{Ref}\left[\frac{\overline{b_{2_i}}\,(C_1+C_2+C_3+C_4)_p+\overline{b_{1_i}}\,(C_5+C_6)_p+\overline{b_{0_i}}\,(C_7)_p}{\sum_{k=1}^{8}C_{k_p}}\right]\\
&\text{Sample i+1: } V_{Ref}\left[\frac{\overline{b_{2_{i+1}}}\,(C_5+C_6+C_7+C_8)_p+\overline{b_{1_{i+1}}}\,(C_1+C_2)_p+\overline{b_{0_{i+1}}}\,(C_3)_p}{\sum_{k=1}^{8}C_{k_p}}\right]\\
&\text{Average: } \frac{V_{Ref}}{2}\left[\begin{array}{l}\dfrac{\overline{b_{2_i}}\,(C_1+C_2+C_3+C_4)_p+\overline{b_{2_{i+1}}}\,(C_5+C_6+C_7+C_8)_p}{\sum_{k=1}^{8}C_{k_p}}+\\ \dfrac{\overline{b_{1_i}}\,(C_5+C_6)_p+\overline{b_{1_{i+1}}}\,(C_1+C_2)_p+\overline{b_{0_i}}\,(C_7)_p+\overline{b_{0_{i+1}}}\,(C_3)_p}{\sum_{k=1}^{8}C_{k_p}}\end{array}\right]
\end{aligned} \tag{B.1}$$

If a sample is equal to the previous one, then the error cancellation is observed as equation (B.2) suggests.

$$\begin{aligned}
&\text{Average with Sample i = Sample i+1 :}\\
&\frac{V_{Ref}}{2}\left[\begin{array}{l}\dfrac{\overline{b_{2_i}}\,(C_1+C_2+C_3+C_4+C_5+C_6+C_7+C_8)_p}{\sum_{k=1}^{8}C_{k_p}}\\ +\dfrac{\overline{b_{1_i}}\,(C_5+C_6+C_1+C_2)_p+\overline{b_{0_i}}\,(C_7+C_3)_p}{\sum_{k=1}^{8}C_{k_p}}\end{array}\right]\Rightarrow\\
&\frac{V_{Ref}}{2}\left[\overline{b_{2_i}}\,\frac{\sum_{k=1}^{8}C_{k_p}}{\sum_{k=1}^{8}C_{k_p}}\overset{1}{\nearrow}+\frac{\overline{b_{1_i}}\,(C_5+C_6+C_1+C_2)_p+\overline{b_{0_i}}\,(C_7+C_3)_p}{\sum_{k=1}^{8}C_{k_p}}\right]
\end{aligned} \tag{B.2}$$

C Full Resolution Level MSB Permutation

The capacitor swapping that can perform a 3 level MSB permutation, full resolution level MSB permutation for a SAR stage of 3 bits, is given by table C.1

Table C.1: Permutation cycle of unit capacitors according to the full resolution-level MSB permutation and their respective positions in a C-DAC of a 3 bit stage.

		Capacitors							
		C_{MSB}				C_{MSB-1}		C_{LSB}	C_{bin}
Sample Number	Vin_1	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
	Vin_2	C_5	C_6	C_7	C_8	C_1	C_2	C_3	C_4
	Vin_3	C_3	C_4	C_1	C_2	C_7	C_8	C_5	C_6
	Vin_4	C_7	C_8	C_5	C_6	C_3	C_4	C_1	C_2
	Vin_5	C_2	C_1	C_4	C_3	C_6	C_5	C_8	C_7
	Vin_6	C_6	C_5	C_8	C_7	C_2	C_1	C_4	C_3
	Vin_7	C_4	C_3	C_2	C_1	C_8	C_7	C_6	C_5
	Vin_8	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1

The block diagram that could conceptually perform the full resolution level MSB permutation is show in figure C.1.

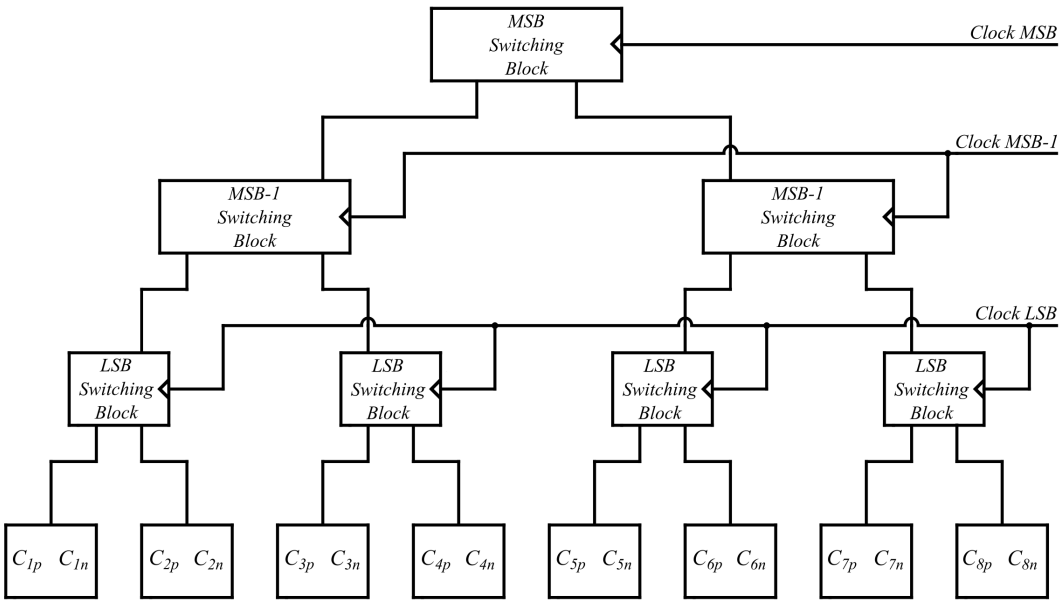


Figure C.1: Block diagram of the possible digital implementation of the algorithm full resolution-level MSB permutation for a 3 bit stage

As seen in figure C.1 the block diagram has as inputs n_{bits} clocks that whose properties can be described by table C.2.

Table C.2: Clock properties for the implementation of the full resolution-level MSB permutation according to the block diagram shown.

