

UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Generalized Tree-Structured DEM DAC and Enhanced Harmonic Distortion
Correction in Pipelined ADCs

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Electronic Circuits and Systems)

by

Nevena Rakuljic

Committee in charge:

Professor Ian Galton, Chair
Professor James Buckwalter
Professor William S. Hodgkiss
Professor Truong Nguyen
Professor Frank Wuerthwein

2012

Copyright
Nevena Rakuljic, 2012
All rights reserved.

The dissertation of Nevena Rakuljic is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2012

DEDICATION

To my mother and father.

TABLE OF CONTENTS

Signature Page	iii
Dedication	iv
Table of Contents	v
List of Figures	viii
List of Tables.....	x
Acknowledgements.....	xi
Vita.....	xiii
Abstract of the dissertation	xiv
Chapter 1 Tree-Structured DEM DACs with Arbitrary Numbers of Levels	1
I. Introduction.....	1
II. Unity-Weighted DEM DAC Overview.....	3
III. Decomposition of Arbitrary DEM Encoders into Tree Structure.....	8
A. Preliminary Definitions.....	8
B. Decomposition Analysis.....	10
IV. Synthesis of Unity-Weighted Tree-Structured DEM Encoders.....	15
V. A Design Example.....	18
VI. Conclusion	24
VII. Acknowledgements.....	24
VIII. Figures.....	25
IX. References.....	31
Chapter 2 Suppression of Quantization-Induced Convergence Error in Pipelined	

ADCs With Harmonic Distortion Correction	35
I. Introduction.....	35
II. Background Information.....	38
A. Pipelined ADC and HDC Overview	38
III. Effect of Quantization Error on HDC Coefficient Estimation	45
IV. Solution To The Quantization Error Problem	51
A. Implementation Details.....	52
B. Theory of Operation.....	54
C. Calibration Sequence Choice	58
D. Simulation Results	59
V. Acknowledgements.....	60
VI. Figures.....	61
VII. Tables	66
VIII. References.....	67
 Chapter 3 Convergence Time Analysis for Harmonic Distortion Correction Used in Pipelined ADCs.....	 68
I. Introduction.....	68
II. Background Information	70
A. Pipelined ADC Overview	70
B. HDC Algorithm Overview	73
III. HDC Algorithm Convergence Time Analysis.....	77
A. Convergence time Analysis.....	79

IV. Simulation Results	85
V. Acknowledgements.....	86
VI. Figures.....	87
VII. Tables	89
VIII. Appendix.....	90
IX. References.....	106

LIST OF FIGURES

Figure 1	General DAC Architecture	25
Figure 2	Equivalent form of DAC in Figure 1	25
Figure 3	Equivalent form of DAC ^(u,w)	26
Figure 4	Signal processing performed by S ^(u,v,w)	26
Figure 5a	A type of tree-structured DEM encoder for a 13-level DAC	27
Figure 5b	A type of tree-structured DEM encoder for a 13-level DAC	27
Figure 6	Register transfer level circuitry that generates s ^(1,4,12) for c[n] ∈ {2,5,8,11}	28
Figure 7	Register transfer level circuitry that generates s ^(u,v,w) for (u,v,w) = (1,4,12)	28
Figure 8	Register transfer level circuitry that generates s ^(u,v,w) for (u,v,w) ≠ (1,4,12)	29
Figure 9	Block diagram of the second order ΔΣ ADC	29
Figure 10a	Power spectral density at the output of the second order ΔΣ ADC in Figure 9 without the DEM Encoder	30
Figure 10b	Power spectral density at the output of the second order ΔΣ ADC in Figure 9 with the tree-structured DEM Encoder shown in Figure 5	30
Figure 11	Example pipelined ADC	61
Figure 12	The kth pipelined ADC stage for k = 1, ..., 5 with labels that indicate variable names used throughout the paper	61
Figure 13	The example pipelined ADC with HDC applied to the first four stages	62
Figure 14	The kth pipelined ADC stage with HDC for k = 1, 2, 3, 4, with labels that indicate variable names used throughout the paper	62
Figure 15	Details of the HDC _k block	63
Figure 16	The five values of e _{ADC,2} [n] that occur when the input to flash ADC ₂ is (a) 2Δ - 4c ₁ [n] and (b) -4c ₁ [n] superimposed on a plot of the input-output characteristic of flash ADC ₂ (which contains a single threshold error)	63
Figure 17	Simulation results for pipelined ADC shown in Figure 3 with f _{in} = 0.39f _s and 0 dBFS sinusoidal input signal	64
Figure 18	Pipelined ADC configuration during estimation of α _{1,1} and α _{3,1} with HDC in the first stage and RD applied to the remaining stages	64
Figure 19	Details of the RD _k block	65
Figure 20	Simulation results for pipelined ADC shown in Figure 8 with f _{in} = 0.39f _s and 0 dBFS sinusoidal input signal and with representative Power Spectral Density plots of the output	65
Figure 21	Example pipelined ADC	87
Figure 22	The kth pipelined ADC stage for k = 1, ..., 5 with labels that indicate variable names used throughout the paper	87
Figure 23	Pipelined ADC configuration during estimation of α _{1,1} and α _{3,1} with	

	HDC in the first stage and RD applied to the remaining stages	88
Figure 24	A comparison plot of calculated standard deviation of $\tilde{\gamma}_{3,1}$ estimate (bold) vs. simulated standard deviation of $\tilde{\gamma}_{3,1}$, plotted as a function of the number of averaged samples by the HDC ₁ block.....	88

LIST OF TABLES

Table 1	All possible sets of values that $t_{1,j}[n]/A, t_{2,j}[n]/A, t_{3,j}[n]/A, t_{4,j}[n]/A$ can take on along with the corresponding values of $c_j[n]/A, c_{1,j}[n], c_{3,j}[n]$, and $c_{1,j}[n] \pm c_{3,j}[n]$	66
Table 2	All possible sets of values that $t_{1,j}[n]/A, t_{2,j}[n]/A, t_{3,j}[n]/A, t_{4,j}[n]/A$ can take on along with the corresponding values of $c_j[n]/A, c_{1,j}[n], c_{3,j}[n]$, and $c_{1,j}[n] \pm c_{3,j}[n]$	89

ACKNOWLEDGEMENTS

I would like to thank my academic advisor, Professor Ian Galton. He has been a great mentor and a good friend, supporting my intellectual and personal growth throughout these years. His guidance helped me deepen my understanding of the topics that I worked on, and his commitment to this work kept me motivated through the tough times.

I would also like to thank all my lab colleagues and friends for their friendship, understanding and encouragement. I want to offer my special thanks to Andrea Panigada, for the many hours he spent helping me navigate through the world of pipelined ADCs and for his incredible support.

And, this dissertation would not have been possible without the support of my parents. I want to thank them for believing in me, encouraging me and changing their lives so that I have an opportunity for a better one.

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 57, number 2, pages 313-322, February 2010. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

Chapter 2, in full, has been accepted for publication by IEEE Transactions on Circuits and Systems I: Regular Papers. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

Chapter 3, in part is currently being prepared for submission for publication of

the material. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

VITA

- 2012 Ph. D. in Electrical Engineering (Electronic Circuits and Systems),
University of California, San Diego
- 2009 M. S. in Electrical Engineering (Electronic Circuits and Systems),
University of California, San Diego
- 2006 B. S. in Engineering Physics,
University of California, San Diego

ABSTRACT OF THE DISSERTATION

A Generalized Tree-Structured DEM DAC and Enhanced Harmonic Distortion
Correction in Pipelined ADCs

by

Nevena Rakuljic

Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems)

University of California, San Diego, 2012

Professor Ian Galton, Chair

The first chapter of this dissertation discusses the tree-structured dynamic element matching (DEM) technique for unity-weighted, multi-bit digital to analog converters (DACs). In general, mismatches in nominally identical components of a unity-weighted multi-bit DAC introduce non-linear distortion at the output of the DAC. For this reason, a DEM encoder is usually employed in the implementation of unity-weighted multi-bit DACs, which objective is to permute the inputs to the nominally

identical components of the DAC, so that the non-linear distortion at the output is avoided or at least minimized. The best among such DEM techniques is the tree-structured DEM technique. However, prior to the work presented in Chapter 1, this technique could not be applied to unity-weighted multi-bit DACs with a number of components that is not a power of two.

The second chapter of this dissertation focuses on the topic of digital calibration of residue amplifiers in pipelined analog to digital converters (ADCs). Pipelined ADCs provide a high resolution digital representation of analog input signals. In recent years, many digital calibration techniques have been developed that enable the design of pipelined ADCs with low-power analog components that behave non-ideally. One such digital calibration technique is Harmonic Distortion Correction (HDC) which compensates for the non-ideal residue amplifier behavior. This technique is the best digital calibration technique known to the authors that addresses the problem of residue amplifiers. Nevertheless, the HDC technique, when implemented in the pipelined ADC, cannot accurately eliminate errors introduced by residue amplifiers under all pipelined ADC's input conditions. In particular, the problem in the implementation of the HDC technique arises due to the leakage of quantization error from the stages subsequent to the calibration stage. Chapter 2 presents an analysis of this problem and an all-digital solution which enables the HDC technique to properly operate regardless of the input signal level.

The third chapter of this dissertation analyzes the number of samples that need to be averaged by the HDC algorithm in order to reliably estimate, and therefore

eliminate, the residue amplifier errors from the pipelined ADC's output.

Chapter 1

Tree-Structured DEM DACs with Arbitrary Numbers of Levels

Abstract—Unity-weighted tree-structured dynamic element matching (DEM) DACs are widely used in delta-sigma ($\Delta\Sigma$) data converters to ensure that mismatches among nominally identical analog components give rise to shaped noise instead of nonlinear distortion. Tree-structured DEM DACs offer an advantage over other published DEM DACs in that the shaped noise from component mismatches can be made free of spurious tones. However, previously published unity-weighted tree-structured DEM DACs have the disadvantage that they require a power-of-two number of nominally identical 1-bit DACs. When applied to a $\Delta\Sigma$ data converter with a non-power-of-two number of quantization steps, this requires the DEM DAC to have a larger input range than needed by the $\Delta\Sigma$ data converter which wastes power and circuit area. This paper presents a generalized tree-structured DEM encoder applicable to DEM DACs with any number of 1-bit DACs, thereby avoiding this limitation.

I. INTRODUCTION

A typical unity-weighted dynamic element matching (DEM) digital-to-analog

Manuscript received January 13, 2009. First published June 02, 2009; current version published February 10, 2010. This work was supported by the National Science Foundation under Award 0515286, by the UCSD Center for Wireless Communications and by the University of California Discovery Program. This paper was recommended by Associate Editor G. Manganaro. The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail:galton@ece.ucsd.edu).
Digital Object Identifier 10.1109/TCSI.2009.2023931

converter (DAC) consists of a DEM encoder followed by a bank of N nominally identical 1-bit DACs, the outputs of which are summed to form the output of the DEM DAC. The DEM encoder maps the input sequence into N 1-bit sequences, each of which drives one of the 1-bit DACs. As described in the next section, the DEM encoder exploits flexibility in its choice of output bits each sample period to cause the error arising from mismatches among the 1-bit DACs to have a noise-like structure that is free of nonlinear distortion and spectrally shaped as appropriate for the application.

Unity-weighted DEM DACs are widely used in oversampling $\Delta\Sigma$ data converters, i.e., $\Delta\Sigma$ ADCs and $\Delta\Sigma$ DACs, to prevent component mismatches from degrading data converter precision [1] – [30]. The DACs within a typical $\Delta\Sigma$ data converter need only convert digital signals with a small number of levels, but they must not add significant error within the $\Delta\Sigma$ data converter’s relatively narrow signal band. Therefore, DEM is used to spectrally shape the error introduced by the DACs so as to suppress the error within the signal band.

Many types of unity-weighted DEM encoders have been published to date, one of which is the tree-structured DEM encoder [5], [9] – [12], [31], [32]. To the knowledge of the authors the tree-structured DEM encoder is the only of these in which the error caused by 1-bit DAC mismatches has been made spectrally shaped and free of spurious tones. This is a significant advantage in high-performance $\Delta\Sigma$ data converters, which tend to be used in applications which are highly sensitive to spurious tones. However, previously published tree structured DEM DACs have the

disadvantage that they require a power-of-two number of 1-bit DACs. When applied to a $\Delta\Sigma$ data converter with a non-power-of-two number of quantization steps, this requires the DEM DAC to have a larger input range than needed by the $\Delta\Sigma$ data converter which wastes power and circuit area. This paper presents a generalized tree-structured DEM encoder applicable to DEM DACs with any number of 1-bit DACs, thereby avoiding this limitation.

II. UNITY-WEIGHTED DEM DAC OVERVIEW

The purpose of a DAC is to convert a sequence of input values, $x[n]$, $n = 0, 1, 2, \dots$, represented as a sequence of digital codewords updated at times nT , where T is the duration of each sample period, into an analog waveform. In this paper, for a DAC with $N+1$ levels each codeword is interpreted by design convention to have a numerical value in the range

$$\left\{ -\frac{N}{2}\Delta, -\left(\frac{N}{2}-1\right)\Delta, -\left(\frac{N}{2}-2\right)\Delta, \dots, \frac{N}{2}\Delta \right\} \quad (1)$$

where Δ is the minimum *step-size* of $x[n]$. Ideally, the output of the DAC during the n th sample period, i.e. during the time interval $nT \leq t < (n+1)T$, is an analog pulse given by

$$y(t) = a(t - nT)x[n] \quad (2)$$

where $a(t)$ is called the *unit output pulse* and is zero outside of $0 \leq t < T$.

A general architecture for such a DAC is shown in Figure 1. It consists of an all-digital *encoder* followed by a bank of N 1-bit DACs, the outputs of which are

summed to form the DAC output waveform, $y(t)$. The encoder maps the sequence of input codewords into N 1-bit sequences denoted as $c_i[n]$, $i = 1, 2, \dots, N$, each of which takes on a value of 0 or 1 for each n . The encoder chooses its N output bits once per sample period under the constraint

$$x[n] = \Delta \sum_{i=1}^N (c_i[n] - \frac{1}{2}). \quad (3)$$

The output of the i th 1-bit DAC during the n th sample period is given by

$$y_i(t) = \begin{cases} \frac{\Delta}{2} a(t - nT) + e_{hi}(t - nT) & \text{if } c_i[n] = 1 \\ -\frac{\Delta}{2} a(t - nT) + e_{li}(t - nT) & \text{if } c_i[n] = 0 \end{cases} \quad (4)$$

where $e_{hi}(t)$ and $e_{li}(t)$ are *mismatch error pulses* that result from inadvertent process variations during IC fabrication. The only assumption made about $e_{hi}(t)$ and $e_{li}(t)$ in this paper is that they are zero outside of $0 \leq t < T$. The output of the overall DAC is given by

$$y(t) = \sum_{i=1}^N y_i(t). \quad (5)$$

The output of the i th 1-bit DAC during the n th sample period as given by (4) can equivalently be written as

$$y_i(t) = \alpha_i(t - nT) (c_i[n] - \frac{1}{2}) \Delta + \beta_i(t - nT) \quad (6)$$

where

$$\alpha_i(t) = a(t) + \frac{e_{hi}(t) - e_{li}(t)}{\Delta} \quad \text{and} \quad \beta_i(t) = \frac{e_{hi}(t) + e_{li}(t)}{2}. \quad (7)$$

This can be verified by substituting (7) into (6) to obtain (4).

Substituting (6) and (7) into (5) yields

$$y(t) = a(t - nT)x[n] + \beta(t - nT) + \varepsilon(t) \quad (8)$$

during the n th sample period, where

$$\beta(t) = \sum_{i=1}^N \beta_i(t) \quad \text{and} \quad (9)$$

$$\varepsilon(t) = \sum_{i=1}^N (c_i[n] - \frac{1}{2})(e_{hi}(t - nT) - e_{li}(t - nT)). \quad (10)$$

The mismatch error pulses are responsible for the terms $\beta(t - nT)$ and $\varepsilon(t)$ in (8). The first of these terms is a fixed pulse that repeats each sample period, independent of $x[n]$. Consequently, it results only in spurious tones at multiples of the sample frequency which do not degrade the signal-to-noise ratio (SNR) or the in-band spurious-free-dynamic-range (SFDR) of the DAC. In sampled DACs such as those implemented with switched capacitor circuits, it aliases down to a fixed offset, in which case it does not introduce any tones. In contrast, the $\varepsilon(t)$ term represents the dynamic error caused by the mismatch error pulses so it has the potential to degrade both the SNR and SFDR of the DAC.

Unfortunately, short of eliminating the mismatch error pulses, it is not possible to make $\varepsilon(t)$ zero. However, the encoder does have some control over the structure of $\varepsilon(t)$ because for each DAC input value except $x[n] = -N\Delta/2$ and $x[n] = N\Delta/2$ the encoder can choose among multiple sets of bits $\{c_1[n], c_2[n], \dots, c_N[n]\}$ that satisfy (3). For example, for any value of n at which $x[n] = -(N/2 - 1)\Delta$ the encoder can satisfy (3) by setting $c_i[n] = 1$ for any single value of i in the set $\{1, 2, \dots, N\}$ and set-

ting $c_j[n] = 0$ for all $j \neq i$. Encoders that dynamically exploit this flexibility to impart desirable properties to $\varepsilon(t)$ are called DEM encoders.

Since $\varepsilon(t)$ cannot be eliminated, its presence would be most tolerable if a DEM encoder could cause it to be a random process that is uncorrelated with $x[n]$, free of spurious tones, and spectrally shaped as appropriate for the application at all times regardless of $x[n]$ and the mismatch error pulses. A necessary condition for $\varepsilon(t)$ to have these properties is that its expectation during each sample period must be independent of $x[n]$. If this necessary condition were not satisfied, the expectation of $\varepsilon(t)$ would be a deterministic function of $x[n]$.

Unfortunately, $\varepsilon(t)$ does not satisfy this necessary condition. To see this, suppose that $x[n] = -N\Delta/2$ or $x[n] = N\Delta/2$ at some sample time n . Then to satisfy (3), the DEM encoder must set $c_i[n] = 0$ or $c_i[n] = 1$, respectively, for all $i = 1, 2, \dots, N$. It follows from (3) and (10) that for either of these cases $\varepsilon(t)$ is deterministic, so it is equal to its expectation and is given by

$$\varepsilon(t) = p(t - nT)x[n] \quad (11)$$

during the n th sample interval where

$$p(t) = \frac{1}{N\Delta} \sum_{i=1}^N [e_{hi}(t) - e_{li}(t)]. \quad (12)$$

Thus, the expectation of $\varepsilon(t)$ depends on $x[n]$ for any mismatch error pulses that do not cause $p(t)$ to be zero.

The above reasoning implies that at least a component of $\varepsilon(t)$ must be a deterministic function of $x[n]$. If the deterministic function were nonlinear, then the effect

of the mismatch error pulses would be to cause the DAC to introduce nonlinear distortion, which is unacceptable in most applications. Hence, the best possible outcome would be for the deterministic function to be linear. Given that (11) holds for the minimum and maximum values of $x[n]$, the only possible form for $\varepsilon(t)$ in which the deterministic function is linear is

$$\varepsilon(t) = p(t - nT)x[n] + e_{DAC}(t) \quad (13)$$

during the n th sample period, where $p(t)$ is given by (12), and $e_{DAC}(t)$ is a random process whose expectation is zero regardless of $x[n]$ and the mismatch error pulses. By definition $e_{DAC}(t) = 0$ during any sample interval in which $x[n] = -N\Delta/2$ or $x[n] = N\Delta/2$. Therefore, if the expectation of $e_{DAC}(t)$ were not zero over all sample intervals it would have the form of a nonlinear deterministic function of $x[n]$ plus a zero-mean random process.

It follows from (7) and (12) that $p(t) = \alpha(t) - a(t)$, where

$$\alpha(t) = \frac{1}{N} \sum_{i=1}^N \alpha_i(t). \quad (14)$$

Therefore, the analysis presented above implies that a necessary condition for a DEM DAC to avoid introducing nonlinear distortion is that its output during the n th sample interval is

$$y(t) = \alpha(t - nT)x[n] + \beta(t - nT) + e_{DAC}(t) \quad (15)$$

for each n , where $\alpha(t)$ is given by (14), $\beta(t)$ is given by (9), and $e_{DAC}(t)$ is a random process whose expectation is zero regardless of $x[n]$ and the mismatch error pulses. This is also a sufficient condition for $e_{DAC}(t)$ to be uncorrelated with $x[n]$. The objec-

tives of DEM are to achieve this condition and, as described above, to further ensure that $e_{DAC}(t)$ is free of spurious tones, and spectrally shaped as appropriate for the application regardless of $x[n]$ and the mismatch error pulses.

The three components of the DAC's output signal in (15) are referred to as the *signal pulse sequence*, the *offset pulse sequence*, and the *DAC noise*, respectively [33]. The mismatch error pulses cause $\alpha(t)$ to deviate somewhat from the ideal unit output pulse, $a(t)$, but in most applications this is not a serious problem because it has little effect on the SNR or SFDR of the overall DAC. As described above, the offset pulse sequence does not degrade the SNR or the in-band SFDR of the overall DAC, and the objective of DEM is to render the DAC noise tolerable for the given application.

III. DECOMPOSITION OF ARBITRARY DEM ENCODERS INTO TREE STRUCTURE

A. Preliminary Definitions

When considering the behavior of a DAC in the context of a signal processing system such as a $\Delta\Sigma$ data converter, it is convenient to interpret the sequence of input codewords to have values given by (1) as described above. However, when considering the operation of the DEM encoder, it is convenient to consider each codeword to represent the number of 1-bit DACs whose input bits must be set high during that sample interval, i.e.

$$c[n] = \frac{x[n]}{\Delta} + \frac{N}{2}. \quad (16)$$

Therefore (3) is equivalent to

$$c[n] = \sum_{i=1}^N c_i[n]. \quad (17)$$

and (15) can be written as

$$y(t) = \alpha(t - nT)c[n]\Delta + \beta_c(t - nT) + e_{DAC}(t) \quad (18)$$

during the n th sample period, where

$$\beta_c(t) = \beta(t) - \alpha(t) \frac{N}{2} \Delta. \quad (19)$$

In order to simplify the subsequent analysis, the following definition from [34] is used.

DAC^(u,w) Definition: For any integers u and w that satisfy $1 \leq u \leq w \leq N$, DAC^(u,w) consists of an encoder followed by the u th through w th 1-bit DACs of the DAC shown in Figure 1. The encoder maps a digital input sequence given by

$$c^{(u,w)}[n] = \sum_{i=u}^w c_i[n] \quad (20)$$

to the same 1-bit sequences $c_u[n]$, $c_{u+1}[n]$, ..., $c_w[n]$ generated by the encoder shown in Figure 1. The output of DAC^(u,w) during the n th sample period is

$$y^{(u,w)}(t) = \sum_{i=u}^w y_i(t). \quad (21)$$

Following an analysis almost identical to that presented in the previous section (6) and (21) imply that

$$y^{(u,w)}(t) = \alpha^{(u,w)}(t-nT)c^{(u,w)}[n]\Delta + \beta_c^{(u,w)}(t-nT) + e_{DAC}^{(u,w)}(t) \quad (22)$$

where

$$\alpha^{(u,w)}(t) = \frac{1}{w-u+1} \sum_{i=u}^w \alpha_i(t) \quad \text{and} \quad (23)$$

$$\beta_c^{(u,w)}(t) = -\frac{\Delta}{2}(w-u+1)\alpha^{(u,w)}(t) + \sum_{i=u}^w \beta_i(t). \quad (24)$$

Note that for the special case of $u = w$, $DAC^{(u,w)}$ denotes the u th 1-bit DAC, and that $c^{(u,w)}[n] = c_u[n]$. Furthermore, a comparison of (6) and (22) implies that

$$e_{DAC}^{(u,w)}(t) = 0. \quad (25)$$

This is reasonable given that a 1-bit DAC has only two input levels; the mismatch error pulses give rise to a pulse shape error and an offset pulse, but no DAC noise as defined in the previous section.

B. Decomposition Analysis

It follows from (5), (16), (17), (20), and (21) that any DAC of the form shown in Figure 1 can be redrawn in the equivalent form shown in Figure 2 for any $g \in \{1, 2, \dots, N-1\}$. The equivalent form consists of a digital block labeled $S^{(1,g,N)}$, called a *switching block*, and two sub-DACs, $DAC^{(1,g)}$ and $DAC^{(g+1,N)}$, the outputs of which are summed to form the overall DAC output.

The $S^{(1,g,N)}$ switching block converts the $c[n]$ sequence into the input sequences to $DAC^{(1,g)}$ and $DAC^{(g+1,N)}$, i.e., $c^{(1,g)}[n]$ and $c^{(g+1,N)}[n]$, respectively. It follows from (20) that the bottom and top output sequences from the $S^{(1,g,N)}$ switching block

must be

$$c^{(1,g)}[n] = \sum_{i=1}^g c_i[n], \quad \text{and} \quad c^{(g+1,N)}[n] = \sum_{i=g+1}^N c_i[n], \quad (26)$$

respectively. Equivalently, these sequences can be rewritten as

$$c^{(1,g)}[n] = \frac{g}{N} c[n] + s^{(1,g,N)}[n], \quad \text{and} \quad c^{(g+1,N)}[n] = \frac{N-g}{N} c[n] - s^{(1,g,N)}[n], \quad (27)$$

respectively, where $s^{(1,g,N)}[n]$ is called a *switching sequence* and is given by

$$s^{(1,g,N)}[n] = \frac{N-g}{N} \sum_{i=1}^g c_i[n] - \frac{g}{N} \sum_{i=g+1}^N c_i[n]. \quad (28)$$

This can be verified by substituting (28) into (27) to obtain (26). It follows that the $S^{(1,g,N)}$ switching block can be viewed as a device that somehow generates $s^{(1,g,N)}[n]$ and uses it with (27) to obtain the switching block's two output sequences as functions of its input sequence.

It is next shown that the $s^{(1,g,N)}[n]$ switching sequence plays a key role in determining the behavior of the DAC noise. Given that

$$y(t) = y^{(1,g)}(t) + y^{(g+1,N)}(t), \quad (29)$$

(22) implies that during the n th sample period

$$\begin{aligned} y(t) = & \alpha^{(1,g)}(t-nT) c^{(1,g)}[n] \Delta + \beta_c^{(1,g)}(t-nT) + e_{DAC}^{(1,g)}(t) \\ & + \alpha^{(g+1,N)}(t-nT) c^{(g+1,N)}[n] \Delta + \beta_c^{(g+1,N)}(t-nT) + e_{DAC}^{(g+1,N)}(t). \end{aligned} \quad (30)$$

With (23), (24), and (27) this can be rewritten as

$$\begin{aligned} y(t) = & \alpha(t-nT) c[n] \Delta + \beta_c(t-nT) \\ & + s^{(1,g,N)}[n] \Delta^{(1,g,N)}(t-nT) + e_{DAC}^{(1,g)}(t) + e_{DAC}^{(g+1,N)}(t) \end{aligned} \quad (31)$$

where

$$\Delta^{(1,g,N)}(t) = [\alpha^{(1,g)}(t) - \alpha^{(g+1,N)}(t)]\Delta. \quad (32)$$

Comparison to (18) indicates that

$$e_{DAC}(t) = s^{(1,g,N)}[n]\Delta^{(1,g,N)}(t-nT) + e_{DAC}^{(1,g)}(t) + e_{DAC}^{(g+1,N)}(t) \quad (33)$$

during the n th sample period.