	Sample Number							
	Vin_1	Vin_2	Vin_3	Vin_4	Vin_5	Vin_6	Vin_7	Vin_8
$Clock_{MSB}$	—-	Rising	Rising	Rising	Rising	Rising	Rising	Rising
$Clock_{MSB-1}$	—-	—-	Rising	—-	Rising	—-	Rising	—-
$Clock_{LSB}$	—-	—-	—-	—-	Rising	—-	—-	—-

D Algorithms Mathematical Description

D.1 Mathematical Analysis of the Low-Pass Operation DWA

To realize a mathematical description of the algorithm it is necessary to first describe the capacitors. All capacitors, in a given C-DAC can be represented by an ideal component and a component that refers to the mismatch. The ideal component can be equal to the mean of the capacitor production μ_C, however, since the ratios are obtained with respect to the capacitors present in the C-DAC, a more useful description of the ideal component of the capacitors is the mean of the capacitors present in each of the halves. For practical reasons the ideal component will be assessed as the mean of capacitors $(C_1 \dots C_7)_{p/n}$ since, $(C_8)_{p/n}$ is not present in the algorithm. The mismatch component related to each of the capacitors can be described as the difference between the real capacitance value and the mean, such that the real capacitance value of capacitor $C_{k_{p/n}}$ has the relation in equations (D.1):

$$(w_k)_{p/n} = (w_{mean})_{p/n} + (e_k)_{p/n} \tag{D.1a}$$

$$(w_{mean})_{p/n} = \frac{1}{N-1}\sum_{k=1}^{N-1}(w_k)_{p/n}, \quad N = 2^{n_{bits}} \tag{D.1b}$$

From the previous equations it can be derived that the sum of all the mismatch components is zero, seen in equation (D.2):

$$(w_{mean})_{p/n} = \frac{1}{N-1}\sum_{k=1}^{N-1}\left[(w_{mean})_{p/n} + (e_k)_{p/n}\right] \Rightarrow \sum_{k=1}^{N-1}(e_k)_{p/n} = 0 \tag{D.2}$$

With the algorithm description in section 4.1 it is possible to obtain the expressions for the value of the decision signals. Since the current behaviour of the algorithm depends on the previous state of the algorithm, a recurrent formula is useful in representing the value of both d_p and d_n. The recurrent expressions are presented in equation (D.3).

$$\begin{cases} d_p(i,j) = d_p(i,j-1) + \overline{bit(i,j)}2^{n_{bits}-j} - (N-1)Overflow_p(i,j) \\ d_n(i,j) = d_n(i,j-1) + bit(i,j)2^{n_{bits}-j} - (N-1)Overflow_n(i,j) \end{cases} \tag{D.3}$$

Because the positive and negative expressions are very similar and differ only in some important details, a simplification for equation (D.3) can be written as equation (D.4):

$$d_{p/n}(i,j) = d_{p/n}(i,j-1) + \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j} - (N-1)Overflow_{p/n}(i,j) \tag{D.4}$$

The overflow term is a binary value that is only 1 if the decision signal value exceeds the maximum value, hence it can be described in the same notation used in equation (D.4) by equation (D.5).

$$Overflow_{p/n}(i,j) = \left\lceil \frac{d_{p/n}(i,j-1) + \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j}}{N-1} \right\rceil - 1 \tag{D.5}$$

Using the recurring formula it is possible to also describe the selected capacitors, SC, connected to either V_{Ref}, equation (D.6c), or gnd, equation (D.7), and to also describe the capacitors that are present in the VCM term from set of equations (A.1) to (A.4) ((D.8)).

1. Selected Capacitors for V_{Ref}: When $Overflow_{p/n}(i,j) = 0$ then:

$$SC_{V_{Ref}\,p/n}(i,j) = \left\{\begin{matrix}\overline{bit(i,j)}\\ bit(i,j)\end{matrix}\right\} \sum_{k=d_{p/n}(i,j-1)+1}^{d_{p/n}(i,j)} (w_k)_{p/n} \Rightarrow \sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} \tag{D.6a}$$

   For the case when $Overflow_{p/n}(i,j) = \left\{\begin{matrix}0\\1\end{matrix}\right\} \vee \left\{\begin{matrix}1\\0\end{matrix}\right\}$, then:

$$\begin{aligned} SC_{V_{Ref}\,p/n}(i,j) = \left\{\begin{matrix}\overline{bit(i,j)}\\ bit(i,j)\end{matrix}\right\} \left[\sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} + \sum_{k=d_{p/n}(i,j-1)+1}^{N-1} (w_k)_{p/n}\right] \Rightarrow \\ \left\{\begin{matrix}\overline{bit(i,j)}\\ bit(i,j)\end{matrix}\right\} \left[\sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} + \sum_{k=1}^{N-1} (w_k)_{p/n}\right] \end{aligned} \tag{D.6b}$$

   General expression $Overflow$ dependent:

$$SC_{V_{Ref}\,p/n}(i,j) = \sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} + Overflow_{p/n}(i,j) \sum_{k=1}^{N-1} (w_k)_{p/n} \tag{D.6c}$$

2. Selected Capacitors for V_{Ref}, general formula $Overflow$ dependent:

$$SC_{gnd\,p/n}(i,j) = \sum_{k=1}^{d_{n/p}(i,j)} (w_{N-k})_{p/n} - \sum_{k=1}^{d_{n/p}(i,j-1)} (w_{N-k})_{p/n} + Overflow_{p/n}(i,j) \sum_{k=1}^{N-1} (w_k)_{p/n} \tag{D.7}$$

3. Capacitors present in the VCM term from set of equations (A.1) to (A.4):

$$SC_{VCM\,p/n}(i,j) = SC_{V_{Ref}\,p/n}(i,j) + SC_{gnd\,p/n}(i,j) \tag{D.8}$$

Since each of the equations only describe which capacitors are selected in a given step, to build equation (2.1) using the algorithm, it is needed to account for the whole accumulated set of capacitors. The accumulated selected capacitors, $ASC_{V_{Ref}\,p/n}(i,j)$, $ASC_{gnd\,p/n}(i,j)$ and $ASC_{VCM\,p/n}(i,j)$ are defined in set of equations (D.9).