The above analysis trivially can be generalized to any $DAC^{(u,w)}$, where $1 \leq u < w \leq N$. Specifically, as illustrated in Figure 3, $DAC^{(u,w)}$ can be decomposed into an $S^{(u,v,w)}$ switching block and two sub-DACs, $DAC^{(u,v)}$ and $DAC^{(v+1,w)}$, for any $v \in \{u, u+1, \dots, w-1\}$. It follows from almost identical reasoning as used to obtain (27), (28), and (33) that the bottom and top outputs of the $S^{(u,v,w)}$ switching block are

$$\begin{aligned} c^{(u,v)}[n] &= \frac{v-u+1}{w-u+1} c^{(u,w)}[n] + s^{(u,v,w)}[n], \quad \text{and} \\ c^{(v+1,w)}[n] &= \frac{w-v}{w-u+1} c^{(u,w)}[n] - s^{(u,v,w)}[n], \end{aligned} \quad (34)$$

respectively, where $s^{(u,v,w)}[n]$ is a switching sequence and is given by

$$s^{(u,v,w)}[n] = \frac{w-v}{w-u+1} \sum_{i=u}^v c_i[n] - \frac{v-u+1}{w-u+1} \sum_{i=v+1}^w c_i[n]. \quad (35)$$

During the n th sample period (22) holds with

$$e_{DAC}^{(u,w)}(t) = s^{(u,v,w)}[n]\Delta^{(u,v,w)}(t-nT) + e_{DAC}^{(u,v)}(t) + e_{DAC}^{(v+1,w)}(t) \quad (36)$$

where

$$\Delta^{(u,v,w)}(t) = [\alpha^{(u,v)}(t) - \alpha^{(v+1,w)}(t)]\Delta. \quad (37)$$

Figure 4 shows the signal processing performed by switching block $S^{(u,v,w)}$, where

$$G^{(u,v,w)} = \frac{w-v}{w-u+1}.$$

The above analysis shows that i) any DAC of the form shown in Figure 1 can be decomposed as shown in Figure 2, ii) if $g \geq 2$ then $\text{DAC}^{(1,g)}$ can be decomposed as shown in Figure 3 for $u = 1$, $w = g$, and any $v = v_1$ where $v_1 \in \{1, 2, \dots, g-1\}$, and iii) if $g \leq N-1$ then $\text{DAC}^{(g+1,N)}$ can be decomposed as shown in Figure 3 for $u = g+1$, $w = N$, and any $v = v_2$ where $v_2 \in \{g+1, g+2, \dots, N-1\}$. Using results (ii) and (iii) above, the last two terms in (33) can each be expanded via (36) to obtain

$$\begin{aligned} e_{DAC}(t) = & s^{(1,g,N)}[n]\Delta^{(1,g,N)}(t-nT) + s^{(1,v_1,g)}[n]\Delta^{(1,v_1,g)}(t-nT) + e_{DAC}^{(1,v_1)}(t) \\ & + e_{DAC}^{(v_1+1,g)}(t) + s^{(g+1,v_2,N)}[n]\Delta^{(g+1,v_2,N)}(t-nT) + e_{DAC}^{(g+1,v_2)}(t) + e_{DAC}^{(v_2+1,N)}(t). \end{aligned} \quad (38)$$

This decomposition process can be continued recursively, each time replacing a sub-DAC of the form $\text{DAC}^{(u,w)}$ in which $w \geq u + 1$ by a new switching block $S^{(u,v,w)}$ and two new sub-DACs, $\text{DAC}^{(u,v)}$ and $\text{DAC}^{(v+1,w)}$. The recursive decomposition can be continued until the DAC of Figure 1 has been transformed into a tree of switching blocks that drives N sub-DACs of the form $\text{DAC}^{(u,u)}$ for $u = 1, 2, \dots, N$. By definition $\text{DAC}^{(u,u)}$ is just the u th 1-bit DAC, so the above analysis indicates that any encoder (DEM or otherwise) which satisfies (17) is equivalent to a tree of switching blocks with switching sequences given by (35). Furthermore, since each recursion allows one of the $e_{DAC}^{(u,w)}(t)$ terms in the $e_{DAC}(t)$ expression obtained from the previous recursion step to be expanded via (36), and $e_{DAC}^{(u,u)}(t) = 0$ for all u , it follows that

$$e_{DAC}(t) = \sum_{u,v,w} s^{(u,v,w)}[n]\Delta^{(u,v,w)}(t-nT) \quad (39)$$

during the n th sample period where the sum in (39) is taken over all values of u , v , and w used during the recursive decomposition process. By definition, $\Delta^{(u,v,w)}(t)$ for

each u , v , and w is a fixed pulse that is zero outside of $0 \leq t < T$, so (39) implies that the statistical properties of $e_{DAC}(t)$ are determined by the switching sequences.

At each step in the decomposition process, whenever $w - u \geq 2$ more than one choice exists for v . Therefore, a given encoder can be decomposed into several equivalent trees of switching blocks. Each such tree of switching blocks is called a *tree-structured encoder*. Figures 5a and 5b show examples of tree-structured encoders obtained by decomposing DACs of the form shown in Figure 1 with $N = 12$ and $N = 9$ 1-bit DACs, respectively.

The tree-structured encoders shown in Figure 5 each contain $N-1$ switching blocks, and the following argument indicates that this result holds in general, i.e., that all tree-structured encoders contain exactly $N-1$ switching blocks. As described above, each of the recursion steps used to generate a given tree-structured encoder replaces a DAC of the form $DAC^{(u,w)}$ where $1 \leq u < w \leq N$ by a switching block and two new sub-DACs, $DAC^{(u,v)}$ and $DAC^{(v+1,w)}$. This places a dividing line between the v th and $(v+1)$ th 1-bit DACs, and assigns all of the 1-bit DACs in $DAC^{(u,w)}$ below this dividing line to $DAC^{(u,v)}$ and all those above the dividing line to $DAC^{(v+1,w)}$. The recursion process ends when all sub-DACs are of the form $DAC^{(u,u)}$ for $1 \leq u \leq N$, i.e., when a dividing line has been placed between every pair of 1-bit DACs. A bank of N 1-bit DACs can contain up to $N-1$ such dividing lines, so exactly $N-1$ recursion steps are required to transform the encoder shown in Figure 1 into a tree-structured encoder. Hence, the tree-structured encoder contains $N-1$ switching blocks.

The analysis presented above starts with an arbitrary encoder that satisfies

(17) and shows that there exist multiple equivalent tree-structured encoders with switching sequences specified in terms of the 1-bit output sequences from the original encoder. Therefore, it implies that each tree-structured encoder is completely general in that with the appropriate choice of switching sequences it can mimic any given encoder that satisfies (17). Furthermore, (39) implies that the switching sequences specify the dynamics of the DAC noise. Hence, the derivation uses the tree-structured encoder as an analysis tool.

IV. SYNTHESIS OF UNITY-WEIGHTED TREE-STRUCTURED DEM ENCODERS

The results of the previous section are extended in this section to provide a method with which to synthesize tree-structured DEM encoders that have desired DAC noise properties. The synthesis method involves choosing one of the possible trees of switching blocks derived in the previous section, and then designing switching sequences that result in DAC noise with desired properties under the constraint that (17) is satisfied.

As in the previous section, first consider the $S^{(1,g,N)}$ switching block. It follows from (27) that the outputs of the switching block satisfy

$$c^{(1,g)}[n] + c^{(g+1,N)}[n] = c[n] \quad (40)$$

and that for each n the value of $s^{(1,g,N)}[n]$ determines how $c[n]$ is distributed between $c^{(1,g)}[n]$ and $c^{(g+1,N)}[n]$. Given that $c_i[n] \in \{0, 1\}$ for each n , (26) implies that

$$c^{(1,g)}[n] \in \{0, 1, \dots, g\} \quad \text{and} \quad c^{(g+1,N)}[n] \in \{0, 1, \dots, N-g\}. \quad (41)$$

Therefore, for any value of n at which $c[n] = 0$ the only way to satisfy (40) and (41) is to have $c^{(1,g)}[n] = 0$ and $c^{(g+1,N)}[n] = 0$, which implies that $s^{(1,g,N)}[n]$ must be zero. Similarly, for any value of n at which $c[n] = N$ the only way to satisfy (40) and (41) is to have $c^{(1,g)}[n] = g$ and $c^{(g+1,N)}[n] = N-g$, which implies that $s^{(1,g,N)}[n]$ must be zero.

For each other possible value of $c[n]$, i.e., each value in the set $\{1, 2, \dots, N-1\}$, there exist at least two valid choices of $s^{(1,g,N)}[n]$ because there are at least two different choices of $c^{(1,g)}[n]$ and $c^{(g+1,N)}[n]$ that satisfy (40) and (41). These two valid choices of $s^{(1,g,N)}[n]$ are

$$s^{(1,g,N)}[n] = \left\langle \frac{N-g}{N} c[n] \right\rangle \quad \text{and} \quad s^{(1,g,N)}[n] = \left\langle \frac{N-g}{N} c[n] \right\rangle - 1 \quad (42)$$

where $\langle b \rangle = b - \lfloor b \rfloor$ denotes the fractional part of b and $\lfloor b \rfloor$ denotes the largest integer less than or equal to b . This can be verified by substituting the left and right equations of (42) into (27). Substituting the left equation of (42) into (27) yields

$$c^{(1,g)}[n] = c[n] - \left\lfloor \frac{N-g}{N} c[n] \right\rfloor, \quad \text{and} \quad c^{(g+1,N)}[n] = \left\lfloor \frac{N-g}{N} c[n] \right\rfloor. \quad (43)$$

and substituting the right equation of (42) into (27) yields

$$c^{(1,g)}[n] = c[n] - \left\lfloor \frac{N-g}{N} c[n] \right\rfloor - 1, \quad \text{and} \quad c^{(g+1,N)}[n] = \left\lfloor \frac{N-g}{N} c[n] \right\rfloor + 1. \quad (44)$$

In both cases (41) is satisfied whenever $c[n] \in \{1, 2, \dots, N-1\}$ as required. Thus for each n at which $c[n] \in \{1, 2, \dots, N-1\}$ there must be sets of bits $\{c_1[n], c_2[n], \dots, c_N[n]\}$ that cause (28) to take on the values implied by (42).

The above analysis trivially can be generalized to any switching block $S^{(u,v,w)}$, where $1 \leq u < w \leq N$ with only changes in the notation. Specifically, following identical reasoning as above indicates that valid choices of $s^{(u,v,w)}[n]$ are

$$s^{(u,v,w)}[n] = \begin{cases} 0, & \text{if } c^{(u,w)}[n] \in \{0, w-u+1\}, \\ m \text{ or } m-1, & \text{otherwise} \end{cases} \quad (45)$$

where

$$m = \left\langle \frac{w-v}{w-u+1} c^{(u,w)}[n] \right\rangle.$$

Thus for each n there must be sets of bits $\{c_u[n], c_{u+1}[n], \dots, c_w[n]\}$ that cause (35) to take on the values implied by (45). Conversely, if a tree-structured encoder is designed in which all the switching sequences satisfy (45), then the N 1-bit output sequences from the encoder will satisfy (17) and the DAC noise will satisfy (39).

Note that (45) is not a general expression because it does not represent all possible values that can be assumed by (35). This implies that tree-structured encoders with switching sequences that satisfy (45) are not capable of mimicking any conceivable encoder that satisfies (17). Nevertheless, as shown below and demonstrated in the next section, the switching sequence values implied by (45) are sufficient to achieve desirable DAC noise properties.

As shown in Section II, a necessary condition for a DEM DAC to avoid introducing nonlinear distortion is for the expectation of $e_{DAC}(t)$ to be zero for all t , regardless of the mismatch error pulses and $x[n]$. A tree-structured DAC with switching sequences that satisfy (45) can achieve this necessary condition because for each n ,

regardless of the value of $c[n]$, every switching sequence that satisfies (45) has a value that is zero or is either of two known non-zero values, one of which is positive and the other negative. Therefore each switching block can exercise its choice of possible switching sequence values to ensure that the expectation of each switching sequence is zero. In this case, (39) implies that $e_{DAC}(t)$ satisfies the necessary condition.

Furthermore, for each n at which (45) allows the switching block to have a choice of two non-zero values with opposite signs, the choice can be made without regard to the switching block's input sequence, and therefore without regard to $c[n]$. This makes it possible to use switching sequences that are spectrally shaped random processes which are uncorrelated with $c[n]$. An example of such a DEM DAC is presented in the next section.

V. A DESIGN EXAMPLE

The design of a 13-level DEM DAC with a tree-structured DEM encoder for use in a second-order $\Delta\Sigma$ ADC is described in this section. The objective of the DEM DAC for this application is to cause the DAC noise to be a random process with an expectation of zero, to be uncorrelated with $c[n]$, to be free of spurious tones, and to be highpass shaped, all regardless of the mismatch error pulses. The tree-structured DEM encoder described in Section III and shown in Figure 5a with switching blocks that perform the signal processing operations shown in Figure 4 is used as the starting point for the design. The design tasks are to choose appropriate switching sequences and devise digital logic that generates the switching sequences.

The PSD of a DAC waveform can be estimated in the laboratory using a spectrum analyzer, or, analogously, in simulation using periodogram analysis [35]. Therefore, the spectral properties of the switching sequences are derived below in terms of their periodograms. The length- L periodogram of $s^{(u,v,w)}[n]$ is given by

$$I_{s^{(u,v,w)},L}(\omega) = \frac{1}{L} \left| \sum_{n=0}^{L-1} s^{(u,v,w)}[n] e^{-j\omega n} \right|^2, \quad (46)$$

which can be written equivalently as

$$I_{s^{(u,v,w)},L}(\omega) = \frac{1}{L} \sum_{n=0}^{L-1} \sum_{m=0}^{L-1} s^{(u,v,w)}[n] s^{(u,v,w)}[m] e^{-j\omega(n-m)}. \quad (47)$$

It is well known that in certain cases the expectation of the periodogram converges to the true PSD function in the limit as $L \rightarrow \infty$, but in a DAC application this is not a requirement, or even relevant to the measured performance.

First consider the $s^{(1,4,12)}[n]$ switching sequence. Evaluation of (45) for all possible values of $c[n]$ yields

$$s^{(1,4,12)}[n] = \begin{cases} 0 & \text{if } c[n] \in \{0, 12\} \\ 0 \text{ or } -1 & \text{if } c[n] \in \{3, 6, 9\} \\ \frac{1}{3} \text{ or } -\frac{2}{3} & \text{if } c[n] \in \{2, 5, 8, 11\} \\ \frac{2}{3} \text{ or } -\frac{1}{3} & \text{if } c[n] \in \{1, 4, 7, 10\} \end{cases}. \quad (48)$$

It follows from (48) that for the expectation of $s^{(1,4,12)}[n]$ to be zero regardless of $c[n]$ (and therefore to be uncorrelated with $c[n]$), the probability distribution of $s^{(1,4,12)}[n]$ must satisfy

$$\mathrm{P}\left(s^{(1,4,12)}[n] = 0\right) = 1 \text{ and } \mathrm{P}\left(s^{(1,4,12)}[n] = -1\right) = 0$$

when $c[n] \in \{0, 3, 6, 9, 12\}$,

$$\mathrm{P}\left(s^{(1,4,12)}[n] = \frac{1}{3}\right) = \frac{2}{3} \text{ and } \mathrm{P}\left(s^{(1,4,12)}[n] = -\frac{2}{3}\right) = \frac{1}{3}$$

when $c[n] \in \{2, 5, 8, 11\}$, and

$$\mathrm{P}\left(s^{(1,4,12)}[n] = \frac{2}{3}\right) = \frac{1}{3} \text{ and } \mathrm{P}\left(s^{(1,4,12)}[n] = -\frac{1}{3}\right) = \frac{2}{3} \quad (49)$$

when $c[n] \in \{1, 4, 7, 10\}$, where $\mathrm{P}(s^{(1,4,12)}[n] = \lambda)$ denotes the probability that $s^{(1,4,12)}[n]$ is equal to λ .

As a special case, first suppose that $c[n] \in \{2, 5, 8, 11\}$ for all n . Then one way to satisfy (49) is to let $(s^{(1,4,12)}[3k], s^{(1,4,12)}[3k+1], s^{(1,4,12)}[3k+2])$ be an independent sequence of random variable triples that take on values of $(1/3, 1/3, -2/3)$, $(1/3, -2/3, 1/3)$ and $(-2/3, 1/3, 1/3)$ with equal probability. The sum of terms in each triple is zero, so it follows from (46) that $I_{s^{(u,v,w)},L}(0) < 1/L$ which implies that the expectation of $I_{s^{(u,v,w)},L}(\omega)$ goes to zero at $\omega = 0$ as $L \rightarrow \infty$. Although the expectation of $I_{s^{(u,v,w)},L}(\omega)$ does not go to zero as $L \rightarrow \infty$ when $\omega \neq 0$, (47) and the independence of the triples imply that the expectation of $I_{s^{(u,v,w)},L}(\omega)$ is uniformly bounded for all L and ω . Hence, the sequence is highpass shaped and is free of spurious tones as desired.

A digital logic block that generates $s^{(1,4,12)}[n]$ for this special case is shown in Figure 6. Three flip-flops preloaded with bit values of 1, 1, and 0, respectively, are configured as a re-circulating shift register clocked once per DAC sample interval. Thus, the Q output of the left-most flip-flop is the periodic sequence 1, 1, 0, 1, 1, 0, ..., and the Q outputs of the middle and right-most flip flops are the same sequence

except delayed by one and two DAC sample intervals, respectively. A three-to-one MUX selects as its output one of the three flip-flop outputs based on a pseudo-random number that is updated once every three DAC sample intervals. Each pseudo-random number is chosen independently and takes on values of 0, 1, and 2 with equal probability. A value of $2/3$ is subtracted from the output of the multiplexer to cause the final sequence to take on values of $1/3$, $1/3$, and $-2/3$, as required.

The other special cases can be handled similarly. If $c[n] \in \{1, 4, 7, 10\}$ for all n , the same strategy can be used except with triples that take on values of $(-1/3, -1/3, 2/3)$, $(-1/3, 2/3, -1/3)$ and $(2/3, -1/3, -1/3)$ with equal probability. It is straightforward to verify that the digital logic block shown in Figure 6 with its output multiplied by -1 can be used to generate $s^{(1,4,12)}[n]$ for this special case. Alternatively, if $c[n] \in \{0, 3, 6, 9, 12\}$ for all n , then $s^{(1,4,12)}[n] = 0$ can be used. This satisfies (49) and has the benefit that it does not contribute at all to the DAC noise.

In general, any $c[n]$ sequence can be constructed by interlacing subsequences corresponding to the three special cases described above, and the $s^{(1,4,12)}[n]$ switching sequence can be formed by correspondingly interlacing the switching sequences described above for each subsequence. Since each of the interlaced switching sequences is dc-free, highpass shaped, and free of spurious tones, $s^{(1,4,12)}[n]$ inherits these properties.

A digital logic block that generates the $s^{(1,4,12)}[n]$ is shown in Figure 7. It generates switching sequences for the three special cases described above and combines them to implement the interlacing operation. Therefore, the structure of Figure 4 with

$G^{(1,4,12)} = 2/3$ and the logic block shown in Figure 6 and Figure 7 make up the $S^{(1,4,12)}$ switching block.

Applying the same procedure to design the remaining switching sequences yields

$$s^{(u,v,w)}[n] = \begin{cases} 0, & \text{if } c^{(u,w)}[n] \text{ is even,} \\ \lambda^{(u,v,w)}[n], & \text{if } c^{(u,w)}[n] \text{ is odd,} \end{cases} \quad (50)$$

for each $(u, v, w) \neq (1, 4, 12)$, where $\lambda^{(u,v,w)}[n]$ is a highpass shaped random sequence each sample of which takes on values of $\pm 1/2$ with equal probability. A digital block that implements (50) is shown in Figure 8. Each of the corresponding switching blocks consists of the structure of Figure 4 with $G^{(u,v,w)} = 1/2$ and the digital block shown in Figure 8. Although the notation is slightly different to allow for the generalizations presented in this paper, it is straightforward to verify that for this special case the switching block is equivalent to that presented in [36] for unity-weighted DACs with a power-of-two number of 1-bit DACs.

When used in a $\Delta\Sigma$ ADC, any input-output latency imposed by the DEM encoder adds to the delay around the feedback paths within the $\Delta\Sigma$ ADC so it must be considered when designing the $\Delta\Sigma$ ADC. Although a tree-structured DEM encoder contains clocked components (e.g., the components shown in Figures 7 and 8), these components are not in the data path; they are only used to generate the switching sequences, so they only need to be fast enough to generate switching sequence samples at the sample-rate of the DEM DAC. In contrast, the latency of the DEM encoder is determined by how fast the operations shown in Figure 4 occur within each switching

block, and the largest number of cascaded switching blocks within the DEM encoder.

In general, the largest number of cascaded switching blocks within a tree-structured DEM encoder for a given number, N , of 1-bit DACs depends on the choices made during the recursion process described in Section III. It is straightforward to verify that the minimum value of this number is the smallest integer greater than or equal to $\log_2 N$ and that this minimum value is achieved by at least one of the possible tree-structures. The DEM encoders shown in Figure 5 are such examples.

If necessary, the latency of the DEM encoder can be reduced at the expense of increased complexity. One approach is to represent $c[n]$ as a thermometer code and flatten the tree structure into an equivalent single layer of transmission gates as described in [37]. Other approaches involve modifying the binary number formats used by the individual switching blocks to reduce latency as described in [36].

Figure 9 shows the block diagram of a second order $\Delta\Sigma$ ADC that contains two 13-level DEM DACs of the type designed above. As is common practice in such ADCs, both DEM DACs share the same DEM encoder to save circuit area [13]. Figure 10a and Figure 10b show output spectra from a computer simulation of the $\Delta\Sigma$ ADC with ideal components except for 1-bit DAC mismatches. The simulated 1-bit DAC mismatches were chosen from a Gaussian distribution with a standard deviation of 1%. The simulated input sequence is the sum of a full-scale sinusoid and a small amount of white noise to act as dither [38]. Figure 10a shows the output spectrum from the $\Delta\Sigma$ ADC simulated without the DEM encoder. As expected, the 1-bit DAC mismatches introduce significant distortion in this case. Figure 10b shows the output

spectrum from the $\Delta\Sigma$ ADC simulated with the DEM encoder described above. As expected, the 1-bit DAC mismatches give rise to highpass shaped DAC noise free of spurious tones.

VI. CONCLUSION

A generalized tree-structured DEM encoder that can drive any number of 1-bit DACs is presented in this paper. It removes the limitation of prior work which requires the number of 1-bit DACs to be a power of two. The analysis section of the paper proves that tree-structured encoders with appropriately chosen switching sequences can mimic the behavior of any DAC encoder. The synthesis section presents a way to design switching sequences for any tree-structured DEM encoder such that the DAC noise arising from mismatches is uncorrelated with the DAC's input sequence, spectrally shaped, and free of spurious tones. The last section of the paper demonstrates the key points of the paper in the context of a second order $\Delta\Sigma$ ADC.

VII. ACKNOWLEDGEMENTS

Chapter 1, in full, has been published in the IEEE Transactions on Circuits and Systems I: Regular Papers, volume 57, number 2, pages 313-322, February 2010. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

VIII. FIGURES

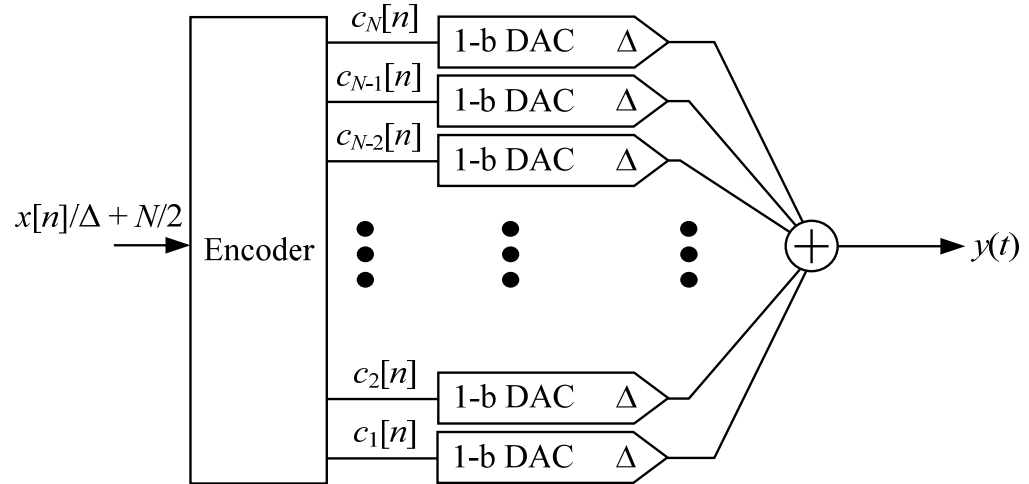


Figure 1 General DAC Architecture.

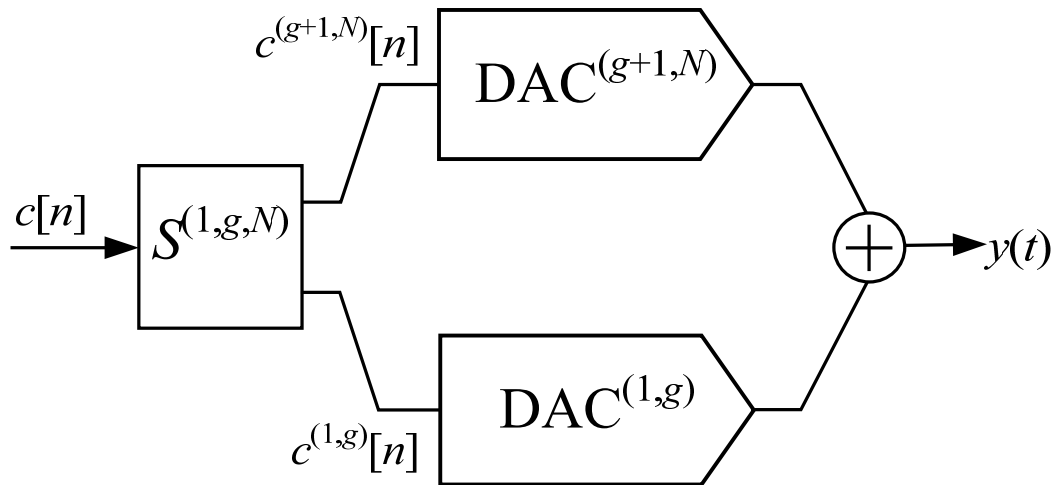


Figure 2 Equivalent form of DAC in Figure 1.