$$ASC_{V_{Ref}\,p/n}(i,j) = \sum_{l=1}^{j} SC_{V_{Ref}\,p/n}(i,l) \tag{D.9a}$$

$$ASC_{gnd\,p/n}(i,j) = \sum_{l=1}^{j} SC_{gnd\,p/n}(i,l) \tag{D.9b}$$

$$ASC_{VCM\,p/n}(i,j) = ASC_{V_{Ref}\,p/n}(i,j) + ASC_{gnd\,p/n}(i,j) \tag{D.9c}$$

Set of equations (D.9) allow for the construction of the C-DAC voltage utilizing the algorithm's selection, an analogous of the subtrahend term of equation (2.1). Equation (D.10)

presents the C-DAC voltage.

$$\begin{aligned} y_{DAC}(i,j) = VCM & \left[\frac{ASC_{VCM\,p}(i,j)}{(N-1)(w_{mean})_p + (w_{bin})_p} - \frac{ASC_{VCM\,n}(i,j)}{(N-1)(w_{mean})_n + (w_{bin})_n} \right] \\ -V_{Ref} & \left[\frac{ASC_{V_{Ref}\,p}(i,j)}{(N-1)(w_{mean})_p + (w_{bin})_p} - \frac{ASC_{V_{Ref}\,n}(i,j)}{(N-1)(w_{mean})_n + (w_{bin})_n} \right] \\ -gnd & \left[\frac{ASC_{gnd\,p}(i,j)}{(N-1)(w_{mean})_p + (w_{bin})_p} - \frac{ASC_{gnd\,n}(i,j)}{(N-1)(w_{mean})_n + (w_{bin})_n} \right] \end{aligned} \tag{D.10}$$

Expanding equation (D.9a) with equation (D.6c) renders equation (D.11) for the V_{Ref} connected capacitors.

$$ASC_{V_{Ref}\,p/n}(i,j) = \sum_{l=1}^{j} \left[\sum_{k=1}^{d_{p/n}(i,l)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,l-1)} (w_k)_{p/n} + Overflow_{p/n}(i,l) \sum_{k=1}^{N-1} (w_k)_{p/n} \right] \tag{D.11}$$

Replacing $(w_k)_{p/n}$ by expression (D.1a), that separates the ideal contribution and the mismatch error in expression (D.11) gives (D.12).

$$\begin{aligned} ASC_{V_{Ref}\,p/n}(i,j) = & \sum_{l=1}^{j} \left[\sum_{k=1}^{d_{p/n}(i,l)} \left((w_{mean})_{p/n} + (e_k)_{p/n} \right) - \sum_{k=1}^{d_{p/n}(i,l-1)} \left((w_{mean})_{p/n} + (e_k)_{p/n} \right) \right] \\ & + \sum_{l=1}^{j} \left[Overflow_{p/n}(i,l) \sum_{k=1}^{N-1} \left((w_{mean})_{p/n} + (e_k)_{p/n} \right) \right] \end{aligned} \tag{D.12}$$

Using the result of equation (D.2), equation (D.12) reduces to equation (D.13).

$$\begin{aligned} ASC_{V_{Ref}\,p/n}(i,j) = & \underbrace{\sum_{l=1}^{j} \left[\sum_{k=1}^{d_{p/n}(i,l)} (e_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,l-1)} (e_k)_{p/n} \right]}_{\text{Mismatch Term}} \\ & \underbrace{+ (w_{mean})_{p/n} \sum_{l=1}^{j} \left[d_{p/n}(i,l) - d_{p/n}(i,l-1) + Overflow_{p/n}(i,l)(N-1) \right]}_{\text{Ideal Term}} \end{aligned} \tag{D.13}$$

From equation (D.13) it is possible to see that $ASC_{V_{Ref}\,p/n}(i,j)$ can be separated into a mismatch term and an ideal one. Remembering the definition of the decision signals d_p and d_n in equation (D.3), it is possible to simplify equation (D.13). Furthermore, if an expansion of the mismatch term is done, further simplifications become apparent. Equation (D.14) demonstrates the simplifications.

$$
\begin{aligned}
ASC_{V_{Ref}\,p/n}(i,j) = {} & \cancel{\sum_{k=1}^{d_{p/n}(i,1)} (e_k)_{p/n}} - \sum_{k=1}^{d_{p/n}(i,0)} (e_k)_{p/n} + \cancel{\sum_{k=1}^{d_{p/n}(i,2)} (e_k)_{p/n}} - \cancel{\sum_{k=1}^{d_{p/n}(i,1)} (e_k)_{p/n}} + \dots \\
& \dots + \cancel{\sum_{k=1}^{d_{p/n}(i,j-1)} (e_k)_{p/n}} - \cancel{\sum_{k=1}^{d_{p/n}(i,j-2)} (e_k)_{p/n}} + \sum_{k=1}^{d_{p/n}(i,j)} (e_k)_{p/n} - \cancel{\sum_{k=1}^{d_{p/n}(i,j-1)} (e_k)_{p/n}} \\
& \underbrace{+ (w_{mean})_{p/n} \sum_{l=1}^{j} \left[\begin{Bmatrix} \overline{bit(i,l)} \\ bit(i,l) \end{Bmatrix} 2^{n_{bits}-l} \right]}_{\text{Ideal Term}} \Rightarrow \\
= {} & \underbrace{\sum_{k=1}^{d_{p/n}(i,j)} (e_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,0)} (e_k)_{p/n}}_{\text{Mismatch Term}} + \underbrace{(w_{mean})_{p/n} \sum_{l=1}^{j} \left[\begin{Bmatrix} \overline{bit(i,l)} \\ bit(i,l) \end{Bmatrix} 2^{n_{bits}-l} \right]}_{\text{Ideal Term}}
\end{aligned}
\tag{D.14}
$$

To simplify further notation, it is possible to define $\sum_{k=1}^{d_{p/n}(i,j)} (e_k)_{p/n}$ as the accumulated mismatch error regarding the capacitors connected to V_{Ref}, $AME_{V_{Ref}\,p/n}(i,j) = \sum_{k=1}^{d_{p/n}(i,j)} (e_k)_{p/n}$. Also notable is the fact that by definition on equation (D.3), $d_{p/n}(i,0) = d_{p/n}(i-1, n_{bits})$, hence equation (D.14) simplifies to equation (D.15).