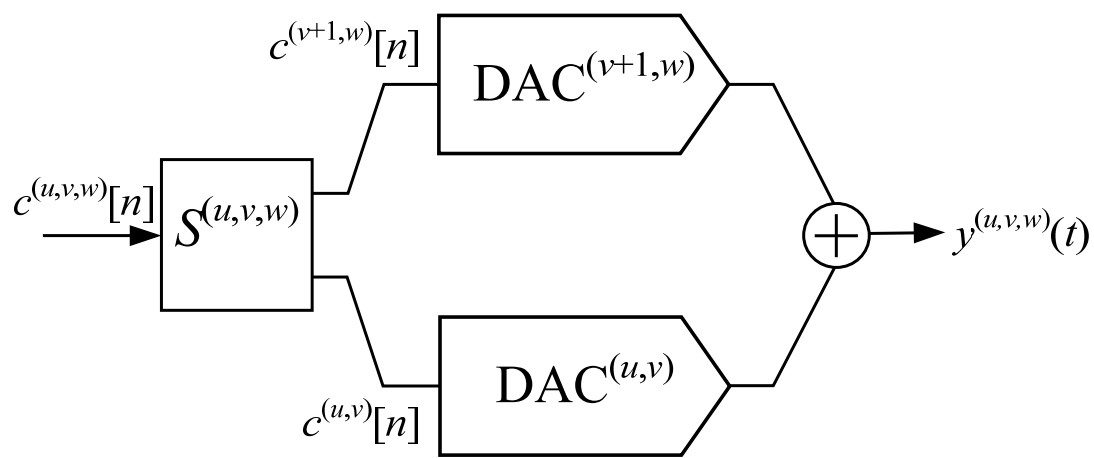


Figure 3 Equivalent form of $\text{DAC}^{(u,w)}$.

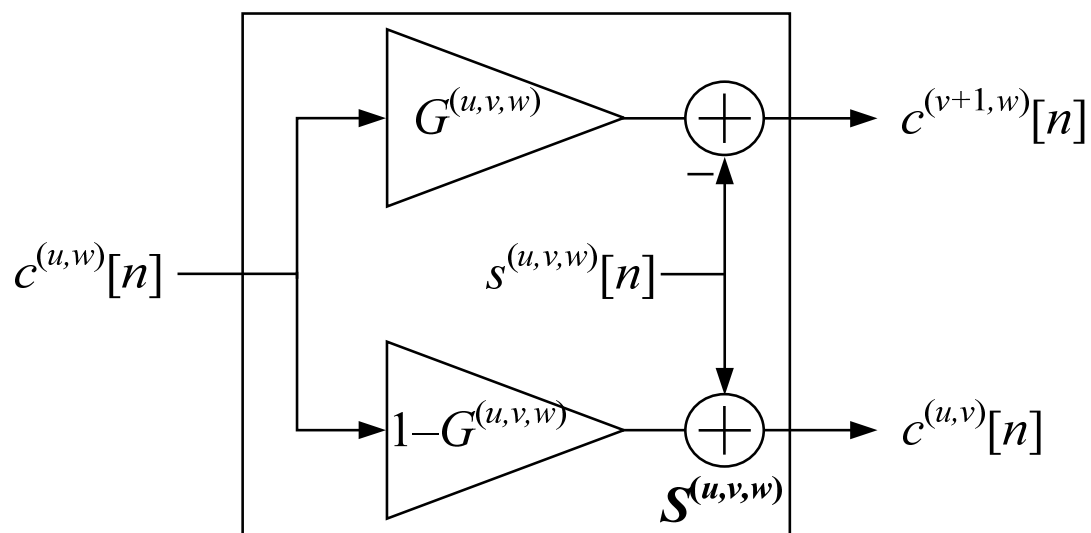


Figure 4 Signal processing performed by $S^{(u,v,w)}$.

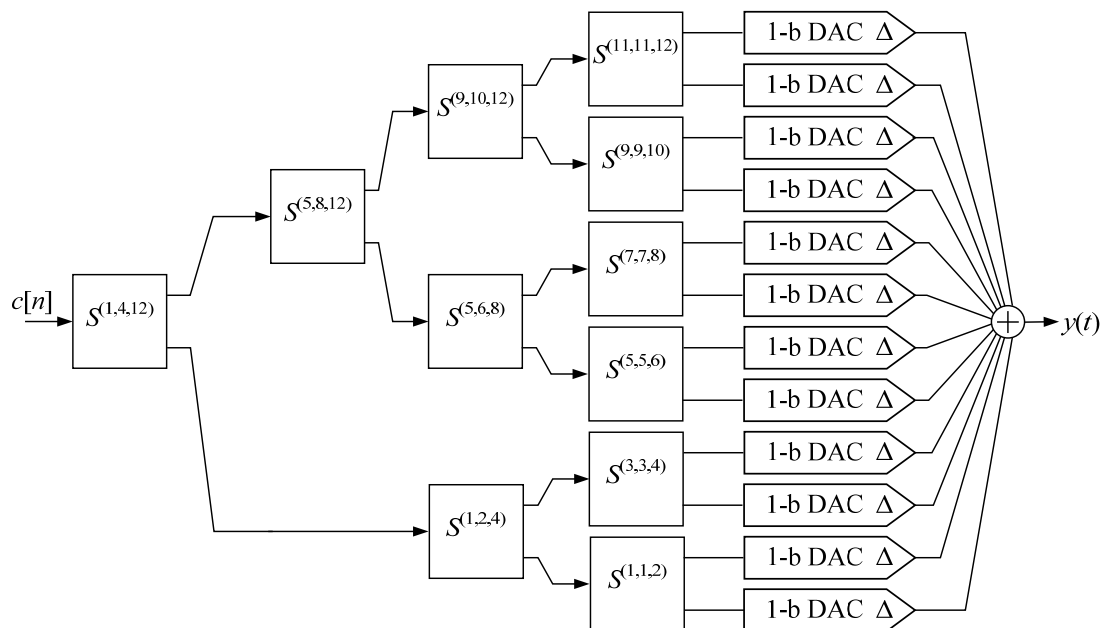


Figure 5a A type of tree-structured DEM encoder for a 13-level DAC.

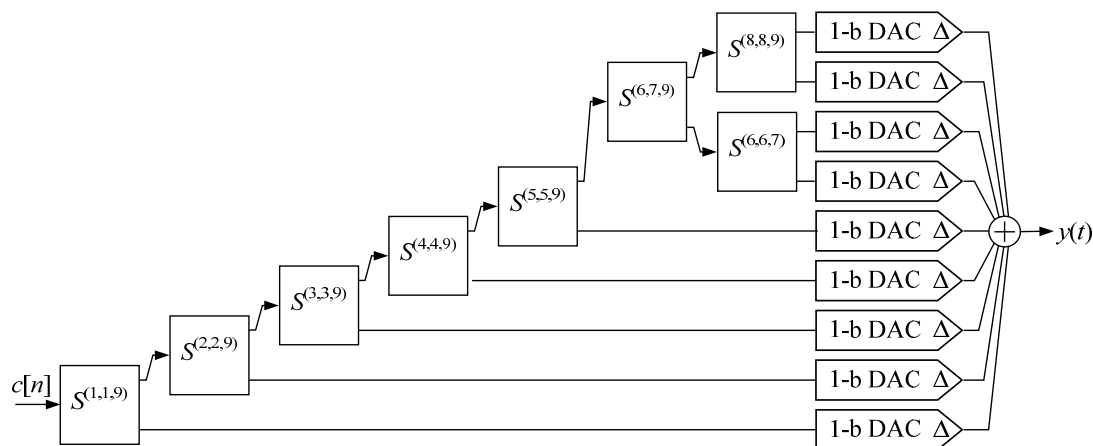


Figure 5b A type of tree-structured DEM encoder for a 13-level DAC.

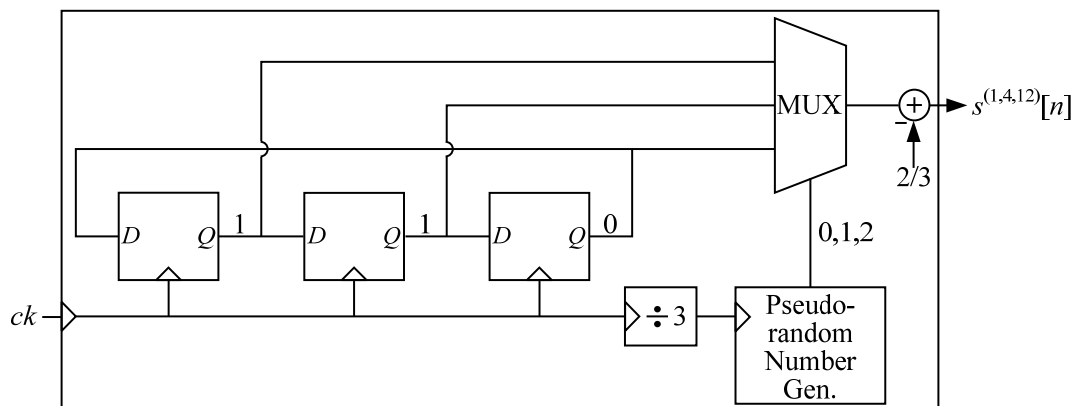


Figure 6 Register transfer level circuitry that generates $s^{(1,4,12)}$ for $c[n] \in \{2, 5, 8, 11\}$.

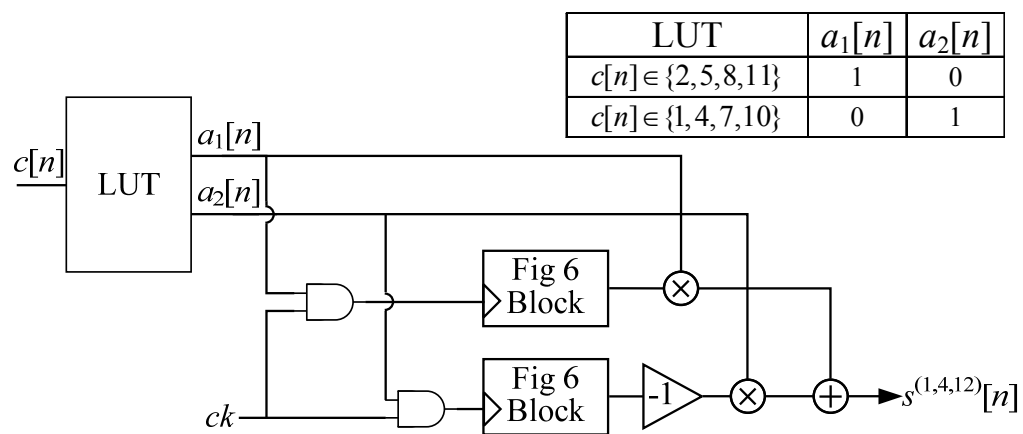


Figure 7 Register transfer level circuitry that generates $s^{(u,v,w)}$ for $(u,v,w) = (1,4,12)$.

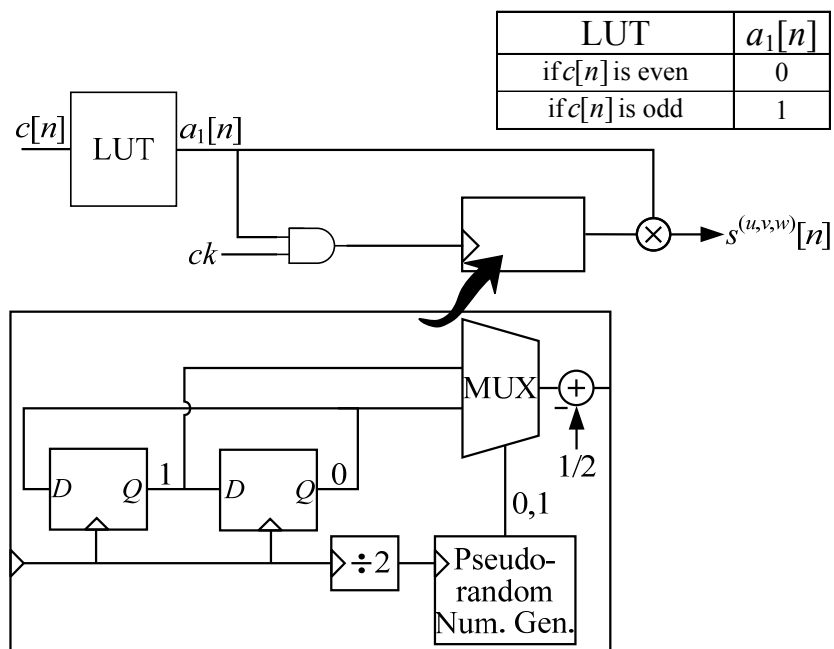


Figure 8 Register transfer level circuitry that generates $s^{(u,v,w)}$ for $(u,v,w) \neq (1,4,12)$.

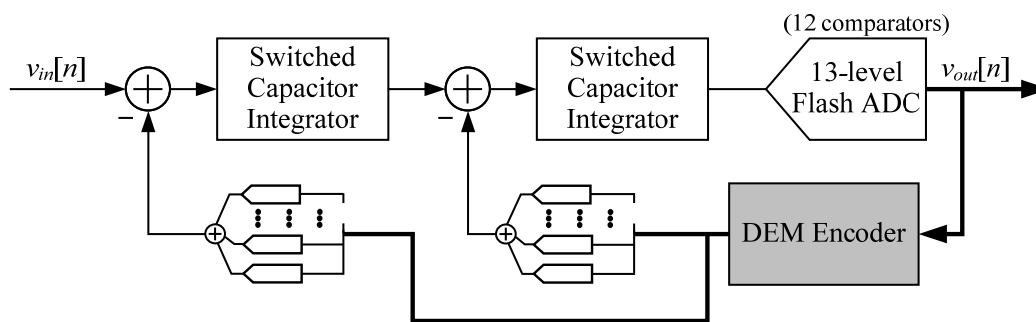


Figure 9 Block diagram of the second order $\Delta\Sigma$ ADC.