$$
\begin{aligned}
ASC_{V_{Ref}\,p/n}(i,j) = {} & \underbrace{AME_{V_{Ref}\,p/n}(i,j) - AME_{V_{Ref}\,p/n}(i-1, n_{bits})}_{\text{Mismatch Term}} \\
& \underbrace{+ (w_{mean})_{p/n} \sum_{l=1}^{j} \left[\begin{Bmatrix} \overline{bit(i,l)} \\ bit(i,l) \end{Bmatrix} 2^{n_{bits}-l} \right]}_{\text{Ideal Term}}
\end{aligned}
\tag{D.15}
$$

An analogous derivation for the expression of $ASC_{gnd\,p/n}(i,j)$ can be done resulting in expression (D.16) if $\sum_{k=1}^{d_{p/n}(i,j)} (e_{N-k})_{p/n}$ is defined as the accumulated mismatch error regarding the capacitors connected to gnd, $AME_{gnd\,p/n}(i,j) = \sum_{k=1}^{d_{p/n}(i,j)} (e_{N-k})_{p/n}$.

$$
\begin{aligned}
ASC_{gnd\,p/n}(i,j) = {} & \underbrace{AME_{gnd\,p/n}(i,j) - AME_{gnd\,p/n}(i-1, n_{bits})}_{\text{Mismatch Term}} \\
& \underbrace{+ (w_{mean})_{p/n} \sum_{l=1}^{j} \left[\begin{Bmatrix} bit(i,l) \\ \overline{bit(i,l)} \end{Bmatrix} 2^{n_{bits}-l} \right]}_{\text{Ideal Term}}
\end{aligned}
\tag{D.16}
$$

Utilizing the definition for $ASC_{VCM\,p/n}(i,j)$ in equation (D.9c), an expanded expression for it is achieved in equation (D.17).

$$
\begin{aligned}
ASC_{VCMp/n}(i,j) &= \sum_{l=1}^{j}\left[\begin{cases}\overline{bit(i,l)}+\cancel{bit(i,l)}\\ \cancel{bit(i,l)}+\overline{bit(i,l)}\end{cases}^{1} 2^{n_{bits}-l}\right] + AME_{V_{Ref}\,p/n}(i,j)\\
&- AME_{V_{Ref}\,p/n}(i-1,n_{bits}) + AME_{gnd\,p/n}(i,j) - AME_{gnd\,p/n}(i-1,n_{bits}) \Rightarrow\\
\Rightarrow ASC_{VCMp/n}(i,j) &= \underbrace{\sum_{l=1}^{j} 2^{n_{bits}-l}}_{\text{Ideal Term}}\\
&+ \underbrace{AME_{V_{Ref}\,p/n}(i,j) - AME_{V_{Ref}\,p/n}(i-1,n_{bits}) + AME_{gnd\,p/n}(i,j) - AME_{gnd\,p/n}(i-1,n_{bits})}_{\text{Mismatch Term}}
\end{aligned}
\tag{D.17}
$$

The combinations of equations (D.15) to (D.9c), with equation (D.10) allows for a general step by step representation of the C-DAC voltages with mismatch errors and the effect that the algorithm has on the mismatch. However, as mentioned earlier (equation (2.1)) in section 2.1 the C-DAC step by step behaviour can be condensed into the last step, $j = n_{bits}$, further simplifying the analysis of the algorithm given that only the last voltage of the conversion needs to be analyzed. The focus of the discussion will be then turned from $y_{DAC}(i,j)$ into $y_{DAC}(i,n_{bits}) = y_{DAC}(i)$, coupled with the fact that similar to equations (D.15) to (D.9c) there is a distinction between the ideal term and the mismatch term, $y_{DAC}(i,j) = y_{DAC\,Ideal}(i,j) + y_{DAC\,Mismatch}(i,j) \Rightarrow y_{DAC}(i) = y_{DAC\,Ideal}(i) + y_{DAC\,Mismatch}(i)$, set of equations (D.18) emerges.

$$
y_{DAC}(i) = y_{DAC\,Ideal}(i) + y_{DAC\,Mismatch}(i) \tag{D.18a}
$$

$$
\begin{aligned}
y_{DAC\,Ideal}(i) &= VCM\left[\frac{(w_{mean})_p(2^{n_{bits}}-1)}{(N-1)(w_{mean})_p+(w_{bin})_p} - \frac{(w_{mean})_n(2^{n_{bits}}-1)}{(N-1)(w_{mean})_n+(w_{bin})_n}\right]\\
&-V_{Ref}\left[\frac{(w_{mean})_p\sum_{l=1}^{n_{bits}}\left[\overline{bit(i,l)}2^{n_{bits}-l}\right]}{(N-1)(w_{mean})_p+(w_{bin})_p} - \frac{(w_{mean})_n\sum_{l=1}^{n_{bits}}\left[bit(i,l)2^{n_{bits}-l}\right]}{(N-1)(w_{mean})_n+(w_{bin})_n}\right]\\
&-gnd\left[\frac{(w_{mean})_p\sum_{l=1}^{n_{bits}}\left[bit(i,l)2^{n_{bits}-l}\right]}{(N-1)(w_{mean})_p+(w_{bin})_p} - \frac{(w_{mean})_n\sum_{l=1}^{n_{bits}}\left[\overline{bit(i,l)}2^{n_{bits}-l}\right]}{(N-1)(w_{mean})_n+(w_{bin})_n}\right]
\end{aligned}
\tag{D.18b}
$$

$$
\begin{aligned}
&y_{DACMismatch}(i) =\\
&VCM\left[\begin{array}{l}\dfrac{AME_{V_{Ref}\,p}(i) - AME_{V_{Ref}\,p}(i-1) + AME_{gnd\,p}(i) - AME_{gnd\,p}(i-1)}{(N-1)(w_{mean})_p+(w_{bin})_p}\\ \quad - \dfrac{AME_{V_{Ref}\,n}(i) - AME_{V_{Ref}\,n}(i-1) + AME_{gnd\,n}(i) - AME_{gnd\,n}(i-1)}{(N-1)(w_{mean})_n+(w_{bin})_n}\end{array}\right]\\
&-V_{Ref}\left[\frac{AME_{V_{Ref}\,p}(i) - AME_{V_{Ref}\,p}(i-1)}{(N-1)(w_{mean})_p+(w_{bin})_p} - \frac{AME_{V_{Ref}\,n}(i) - AME_{V_{Ref}\,n}(i-1)}{(N-1)(w_{mean})_n+(w_{bin})_n}\right]\\
&-gnd\left[\frac{AME_{gnd\,p}(i) - AME_{gnd\,p}(i-1)}{(N-1)(w_{mean})_p+(w_{bin})_p} - \frac{AME_{gnd\,n}(i) - AME_{gnd\,n}(i-1)}{(N-1)(w_{mean})_n+(w_{bin})_n}\right]
\end{aligned}
\tag{D.18c}
$$

Equation (D.18b) although named ideal, it is not completely so, this due to the binary capacitor of weight $(w_{bin})_{p/n}$ not necessarily equating to $(w_{mean})_{p/n}$ nor is it cycled through in the algorithm. Doing so, and maintaining the description given in section 4.1, would result in conflicts, where in some cases the algorithm would select the a group of capacitors to connect to V_{Ref}, and on the next step of the same conversion, the algorithm would choose one or more of the capacitors used in the previous step to connect to gnd. Correcting these conflicts, although not the aim of the presented work, presented greater digital complexity and often with the loss of the Mismatch effect later to be discussed. The binary capacitor not being cycled through is not without consequences but its influence can be diminished if the resolution of the stage is high enough, given that $(N-1)(w_{mean})_{p/n}+(w_{bin})_{p/n}$ can be approximated to $N(w_{mean})_{p/n}$. This way the set of equations (D.18) turn into set of equations (D.19).

$$y_{DAC}(i) = y_{DAC\,Ideal}(i) + y_{DAC\,Mismatch}(i) \tag{D.19a}$$

$$y_{DAC\,Ideal}(i) = -\left[V_{Ref} - gnd\right]\left[\sum_{l=1}^{n_{bits}}\left[\left(\overline{bit(i,l)} - bit(i,l)\right)2^{-l}\right]\right] \tag{D.19b}$$

$$\begin{aligned} &y_{DAC\,Mismatch}(i) = \\ &\frac{VCM}{N}\left[\frac{AME_{V_{Ref}\,p}(i) - AME_{V_{Ref}\,p}(i-1) + AME_{gnd\,p}(i) - AME_{gnd\,p}(i-1)}{(w_{mean})_p} - \frac{AME_{V_{Ref}\,n}(i) - AME_{V_{Ref}\,n}(i-1) + AME_{gnd\,n}(i) - AME_{gnd\,n}(i-1)}{(w_{mean})_n}\right] \\ &-\frac{V_{Ref}}{N}\left[\frac{AME_{V_{Ref}\,p}(i) - AME_{V_{Ref}\,p}(i-1)}{(w_{mean})_p} - \frac{AME_{V_{Ref}\,n}(i) - AME_{V_{Ref}\,n}(i-1)}{(w_{mean})_n}\right] \\ &-\frac{gnd}{N}\left[\frac{AME_{gnd\,p}(i) - AME_{gnd\,p}(i-1)}{(w_{mean})_p} - \frac{AME_{gnd\,n}(i) - AME_{gnd\,n}(i-1)}{(w_{mean})_n}\right] \end{aligned} \tag{D.19c}$$

Equation (D.19b) is the same as (2.2), allowing for the complete separation of the mismatch effects. The mismatch term of $y_{DAC}(i)$ can now be analyzed in terms of frequency response by performing the z-transform of $y_{DAC\,Mismatch}(i)$.

$$\begin{aligned} &\mathcal{Z}\{y_{DAC\,Mismatch}(i)\} = Y_{DAC\,Mismatch}(z) = \\ &\underbrace{\frac{1-Z^{-1}}{N}}_{\text{Explicit Filtering}}\left[\begin{array}{l} VCM\left[\dfrac{AME_{V_{Ref}\,p}(z) + AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z) + AME_{gnd\,n}(z) -}{(w_{mean})_n}\right] \\ \qquad - V_{Ref}\left[\dfrac{AME_{V_{Ref}\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{V_{Ref}\,n}(z)}{(w_{mean})_n}\right] \\ \qquad - gnd\left[\dfrac{AME_{gnd\,p}(z)}{(w_{mean})_p} - \dfrac{AME_{gnd\,n}(z)}{(w_{mean})_n}\right] \end{array}\right] \Rightarrow \\ &Y_{DAC\,Mismatch}(z) = \underbrace{\left[1-Z^{-1}\right]}_{H(z)}\frac{TME(z)}{N} \end{aligned} \tag{D.20}$$

D.2 Mathematical Analysis of the Frequency Contraction Operation of the DWA

The change in the algorithm to add extra zeros in the usable frequency spectrum renders the following equations. Where before the mismatch term could be described by equation (D.19c), it could now be described by equation (D.21).

$$\begin{aligned} &y_{DACMismatch}(i) = \\ &\frac{VCM}{N}\left[\begin{array}{l}\dfrac{AME_{V_{Ref}p}(i)-AME_{V_{Ref}p}(i-a)+AME_{gnd_p}(i)-AME_{gnd_p}(i-a)}{(w_{mean})_p} \\ \quad -\dfrac{AME_{V_{Ref}n}(i)-AME_{V_{Ref}n}(i-a)+AME_{gnd_n}(i)-AME_{gnd_n}(i-a)}{(w_{mean})_n}\end{array}\right] \\ &-\frac{V_{Ref}}{N}\left[\frac{AME_{V_{Ref}p}(i)-AME_{V_{Ref}p}(i-a)}{(w_{mean})_p}-\frac{AME_{V_{Ref}n}(i)-AME_{V_{Ref}n}(i-a)}{(w_{mean})_n}\right] \\ &-\frac{gnd}{N}\left[\frac{AME_{gnd_p}(i)-AME_{gnd_p}(i-a)}{(w_{mean})_p}-\frac{AME_{gnd_n}(i)-AME_{gnd_n}(i-a)}{(w_{mean})_n}\right] \end{aligned} \tag{D.21}$$

The frequency response is then given by equation (D.22).