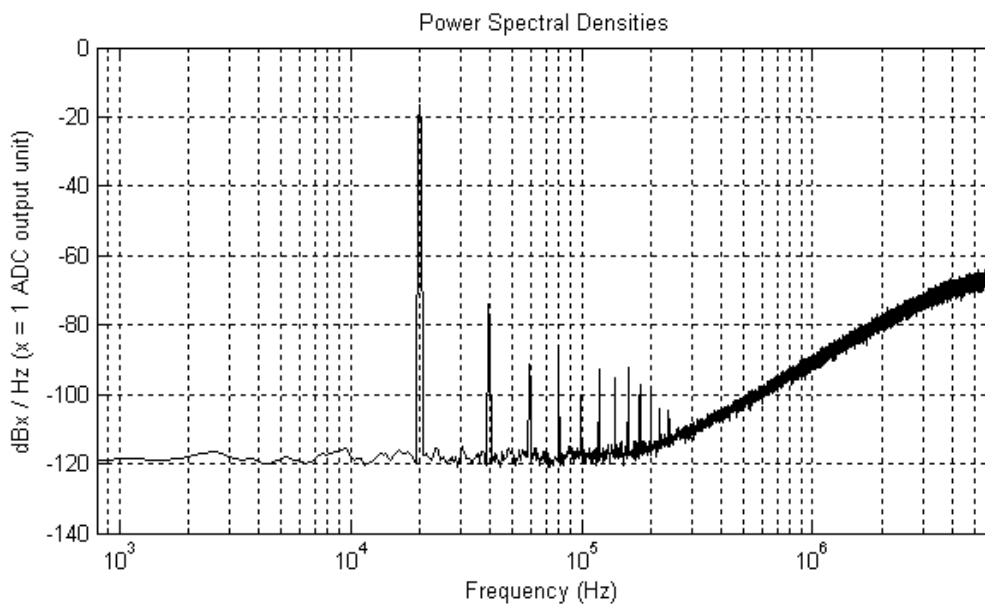


Figure 10a Power spectral density at the output of the second order $\Delta\Sigma$ ADC in Figure 9 without the DEM Encoder.

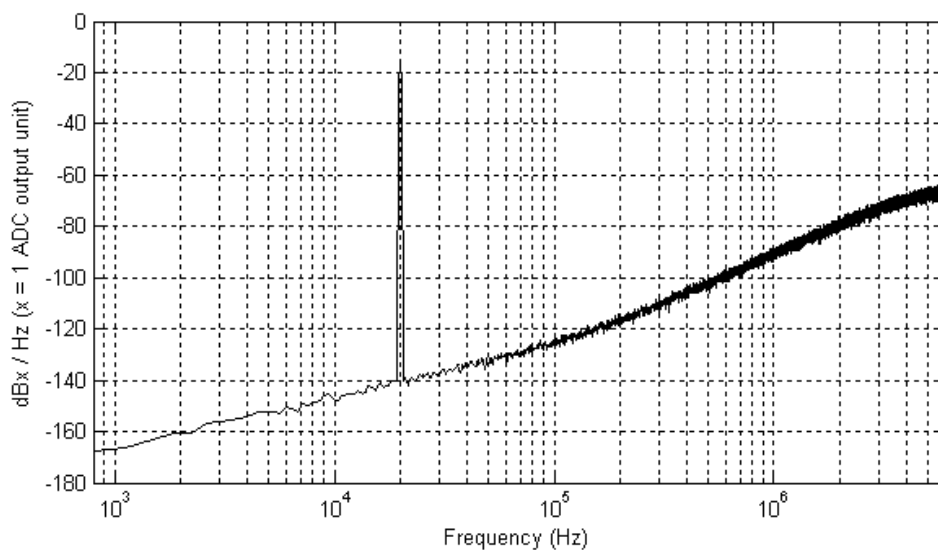


Figure 10b Power spectral density at the output of the second order $\Delta\Sigma$ ADC in Figure 9 with the tree-structured DEM Encoder shown in Figure 5.

IX. REFERENCES

1. L. R. Carley, J. Kenny, "A 16-bit 4'th order noise-Shaping D/A converter," *1988 IEEE Custom Integrated Circuits Conference*, May, 1988.
2. L. R. Carley, "A noise-shaping coder topology for 15+ bit converters," *IEEE Journal of Solid State Circuits*, vol. 24, no. 2, pp. 267-273, April, 1989.
3. B. H. Leung, S. Sutarja, "Multi-bit Σ - Δ A/D converter incorporating a novel class of dynamic element shaping," *IEEE Transactions on Circuits and Systems II, Analog and Digital Signal Processing*, vol. 39, pp. 35-51, January, 1992.
4. F. Chen, B. H. Leung, "A high resolution multibit sigma-delta modulator with individual level averaging," *IEEE Journal of Solid State Circuits*, vol. 30, pp. 453-460, April, 1995.
5. M. J. Story, "Digital to analogue converter adapted to select input sources based on a preselected algorithm once per cycle of a sampling signal," U.S. Patent 5 138 317, August 11, 1992.
6. W. Redman-White, D. J. L. Bourner, "Improved dynamic linearity in multi-level $\Sigma\Delta$ converters by spectral dispersion of D/A distortion products," *IEEE European Conference Circuit Theory and Design*, September, 1989.
7. H. S. Jackson, "Circuit and method of cancelling nonlinearity error associated with component mismatches in a data converter," U.S. Patent 5 221 926, June 22, 1993.
8. R. T. Baird, T. S. Fiez, "Improved $\Delta\Sigma$ DAC linearity using data weighted averaging," *IEEE International Symposium of Circuits and Systems*, May, 1995.
9. R. T. Baird, T. S. Fiez, "Linearity enhancement of $\Delta\Sigma$ A/D and D/A converters using data weighted averaging," *IEEE Transactions of Circuits and Systems II, Analog and Digital Signal Processing*, vol. 42, pp. 753-762, December, 1995.
10. R. W. Adams, T. W. Kwan, "Data-directed scrambler for multi-bit noise-shaping D/A Converters," U.S. Patent 5 404 142, April, 1995.
11. R. Schreier, B. Zhang, "Noise-shaped multibit D/A converter employing unit elements," *Electronics Letters*, vol. 31, pp. 1712-1713, September, 1995.

12. I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Transactions of Circuits and Systems II, Analog and Digital Signal Processing*, vol. 44, pp. 808-817, October, 1997.
13. E. Fogleman, I. Galton, W. Huff, H. Jensen "A 3.3-V single-poly CMOS audio ADC delta-sigma modulator with 98-dB peak SINAD and 105-dB peak SFDR," *IEEE Journal of Solid State Circuits*, vol. 35, no. 3, pp. 297-307, March, 2000.
14. E. Fogleman, J. Welz, I. Galton, "An audio ADC Delta-Sigma modulator with 100-dB peak SINAD and 102-dB DR using a second-order mismatch-shaping DAC," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 339-348, March 2001.
15. I. Galton, "Delta-sigma data conversion in wireless transceivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 1, pp. 302-316, January 2002.
16. J. Grilo, I. Galton, K. Wang, R. G. Montemayor, "A 12-mW ADC delta-sigma modulator with 80 dB of dynamic range integrated in a single-chip bluetooth transceiver," *IEEE Journal of Solid State Circuits*, vol. 37, no. 3, pp. 271-278, March, 2002.
17. T. Shui, R. Schreier, F. Hudson, "Mismatch shaping for a current-mode multibit delta-sigma DAC," *IEEE Journal of Solid State Circuits*, vol. 34, no. 3, pp. 331-338, March, 1999.
18. I. Fujimori, A. Nogi, T. Sugimoto, "A multibit delta-sigma audio DAC with 120-dB dynamic range," *IEEE Journal of Solid State Circuits*, vol. 35, no. 8, pp. 1066-1073, August, 2000.
19. R. E. Radke, A. Eshraghi, T. F. Fiez, "A 14-bit current-mode $\Sigma\Delta$ DAC based upon rotated data weighted averaging," *IEEE Journal of Solid State Circuits*, vol. 35, no. 8, pp. 1074-1084, August, 2000.
20. E. Tuijl, J. Homberg, D. Reefman, C. Bastiaansen, L. Dussen, "A 128fs, multi-bit $\Sigma\Delta$ CMOS audio DAC with real-time DEM and 115dB SFDR," *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, February, 2004.
21. M. Clara, W. Klatzer, A. Wiesbauser, D. Straeusnigg, "A 350MHz low-OSR

- $\Sigma\Delta$ current-steering DAC with active termination in 0.13 μm CMOS,” *IEEE International Solid-State Circuits Conference*, Digest of Technical Papers, February, 2005.
22. Z. Zhang, G. Temes, “A segmented data-weighted-averaging technique,” *IEEE International Symposium of Circuits and Systems*, May, 2007.
 23. T. S. Kaplan, J. F. Jensen, C. H. Fields, M. F. Chang, “A 2-GS/s 3-bit $\Sigma\Delta$ -modulated DAC with tunable bandpass mismatch shaping,” *IEEE Journal of Solid State Circuits*, vol. 40, no. 3, pp. 603-610, March, 2005.
 24. S. Reekmans, J. D. Maeyer, P. Rombouts, L. Weyten, “Quadrature mismatch shaping for digital-to-analog Converters,” *IEEE Transactions of Circuits and Systems I, Regular Papers*, vol. 53, no. 12, pp. 2529-2538, December, 2006.
 25. H. Hsieh, L. Lin, “A first-order tree-structured DAC with reduced signal-band noise,” *IEEE Transactions of Circuits and Systems II: Express Briefs*, vol. 54, no. 5, pp. 392-396, May, 2007.
 26. M. Vadipour, “Techniques for preventing tonal behavior of data weighted averaging algorithm in $\Sigma\Delta$ modulators,” *IEEE Transactions of Circuits and Systems II, Analog and Digital Signal Processing*, vol. 47, pp. 1137-1144, November, 2000.
 27. J. Arias, P. Kiss, V. Boccuzzi, L. Quintanilla, L. Enriquez, J. Vicente, D. Bisbal, J. S. Pablo, J. Barbolla, “Nonlinearity correction for multibit $\Delta\Sigma$ DACs,” *IEEE Transactions of Circuits and Systems I, Regular Papers*, vol. 52, no. 6, pp. 1033-1041, June, 2005.
 28. A. A. Hamoui, K. W. Martin, “High-order multibit modulators and pseudo data-weighted-averaging in low-oversampling $\Delta\Sigma$ ADCs for broad-band applications,” *IEEE Transactions of Circuits and Systems I, Regular Papers*, vol. 51, no. 1, pp. 72-85, January, 2004.
 29. D. H. Lee, T. H. Kuo, “Advancing data weighted averaging technique for multi-bit sigma-delta,” *IEEE Transactions of Circuits and Systems II: Express Briefs*, vol. 54, no. 10, pp. 838-842, October, 2007.
 30. R. Schreier, G. C. Temes, *Understanding Delta-Sigma Data Converters*, John Wiley and Sons, Inc, 2005.

31. R. Adams, K. Nguyen, K. Sweetland, "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 1871-1878, December, 1998
32. K. Vleugels, S. Rabii, B. A. Wooley, "A 2.5V sigma-delta modulator for broadband communications applications," *IEEE Journal of Solid-State Circuits*, vol. 36, pp.1887-1899, December, 2001.
33. Chan K.L., Zhu J., Galton I., "Dynamic Element Matching to Prevent Nonlinear Distortion From Pulse-Shape Mismatches in High-Resolution DACs," *IEEE Journal of Solid-State Circuits*, vol. 43, pp.2067-2078, September, 2008.
34. Chan K.L., Rakuljic N., Galton I., "Segmented Dynamic Element Matching for High-Resolution Digital-to-Analog Conversion," *IEEE Transactions On Circuits and Systems I*, vol. 55, pp.3383-3392, December, 2008
35. A. V. Oppenheim, R. W. Schaffer, J. R. Buck, "Discrete-Time Signal Processing," Prentice-Hall, Inc, Second Ed., 1999.
36. Welz J., Galton I., "Simplified Logic For First-Order and Second-Order Mismatch-Shaping Digital-to-Analog Converters," *IEEE Transactions on Circuits and Systems II*, vol.48, no. 11, pp.1014-1027, November, 2001
37. E. Siragusa, I. Galton, "A Digitally Enhanced 1.8V 15b 40MS/s CMOS Pipelined ADC," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2126-2138, December 2004.
38. I. Galton, "Granular quantization noise in a class of delta-sigma modulators," *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 848-859, May, 1994

Chapter 2

Suppression of Quantization-Induced Convergence Error in Pipelined ADCs With Harmonic Distortion Correction

Abstract—Harmonic Distortion Correction (HDC) is one of two published digital background calibration techniques that compensate for residue amplifier nonlinearity in pipelined ADCs. The techniques make it possible to reduce the gains and bandwidths, and therefore the power dissipations, of the op-amps that make up the residue amplifiers without sacrificing pipelined ADC accuracy. Unfortunately, the previously published techniques fail to operate properly when they measure residue amplifier distortion for certain pipelined ADC input signals, most notably input signals with small peak-to-peak variations about certain constant values. This paper identifies the cause of the problem, quantifies its effects, and provides an all-digital solution applicable to the HDC technique.

I. INTRODUCTION

Pipelined analog-to-digital converters (ADCs) are widely used in applications that require greater accuracy than can be achieved practically by flash ADCs, and greater signal bandwidth than can be achieved practically by oversampling or succes-

Copyright © 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

N. Rakuljic and I. Galton are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: galton@ece.ucsd.edu).

sive approximation ADCs. With present IC technology, they are most commonly used in applications that require greater than 50 dB of signal to noise and distortion ratio (SNDR) and greater than 50 MHz of signal bandwidth.

The residue amplifiers in the first few stages of a pipelined ADC must have high linearity for the ADC to achieve a high SNDR. In a conventional pipelined ADC, this necessitates op-amps with high open-loop gains, high bandwidths, and relatively low output swings. Consequently, the op-amps tend to dominate the overall power dissipation in conventional pipelined ADCs.

Recently, digital background calibration techniques have been proposed that make it possible to reduce the performance and, hence, the power dissipation of the op-amps without sacrificing pipelined ADC accuracy [39 – 43]. The techniques use digital correlation algorithms to measure the residue amplifier distortion coefficients during normal ADC operation, and they use the measured coefficient values to digitally cancel much of the residue amplifier distortion. This allows higher-distortion op-amps to be tolerated without significantly degrading the overall pipelined ADC accuracy. The calibration circuitry is mostly digital and tends to dissipate relatively little power, so the net reduction in pipelined ADC power dissipation offered by the techniques can be significant.