$$\begin{aligned} &\mathcal{Z}\{y_{DACMismatch}(i)\} = Y_{DACMismatch}(z) = \\ &\underbrace{\frac{1-Z^{-a}}{N}}_{\text{Explicit Filtering}}\left[\begin{array}{l} VCM\left[\dfrac{AME_{V_{Ref}p}(z)+AME_{gnd_p}(z)}{(w_{mean})_p}-\dfrac{AME_{V_{Ref}n}(z)+AME_{gnd_n}(z)-}{(w_{mean})_n}\right] \\ \qquad -V_{Ref}\left[\dfrac{AME_{V_{Ref}p}(z)}{(w_{mean})_p}-\dfrac{AME_{V_{Ref}n}(z)}{(w_{mean})_n}\right] \\ \qquad -gnd\left[\dfrac{AME_{gnd_p}(z)}{(w_{mean})_p}-\dfrac{AME_{gnd_n}(z)}{(w_{mean})_n}\right] \\ \\ \end{array}\right] \Rightarrow \\ &Y_{DACMismatch}(z) = \underbrace{\left[1-Z^{-a}\right]}_{H(z)}\frac{TME(z)}{N} \end{aligned} \tag{D.22}$$

Mathematically the change in the algorithm can be described by a change in the boundary conditions, summarized in system of equations (D.23).

$$\begin{cases} d_{p/n}(i,0) = d_{p/n}(i-a,n_{bits}) = d_{p/n}(i-a) \\ d_{p/n}(A,0) = 2^{n_{bits}}-1, \forall A \in \mathbb{Z} : A \subset [-(a-2);1] \end{cases} \tag{D.23}$$

D.3 Mathematical Analysis of the general Band-Pass, Odd and Even Operation of the DWA

The changes to the algorithm that incorporated the switching signal, $SS(i)$, gives rise to equations (D.24) with $SS(i) = -1$ for the decision signal, and equation (D.25) for the overflow term.

$$d_{p/n}(i,j) = d_{p/n}(i,j-1) - \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j} - (N-1)Overflow_{p/n}(i,j) \tag{D.24}$$

$$Overflow_{p/n}(i,j) = \left\lceil \frac{d_{p/n}(i,j-1) - \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j}}{N-1} \right\rceil - 1 \tag{D.25}$$

In a general fashion, both equations can be altered to equations (D.26) and (D.27) to contain the term $SS(i)$ if it is a binary valued function with positive value 1 and negative value -1.

$$d_{p/n}(i,j) = d_{p/n}(i,j-1) + SS(i) \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j} - (N-1)Overflow_{p/n}(i,j) \tag{D.26}$$

$$Overflow_{p/n}(i,j) = \left\lceil \frac{d_{p/n}(i,j-1) + SS(i) \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} 2^{n_{bits}-j}}{N-1} \right\rceil - 1 \tag{D.27}$$

The capacitor selection for V_{Ref} if $SS(i) = -1$ can be described by equation (D.28) for the case of $Overflow_{p/n}(i,j) = 0$, equation (D.29) for the case when either the negative or the positive $Overflow(i,j) = -1$ (the analogous of $Overflow(i,j) = 1$ if $SS(i) = 1$), and in a general way by equation (D.30).

$$SC_{V_{Ref}\,p/n}(i,j) = \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} \sum_{k=d_{p/n}(i,j)+1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} \Rightarrow \sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} \tag{D.28}$$

$$\begin{aligned} SC_{V_{Ref}\,p/n}(i,j) = \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} \left[\sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} + \sum_{k=d_{p/n}(i,j)+1}^{N-1} (w_k)_{p/n} \right] \Rightarrow \\ \begin{Bmatrix} \overline{bit(i,j)} \\ bit(i,j) \end{Bmatrix} \left[\sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} + \sum_{k=1}^{N-1} (w_k)_{p/n} \right] \end{aligned} \tag{D.29}$$

$$SC_{V_{Ref}\,p/n}(i,j) = SS(i) \left[\sum_{k=1}^{d_{p/n}(i,j-1)} (w_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,j)} (w_k)_{p/n} + Overflow_{p/n}(i,j) \sum_{k=1}^{N-1} (w_k)_{p/n} \right] \tag{D.30}$$

The same can be derived for the selected capacitors to connect to gnd in equation (D.31), if $SS(i) = -1$. The equation that refers to the capacitors present in the VCM term of the C-DAC voltages, remains unchanged, hence it can still be described by equation (D.8) regardless of the

value of $SS(i)$.

$$SC_{gnd\,p/n}(i,j) = SS(i)\left[\sum_{k=1}^{d_{p/n}(i,j-1)}(w_{N-k})_{p/n} - \sum_{k=1}^{d_{p/n}(i,j)}(w_{N-k})_{p/n} + Overflow_{p/n}(i,j)\sum_{k=1}^{N-1}(w_k)_{p/n}\right] \tag{D.31}$$

It should be noted that the value of $SS(i)$ is affecting multiplicatively all the terms, this means that the equalities expressed when defining ASC in equations (D.9a) to (D.9b) still hold when considering $SS(i) = -1$, the only alteration needed is that every term is multiplied by $SS(i)$ and it can be factored out, as seen in equation (D.32).

$$\begin{aligned} ASC_{V_{Ref}\,p/n}(i,j) =& SS(i)\left[\underbrace{\sum_{k=1}^{d_{p/n}(i,j)}(e_k)_{p/n} - \sum_{k=1}^{d_{p/n}(i,0)}(e_k)_{p/n}}_{\text{Mismatch Term}}\right] \\ &+ SS(i)\left[\underbrace{(w_{mean})_{p/n}\sum_{l=1}^{j}\left[\begin{Bmatrix}\overline{bit(i,l)}\\ bit(i,l)\end{Bmatrix}2^{n_{bits}-l}\right]}_{\text{Ideal Term}}\right] \end{aligned} \tag{D.32}$$

When performing the substitution of the ideal term of equation (D.32) by the relation set in equation (D.26), the value of $SS(i)$ is squared on the ideal term of and the demonstration on the ideal end follows the same path as described before. Only the mismatch term is affected by the value of $SS(i)$ that can be always factored out, hence the last relationship in equation (D.33) proves that the change in the algorithm does performs the convolution mentioned above.

$$\begin{aligned} \mathcal{Z}\{y_{DAC\,Mismatch}(i)\} =& \frac{1}{N}\mathcal{Z}\{SS(i)\cdot TME(i)\} \Rightarrow \\ \Rightarrow& \frac{1}{N}\mathcal{Z}\{SS(i)\} \circledast \mathcal{Z}\{TME(i)\} \Rightarrow \\ \Rightarrow& \frac{TME(z)}{N}\left[\mathcal{Z}\{SS(i)\} \circledast \left[1 - Z^{-a}\right]\right] \end{aligned} \tag{D.33}$$

E Digital Implementation of unified DEM algorithm

The digital implementation of the algorithm that unifies the Odd and Even implementations can be guided by the equations derived in appendix D. In a general way, equations (D.26) and (D.27) can be digitally implemented using the circuit seen in figure E.1.

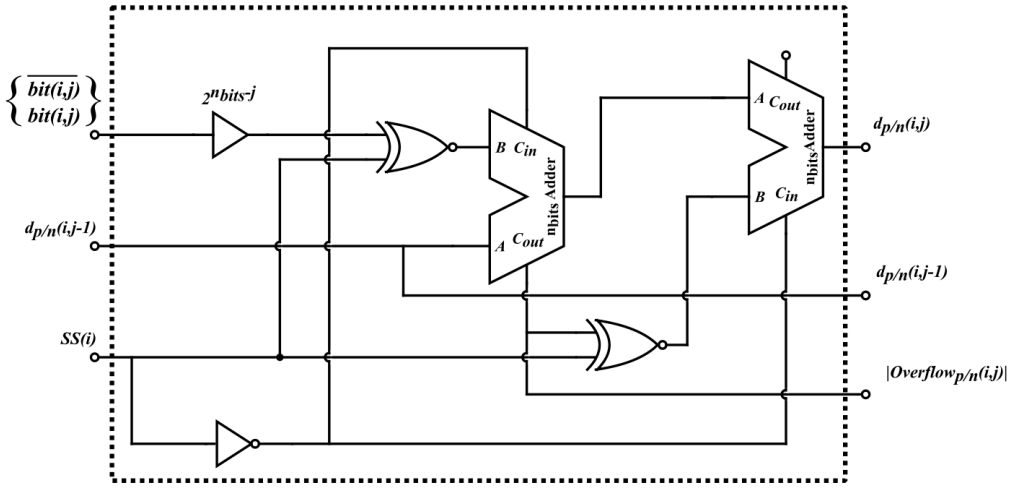


Figure E.1: Digital Implementation of both the decision's signals and $Overflow$ progressive values.

The diagram presented in figure E.1, can perform both the decision signal's progression as well as the overflow progression for the binary values of $SS(i) = 0$ and $SS(i) = 1$. If $SS(i) = 0$, then the full adder takes the 2's complement of the number of capacitors to be selected (in the bit branch) and adds 1 as a carry, this will in turn provide the subtraction with the full adder block. The carry/borrow of this addition/subtraction represents the value of $Overflow$, only in absolute form. The carry, is then added, or subtracted using the 2's complement technique, to the value previously calculated. The result is the value of $d_{p/n}(i, j)$. It is important to remind that these calculations are performed in the binary base, but the capacitor selection later is not. To address each capacitor separately a thermometer decoder can be used as seen in figure E.2. The thermometer decoder can then separate the decision signal's value into a vector of length $2^{n_{bits}} - 1 = N - 1$.

To select the capacitors, the description given in sections 4.1 and 4.3, with figures 4.3 to 4.7 and 4.15 is useful to understand that when performing the element wise XOR function, when $|Overflow| = 0$ and the $XNOR$ function when $|Overflow| = 1$ between $d_{p/n}(i, j)$ and $d_{p/n}(i, j-1)$, the output is an array that only equals one if the index is the same as that of the capacitor that should be selected. This way, to connect C_{k_p} or C_{k_n} to V_{Ref}, circuits shown in figure E.3 can be used.

The circuit takes the value of $Overflow$ to select between the XOR and $XNOR$ functionality. When $Overflow = 0$ then the XOR, whose output is net Y, becomes a buffer to the second input (net X), and when $Overflow = 1$, the gate XOR takes the functionality of a NOT gate. After the calculations are performed the value of net Y either turns ON or OFF the transmission

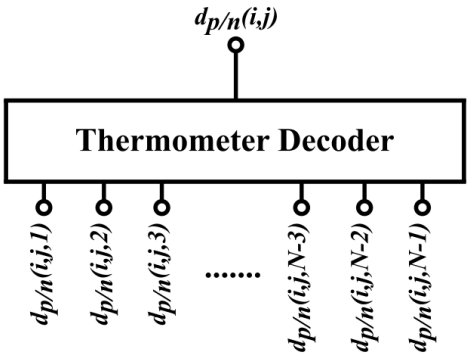


Figure E.2: Thermometer decoder to translate the binary value of the decision signal into an array.

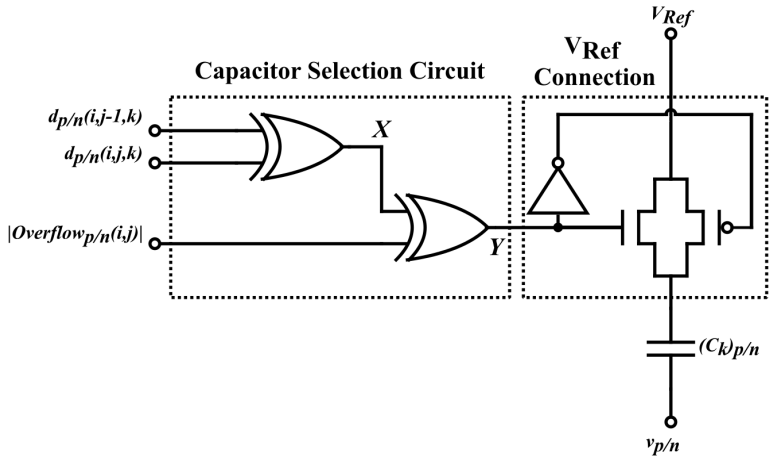


Figure E.3: Digital implementation of the capacitor selection for V_{Ref} in a given step.

gate that connects the capacitor to V_{Ref}. If the decision signal's value does not change in that step ($bit = 0$), then the overflow is necessarily 0, which means that net X is not inverted and is zero, connecting none of the capacitors. To connect capacitors C_{k_p} or C_{k_n} to gnd the same circuits can be used but the index of the decision signals must be reversed, $N-k$, as shown in figure E.4.