Unfortunately, all of the previously published digital background calibration techniques fail to measure the residue amplifier distortion coefficients properly for certain pipelined ADC input signals. As explained in [42] and [43], the most robust of the techniques in this respect is the Harmonic Distortion Correction (HDC) technique.

Nevertheless, for certain input signals, most notably those with small peak-to-peak variations about certain constant values, it too fails to accurately measure the residue amplifier distortion coefficients. Although it operates properly for the majority of pipelined ADC input signals, its failure to work properly even for a small class of input signals presents a problem in practice.

This paper identifies and quantifies the failure mechanism, and proposes a simple all-digital modification of the HDC technique that solves the problem. As explained in the paper, the problem arises because the small amount of quantization error introduced by the pipelined ADC corrupts the coefficient measurement process under certain conditions. The problem is subtle because the corruption occurs even when the variance of the quantization noise is much smaller than the dominant error sources in the pipelined ADC. Although the paper describes the problem in the context of the HDC technique, the problem also affects the other previously published digital calibration techniques, because the quantization error is always present during the coefficient measurement process regardless of the technique used.

The paper consists of three main sections. Section II reviews the HDC technique in the context of an example pipelined ADC architecture. Section III identifies and quantifies the HDC failure mechanism, and Section IV presents the proposed solution.

II. BACKGROUND INFORMATION

A. Pipelined ADC and HDC Overview

Figure 11 shows a six-stage, 14-b pipelined ADC example. The input to the pipelined ADC is a sequence of sampled analog voltages, $v_{in,1}(nT_s)$, where T_s is the sample interval. In practice each stage in a pipelined ADC contains delay elements, but the delay elements have been omitted in the pipelined ADC example of Figure 11. This reduces the complexity of the notation presented in the paper without significantly affecting the results of the paper.

All but the last pipelined ADC stage in Figure 11 have the form shown in Figure 12. Each consists of an 8-level flash ADC with a nominal quantization step-size of Δ , an 8-level dynamic element matching (DEM) DAC, and a *residue amplifier*. The last stage of the pipelined ADC consists only of a 16-level flash ADC with a nominal quantization step-size of $\Delta/2$.

The output of the k th stage's flash ADC (flash ADC $_k$) is

$$x_k[n] = v_{in,k}(nT_s) + e_{ADC,k}[n] \quad (51)$$

where $v_{in,k}(nT_s)$ is the input sequence to flash ADC $_k$, and $e_{ADC,k}[n]$ is error introduced by flash ADC $_k$. This error is the output minus the input of flash ADC $_k$, with the least significant bit of the output taken to have a weight equal to the nominal quantization step-size of flash ADC $_k$. In the absence of non-ideal circuit behavior, $e_{ADC,k}[n]$ is just quantization error, and is bounded in magnitude by half of its nominal quantization step size.

The 8-level DAC in the k th stage (DAC_k) for $k = 1, 2, \dots, 5$ converts $x_k[n]$ into analog format. The difference between the stage's input sequence and the DAC's output sequence, $v_k(nT_s)$, is called the stage's *residue*. It follows from (51) that in the absence of non-ideal circuit behavior the stage's residue is given by

$$v_k(nT_s) = -e_{\text{ADC},k}[n], \quad (52)$$

and is bounded in magnitude by $\Delta/2$. Ideally, the k th stage's residue amplifier (RA_k) scales the residue linearly by a factor of 4, i.e.,

$$v_{\text{out},k}(nT_s) = 4v_k(nT_s). \quad (53)$$

Therefore, the analog output of the k th pipeline stage is ideally bounded in magnitude by 2Δ , which is less than half the input range of the flash ADC in the subsequent pipeline stage. The extra input range, called *over-range margin*, is used to accommodate flash ADC errors that arise from non-ideal circuit behavior such as comparator offset voltages and resistor ladder component mismatches. These errors subsequently cancel in the digital path of the pipelined ADC assuming ideal circuit behavior except for flash ADC errors [44, 45, 46].

It follows from (51) and Figure 12 that the digital output of the k th stage, for $k = 1, 2, \dots, 5$, is given by

$$x_{\text{out},k}[n] = v_{\text{in},k}(nT_s) + e_{\text{ADC},k}[n] + r_k[n] \quad (54)$$

where

$$r_k[n] = \frac{1}{4}(x_{k+1}[n] + r_{k+1}[n]). \quad (55)$$

is called the *digitized residue* of the k th stage. Recursively applying (51)–(55) with

$$v_{in,k+1}(nT_s) = v_{out,k}(nT_s) \quad (56)$$

indicates that the output of the pipelined ADC ideally is

$$x_{out,1}[n] = v_{in,1}(nT_s) + \frac{1}{4^5} e_{ADC,6}[n]. \quad (57)$$

Therefore, in the absence of non-ideal circuit behavior the quantization error from all the flash ADCs except that in the last pipeline stage cancel, so the only quantization error that propagates to the pipelined ADC output is a scaled version of the last stage's quantization error. The pipelined ADC input range is bounded in magnitude by 4Δ and the scaled version of quantization error is bounded in magnitude by $(\Delta/4)/4^5$, so the pipelined ADC ideally performs 14-bit quantization.

Equations (52)–(57) describe the ideal pipelined ADC behavior. In practice, the output deviates from (57) because of non-ideal circuit behavior. In particular, practical residue amplifiers introduce gain error and nonlinear distortion. An often-realistic model of the k th residue amplifier that includes this non-ideal behavior is

$$v_{out,k}(nT_s) = 4 \left[(1 + \alpha_{1,k}) v_k(nT_s) + \alpha_{3,k} v_k^3(nT_s) \right] \quad (58)$$

where $\alpha_{1,k}$ is a gain error coefficient, and $\alpha_{3,k}$ is a third-order nonlinear distortion coefficient [43]. If the pipelined ADC in Figure 11 is ideal except with residue amplifiers that are well-modeled by (58), it follows from (51), (52), (54)–(57), and (58) that

$$x_{out,1}[n] = x_{out,1}[n]_{\text{ideal}} + \underbrace{\sum_{k=1}^5 4^{1-k} \left[\alpha_{1,k} v_k(nT_s) + \alpha_{3,k} v_k^3(nT_s) \right]}_{\text{Distortion Terms}} \quad (59)$$

where $x_{out,1}[n]_{\text{ideal}}$ is $x_{out,1}[n]$ as given by (57). The distortion terms in (59) are undesirable because typically they decrease both the SNDR and the spurious-free dynamic

range (SFDR) of the pipelined ADC.

HDC can be applied to each stage of a pipelined ADC to digitally estimate and cancel the distortion terms [42, 43]. As indicated by (59) the distortion terms contributed by the residue amplifier in the k th pipeline stage are scaled by 4^{1-k} , so the residue amplifiers in the first few pipeline stages usually are the dominant sources of residue amplifier distortion in the pipelined ADC. Thus, in practice HDC usually is applied only to the first few pipeline stages.

Figure 13 shows an example of the 14-b pipelined ADC with HDC applied to the first four stages and Figure 14 shows the k th of these stages in more detail. The implementation of HDC in the k th stage consists of the addition of a calibration sequence, $c_k[n]$, to the output of flash ADC $_k$, an increase in the resolution of DAC $_k$ to accommodate the added sequence, and the addition of a digital logic block, labeled HDC $_k$ in the figure. The k th stage's calibration sequence has the form

$$c_k[n] = \sum_{i=1}^N t_{i,k}[n] \quad (60)$$

where the $t_{i,k}[n]$ sequences are independent, 2-level, zero-mean, pseudorandom sequences that take on values of $\pm A$. In this paper $A = \Delta/16$ as in [43] and, for a reason explained shortly, $N = 4$, so DAC $_k$ must have at least 61 output levels with a minimum step-size of $\Delta/8$ to accommodate $c_k[n]$.

The calibration sequence increases the maximum signal swing at the output of the residue amplifier, so it effectively decreases the over-range margin of the stage. Therefore, a design consideration is that A and N must be small enough that the re-

maining over-range margin is sufficient to accommodate the largest expected stage offset and flash ADC_k errors. In the example design, half the over-range margin is used to accommodate the calibration sequence.

Ideally, each HDC_k block cancels error arising from $\alpha_{1,k}$ and $\alpha_{3,k}$ in the k th stage's residue amplifier. It follows from (52), (54), (57), and (59) that

$$r_k[n] = (1 + \alpha_{1,k})v_k(nT_s) + \alpha_{3,k}v_k^3(nT_s) + \frac{1}{4^5}e_{\text{ADC},6}[n] + g_{k+1}(v_k(nT_s)) \quad (61)$$

for $k = 1, 2, 3$, and 4, where the last term represents error caused by any non-ideal behavior of the stages subsequent to Stage k such as residue amplifier distortion in Stage 5. Therefore, the objectives of HDC applied to the k th stage can be viewed as estimating and canceling the terms proportional to $\alpha_{1,k}$ and $\alpha_{3,k}$ in (61).

It follows from (51) and Figure 14 that

$$v_k(nT_s) = -e_{\text{ADC},k}[n] - c_k[n]. \quad (62)$$

Figure 15 with

$$c_{1,k}[n] = -\frac{t_{1,k}[n]}{A} \quad \text{and} \quad c_{3,k}[n] = -\frac{t_{1,k}[n]t_{2,k}[n]t_{3,k}[n]}{A^3} \quad (63)$$

shows the details of the HDC_k block. To estimate the $\alpha_{1,k}$ and $\alpha_{3,k}$ coefficients, the HDC_k block first computes the time averages

$$\tilde{\gamma}_{1,k} = \frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_k[n] + c_k[n])c_{1,k}[n], \quad \tilde{\gamma}_{3,k} = \frac{1}{6A^3} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_k[n] + c_k[n])c_{3,k}[n] \quad (64)$$

and

$$\tilde{\eta}_{2,k} = \frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_k[n] + c_k[n])^2 c_k[n]c_{1,k}[n], \quad (65)$$

where m is the starting time index of the time averaging operations, and P is the number of averaged samples (e.g., $P = 2^{32}$ in [43])². To the extent that the correlations of $c_{1,k}[n]$ and $c_{3,k}[n]$ with the last term in (61) can be neglected, it follows from (60)-(64), the statistical properties of the $t_{i,k}[n]$ sequences, and the Law of Large Numbers, that for $N = 4$

$$\tilde{\gamma}_{1,k} \cong \tilde{\alpha}_{1,k} + \left(10A^2 - \frac{3\tilde{\eta}_{2,k}}{(1 + \tilde{\alpha}_{1,k})^2} \right) \tilde{\alpha}_{3,k}, \quad (66)$$

and

$$\tilde{\gamma}_{3,k} = \tilde{\alpha}_{3,k}, \quad (67)$$

where

$$\tilde{\alpha}_{1,k} \cong \alpha_{1,k} \quad \text{and} \quad \tilde{\alpha}_{3,k} \cong \alpha_{3,k} \quad (68)$$

to a high degree of accuracy provided P is large.

Combining (66) and (67) results in

$$\tilde{\gamma}_{1,k} \cong \tilde{\alpha}_{1,k} + \left(10A^2 - \frac{3\tilde{\eta}_{2,k}}{(1 + \tilde{\alpha}_{1,k})^2} \right) \tilde{\gamma}_{3,k} \quad (69)$$

which is a cubic equation that can be solved to find $\tilde{\alpha}_{1,k}$ in terms of $\tilde{\gamma}_{1,k}$, $\tilde{\gamma}_{3,k}$ and $\tilde{\eta}_{2,k}$. A closed form solution exists, but it is complicated. An approximate but simpler solution can be obtained by viewing $\tilde{\alpha}_{1,k}$ as a function of $\tilde{\gamma}_{1,k}$, $\tilde{\gamma}_{3,k}$ and $\tilde{\eta}_{2,k}$ and using

²Note that $\tilde{\eta}_{2,k}$ in (65) is different than the corresponding quantity in [42] and [43]. It can be verified that this version of $\tilde{\eta}_{2,k}$ avoids an approximation made in [42] and [43] and therefore yields slightly more accurate results than those obtained in [42] and [43].

a Taylor series expansion around $\tilde{\gamma}_{3,k}$, i.e.,

$$\tilde{\alpha}_{1,k}(\tilde{\gamma}_{1,k}, \tilde{\gamma}_{3,k}, \tilde{\eta}_{2,k}) \cong \tilde{\alpha}_{1,k}(\tilde{\gamma}_{1,k}, 0, \tilde{\eta}_{2,k}) + \sum_{i=1}^3 \frac{1}{k!} \frac{\partial^i \tilde{\alpha}_{1,k}}{\partial \tilde{\gamma}_{3,k}^i}(\tilde{\gamma}_{1,k}, 0, \tilde{\eta}_{2,k}) \tilde{\gamma}_{3,k}^i \quad (70)$$

It follows from (69) that

$$\tilde{\alpha}_{1,k}(\tilde{\gamma}_{1,k}, 0, \tilde{\eta}_{2,k}) = \tilde{\gamma}_{1,k}. \quad (71)$$

Differentiating (69) with respect to $\tilde{\gamma}_{3,k}$ and substituting (71) into the result yields

$$\frac{\partial \tilde{\alpha}_{1,k}}{\partial \tilde{\gamma}_{3,k}}(\tilde{\gamma}_{1,k}, 0, \tilde{\eta}_{2,k}) = \frac{3\tilde{\eta}_{2,k}}{(1 + \tilde{\gamma}_{1,k})^2} - 10A^2, \quad (72)$$

and continuing this process recursively yields the remaining two terms on the right side of (70). Substituting these results into (70) yields

$$\tilde{\alpha}_{1,k} \cong \tilde{\gamma}_{1,k} + \left(\frac{3\tilde{\eta}_{2,k}}{(1 + \tilde{\gamma}_{1,k})^2} - 10A^2 \right) \left(\tilde{\gamma}_{3,k} - \frac{6\tilde{\eta}_{2,k}}{(1 + \tilde{\gamma}_{1,k})^3} \tilde{\gamma}_{3,k}^2 + \frac{63\tilde{\eta}_{2,k}^2 - 90A^2\tilde{\eta}_{2,k}(1 + \tilde{\gamma}_{1,k})^2}{(1 + \tilde{\gamma}_{1,k})^6} \tilde{\gamma}_{3,k}^3 \right). \quad (73)$$

The HDC_k block uses the estimates of $\alpha_{1,k}$ and $\alpha_{3,k}$ given by (73) and (67), respectively, to calculate the *corrected digitized residue*:

$$r_k[n] \Big|_{\text{corrected}} = \frac{1}{1 + \tilde{\alpha}_{1,k}} r_k[n] - \frac{\tilde{\alpha}_{3,k}}{(1 + \tilde{\alpha}_{1,k})^4} r_k^3[n]. \quad (74)$$

It can be verified that this causes $x_{out,k}[n] \cong x_{out,k}[n] \Big|_{\text{ideal}}$ [42].