The connectivity to the VCM node can be incorporated by acknowledging that if either C_{k_p} or C_{k_n} are connected to either V_{Ref} or gnd the connection to VCM is severed. If neither C_{k_p} or C_{k_n} are connected to either V_{Ref} or gnd, then the connection to VCM is maintained. This functionality can be performed by gate NOR joining nets Y_a and Y_b as seen in figure E.5.

To implement the connection to VCM it is necessary only to connect the nets Y of figure E.3 and figure E.4 to a NOR gate. The digital implementation discussed is only able to accurately select the capacitors in a given step, since the pair of values of ($d_{p/n}(i, j-1)$, $d_{p/n}(i, j)$) is altering every conversion step, so is the value of the XOR and $XNOR$ functions. This means there is no guarantee that the capacitors that are correctly connected in a given step will remain connected to the same two nets until the end of the conversion. Therefore, between the digital implementation of the capacitor selection (net Y/Z) and the transmission gate of the capacitor, there must be a reset circuit that in the beginning of the conversion automatically connects all

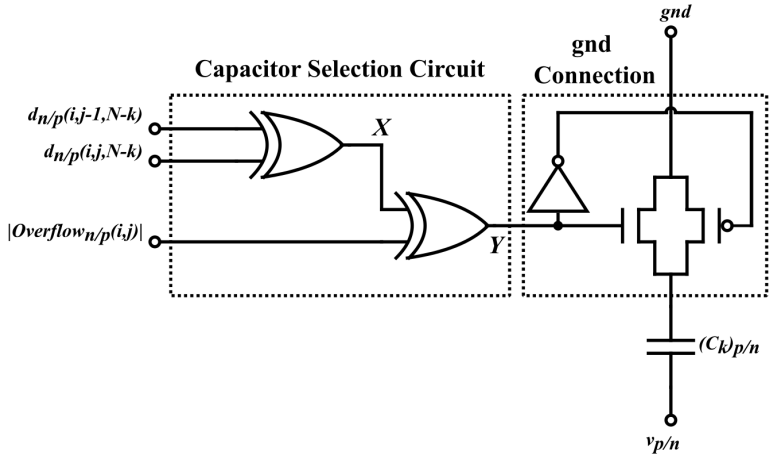


Figure E.4: Digital implementation of the capacitor selection for gnd in a given step.

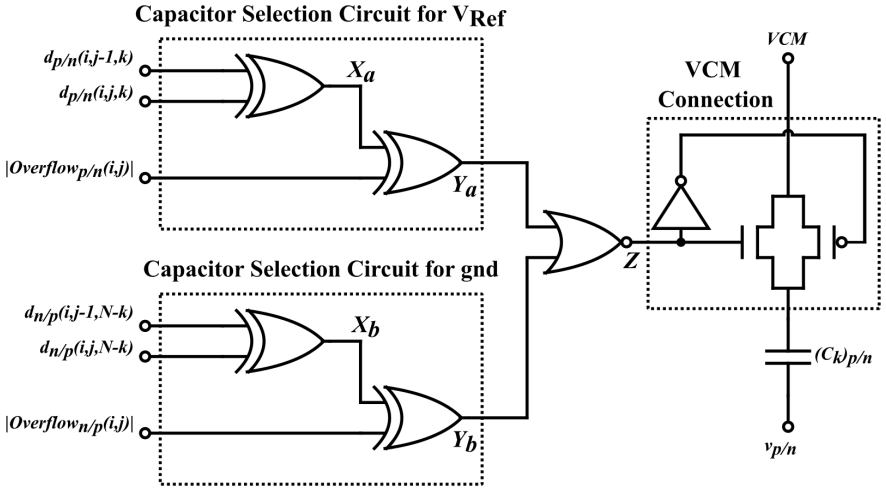
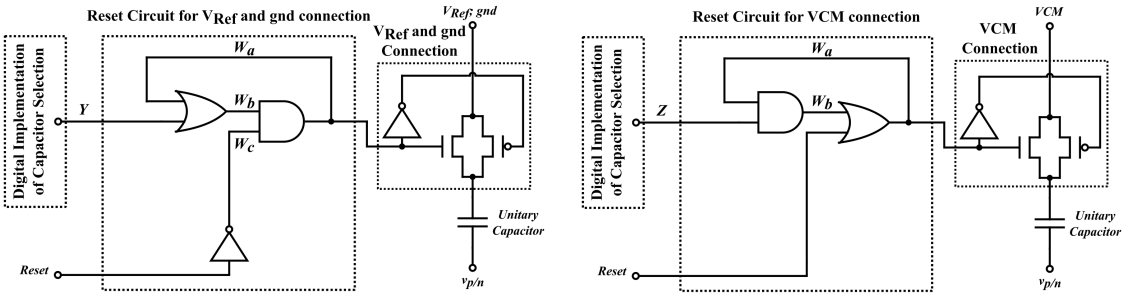


Figure E.5: Digital implementation of the capacitor selection for VCM in a given step.

capacitors to VCM and disconnects all capacitors from gnd or V_{Ref}. Additionally, the circuit must be able to detect the first time the capacitor is either connected to V_{Ref} or gnd or disconnected from VCM, and able to hold that connection/disconnection throughout the rest of the conversion. These functionalities can be performed by the circuits introduced in figure E.6a for the transmission gates connecting the capacitors to either V_{Ref} or gnd, and in figure E.6b for the ones connecting the capacitors to VCM.

The circuit refreshes its state with a reset signal that can be generated using the sampling clock. On figure E.6a when the reset value is 1 net W_c is 0 and net W_a is 0, this makes the connection of the unitary capacitor to V_{Ref} or gnd severed. In figure E.6b if reset is 1 then net W_a is 1 and the circuit forces the connection to VCM. When the reset value is 0 in both circuits the circuit enters in a "listening mode" and net W_b is equal to the input net Y or Z. In figure E.6a the first time Y is 1, the circuit's feedback forces the connection with V_{Ref} or gnd regardless of the next conversion's step Y value. Similarly in figure E.6b, when net Z is 0 for the first time, then the feedback forces the VCM connection to be severed regardless of the next



(a) Connection to either V_{Ref} or gnd

(b) Connection to either VCM

Figure E.6: Digital implementation of the reset circuit capable of maintaining the connection or disconnection of the capacitors once performed throughout the whole conversion.

conversion's step Z value. After all steps have been performed then the $Reset$ value is 1 and the circuit refreshes to its sampling state and the cycle repeats itself.