The accuracy with which each HDC block estimates its nonlinearity coefficients depends, in part, on how well the subsequent HDC blocks have corrected the nonlinearity introduced by the residue amplifiers in their respective stages. Therefore,

the HDC_k blocks for $k = 1, 2, 3$, and 4 perform their measurements of $\tilde{\alpha}_{1,k}$ and $\tilde{\alpha}_{3,k}$ sequentially and periodically, first for $k = 4$, then for $k = 3$, then for $k = 2$, and then for $k = 1$, after which the process repeats [43]. Each HDC block continually implements (74) with the most recent $\tilde{\alpha}_{1,k}$ and $\tilde{\alpha}_{3,k}$ values it measured.

III. EFFECT OF QUANTIZATION ERROR ON HDC COEFFICIENT ESTIMATION

The goal of HDC in each stage is to perfectly cancel the distortion terms introduced by that stage's residue amplifier. Unfortunately, the cancellation is never perfect in practice because the operations that the HDC_k blocks perform to estimate $\alpha_{1,k}$ and $\alpha_{3,k}$, and to cancel the nonlinear distortion terms involve approximations.

For example, suppose that the pipelined ADC of Figure 13 is ideal except that the flash ADCs have threshold errors and the $\alpha_{1,1}$, $\alpha_{3,1}$, and $\alpha_{1,2}$ coefficients are non-zero. If the HDC blocks correctly measure $\tilde{\alpha}_{3,k} = 0$ for $k = 2, 3$, and 4, and $\tilde{\alpha}_{1,k} = 0$ for $k = 3$, and 4, then the only significant contribution to the last term in (61) with $k = 1$ occurs because of the HDC_2 block's imperfect estimation of $\alpha_{1,2}$. In this case it follows from (51), (55), (56), (58), (74) and Figure 14, that (61) becomes

$$r_1[n] = (1 + \alpha_{1,1})v_1(nT_s) + \alpha_{3,1}v_1^3(nT_s) + \frac{1}{4^5}e_{ADC,6}[n] + e_{HDC,2}[n] \quad (75)$$

where

$$e_{HDC,2}[n] = -\frac{\tilde{\alpha}_{1,2}}{4^5(1 + \tilde{\alpha}_{1,2})}e_{ADC,6}[n] + \frac{\lambda_2}{4}(e_{ADC,2}[n] + c_2[n]), \quad (76)$$

and

$$\lambda_2 = \frac{(\tilde{\alpha}_{1,2} - \alpha_{1,2})}{(1 + \tilde{\alpha}_{1,2})}. \quad (77)$$

The $e_{HDC,2}[n]$ term represents the error in $r_1[n]$ caused by the HDC_2 block's imperfect estimation of $\alpha_{1,2}$. It adds directly to the pipelined ADC output, but even if its mean squared value is below the noise floor of the pipelined ADC, it still can sometimes degrade the performance of the pipelined ADC by corrupting the HDC_1 block's estimates of $\alpha_{1,1}$ and $\alpha_{3,1}$. This happens because for some pipelined ADC input signals $e_{ADC,2}[n]$ is very strongly correlated with both $c_{1,1}[n]$ and $c_{3,1}[n]$.

For example, consider a special case of the above example wherein $\alpha_{1,1} = 0$, $\alpha_{3,1} = 0$, flash ADC₂ has just a single threshold error, $\lambda_2 = 0.001$ (which is consistent with the measured results presented in [43]), and the pipelined ADC's input sequence is $V_{in}(nT_s) = \Delta$ for all n . Given that $\alpha_{1,1} = 0$ and $\alpha_{3,1} = 0$, the ideal operation of the HDC_1 block would be to calculate $\tilde{\alpha}_{1,1} = 0$ and $\tilde{\alpha}_{3,1} = 0$ in which case it would have no effect on $r_1[n]$ and the only effect of the $e_{HDC,2}[n]$ term would be to decrease the pipelined ADC's SNDR by about 1.5 dB relative to its ideal (quantization noise only) value.

Unfortunately, the $e_{HDC,2}[n]$ term causes the HDC_1 block not to operate ideally for this example. The input to flash ADC₂ is $2\Delta - 4c_1[n]$, so $e_{ADC,2}[n]$ for each n takes on one of the five points shown in Figure 16(a) (one of which is affected by the threshold error). The resulting correlations of $e_{ADC,2}[n]$ with $c_{1,1}[n]$ and $c_{3,1}[n]$ are both $-\Delta/16$ which causes the HDC_1 block to incorrectly calculate $\tilde{\alpha}_{1,1} \cong 0.0082$ and

$\tilde{\alpha}_{3,1} \cong -0.17$ (via (64) and (73)). By implementing (74) with these incorrect estimates of $\alpha_{1,1}$ and $\alpha_{3,1}$, the HDC_1 block introduces significant nonlinear distortion in this case such that the pipelined ADC's SNDR is reduced by about 23 dB relative to its ideal value.

This problem is highly dependent upon the pipelined ADC's input sequence. For example, suppose that the example above is changed only in that $V_{in}(nT_s) = 0.5\Delta$ for all n . Then, the input to flash ADC₂ is $-4c_1[n]$, and the possible values of $e_{ADC,2}[n]$ are the five points shown in Figure 16(b). The resulting correlations of $e_{ADC,2}[n]$ with $c_{1,1}[n]$ and $c_{3,1}[n]$ are both zero, so the HDC_1 block correctly measures $\tilde{\alpha}_{1,1} = 0$ and $\tilde{\alpha}_{3,1} = 0$ in this case. Thus, for this input sequence, the leakage of $e_{ADC,2}[n]$ into $r_1[n]$ does not lead to incorrect estimates of $\alpha_{1,1}$ and $\alpha_{3,1}$, so the pipelined ADC's SNDR is only degraded by approximately 1.5 dB from the presence of $e_{HDC,2}[n]$ in the output sequence.

Returning to the more general situation in which $\alpha_{1,1}$, $\alpha_{3,1}$, and $\alpha_{1,2}$ are non-zero, it is straightforward to verify that (66), (75), and (76) imply that the errors in the HDC_1 block's estimates of $\gamma_{1,1}$ and $\gamma_{3,1}$ caused by correlations of $e_{ADC,2}[n]$ with $c_{1,1}[n]$ and $c_{3,1}[n]$ have magnitudes that are bounded by

$$\frac{\lambda_2 e_{ADC,max}}{4A}, \quad \text{and} \quad \frac{\lambda_2 e_{ADC,max}}{24A^3}, \quad (78)$$

respectively, where $e_{ADC,max}$ is the largest possible magnitude of $e_{ADC,k}[n]$ for all n and k . While the bounds given by (78) are not tight, specific pipelined ADC input values are known to the authors for which the errors in the HDC_1 block's estimates of $\gamma_{1,1}$

and $\gamma_{3,1}$ have magnitudes that are larger than half those in (78).

Therefore, in the worst-case scenarios the HDC_1 block's estimates of $\alpha_{1,1}$ and $\alpha_{3,1}$ are corrupted by error terms that depend on A^{-1} and A^{-3} . If these error terms have magnitudes that are significant relative to the magnitudes of $\alpha_{1,1}$ and $\alpha_{3,1}$, then the HDC_1 block at best will not cancel distortion from the first stage's residue amplifier accurately, and at worst can actually introduce extra distortion (as in the example described above). The error terms can be reduced by increasing A , but, as described in the previous section, increasing A uses up more of the over-range margin of the subsequent stage. This places a practical upper bound on A , so it is not always possible to make A large enough that the errors caused by correlations of $e_{\text{ADC},2}[n]$ with $c_{1,1}[n]$ and $c_{3,1}[n]$ are negligible for all pipelined ADC input signals.

Similar results hold for the other HDC blocks. In general, for the worst-case input signals the estimation process in each HDC block is highly sensitive to quantization error terms from subsequent stages that leak into its stage's digitized residue. For some ADC input sequences the error terms corrupt the HDC block's $\alpha_{i,k}$ coefficient estimates even when their average power is negligible compared to those of the other error sources in the pipelined ADC. In such cases the error terms do not significantly reduce the pipelined ADC SNDR directly, but rather they cause HDC blocks to introduce error that reduces the SNDR as result of the inaccurate estimates of the $\alpha_{i,k}$ coefficients.

A pipelined ADC converts each sample of its input sequence to a digital number independently of all prior input samples, and each sample of the correlation se-

quence is statistically independent of all prior correlation sequence samples by design, so the statistical expectation of the HDC estimation error caused by quantization error leakage at time n has no dependence on prior pipelined ADC input samples. Furthermore, a pipelined ADC implements a time-invariant discrete-time system and the calibration sequence is a stationary random process by design, so any dependence of the expectation of the estimation error on each pipelined ADC input sequence value must be independent of the sample time n . It follows that there is at least one input value that maximizes this estimation error expectation at any time n , so keeping the input signal constant at this worst-case value for all sample times maximizes the effect of the problem. This is why the set of worst-case pipelined ADC input sequences includes one or more constant sequences.

It follows that the full extent of the problem can be evaluated by considering the HDC coefficient estimation process for all constant input sequences. Furthermore, if the HDC technique is modified such that the HDC coefficients are estimated accurately for every constant pipelined ADC input sequence, it follows that the modification will also cause the coefficients to be estimated accurately for every non-constant pipelined ADC input sequence. Consequently, the simulation results presented in the remainder of this paper only consider cases in which the HDC blocks estimate their coefficients for constant pipelined ADC input sequences.

As explained in [43], the problem can be mitigated by using $N = 5$ $t_{i,k}[n]$ sequences in (60) instead of $N = 3$ as originally proposed in [42]. The two extra $t_{i,k}[n]$ sequences act as dither which tends to reduce the correlations of $e_{ADC,2}[n]$ with $c_{1,1}[n]$

and $c_{3,1}[n]$. Unfortunately, even with $N = 5$ the problem still occurs for pipelined ADC input signals that have small peak-to-peak variations about certain constant values.

Figure 17 shows simulation results that illustrate the problem for the example pipelined ADC shown in Figure 13 using calibration sequences given by (60) with $N = 5$. The simulated pipelined ADC includes DEM DACs and the DAC noise cancellation (DNC) technique as described in [43] with capacitor mismatches chosen such that the pipelined ADC's SNR would be limited to about 67 dB in the absence of other errors if DNC were disabled (DNC is not shown in Figure 13). The flash ADC threshold errors were chosen randomly with a standard deviation of $\Delta/25$. The distortion coefficients of the first stage were chosen to be $\alpha_{1,1} = -0.085$ and $\alpha_{3,1} = -0.3$, and the remaining stages' distortion coefficients are similar and are consistent with the measured results reported in [43].

The SNDR and SFDR values shown in Figure 17 correspond to a 0 dBFS sinusoidal input signal with a frequency of $0.39f_s$ where f_s is the input sample-rate of the pipelined ADC. Each pair of SNDR and SFDR values were obtained by simulating the pipelined ADC with the sinusoidal input sequence but with the HDC blocks using nonlinearity coefficients that were obtained from a previous simulation with a constant pipelined ADC input sequence. Each SNDR and SFDR value is plotted versus the amplitude of the constant input sequence for which the corresponding nonlinearity coefficients were estimated. As expected from the problem explanation above, there are significant reductions (of approximately 16 dB) in SNDR and SFDR when the HDC coefficients are measured for certain constant input signals.

As demonstrated by the examples described above, the extent to which the quantization error in each pipeline stage is correlated with $c_{1,k}[n]$ and $c_{3,k}[n]$ depends on the pipelined ADC's input sequence. Therefore, it makes sense that the accuracy of the HDC coefficient estimation process depends on the pipelined ADC's input sequence. This effect is exacerbated when the pipelined ADC's input sequence is such that the mean squared value of $e_{ADC,k}[n]$ is large during the coefficient estimation process, particularly for $k = 1$. In these cases $\tilde{\eta}_{2,k}$ is so large that the second term in the factor of $\tilde{\gamma}_{3,k}$ in (69) is dominant and effectively amplifies any error in $\tilde{\gamma}_{3,k}$. The authors have verified that the large dips in SNDR and SFDR shown in Figure 17 correspond to these cases.

IV. SOLUTION TO THE QUANTIZATION ERROR PROBLEM

As described in Section II the HDC blocks estimate their nonlinearity coefficients sequentially, so only one HDC block in the pipelined ADC is in the process of estimating its stage's nonlinearity coefficients at any given time. Therefore, whenever the HDC_j block is in the process of estimating the $\alpha_{1,j}$ and $\alpha_{3,j}$ coefficients, the calibration sequences in the subsequent pipeline stages are not necessary, i.e., calibration sequence $c_k[n]$ need not be added to the output of flash ADC_k for any $k > j$.

The proposed solution to the problem described in the previous section is to replace the $c_k[n]$ sequences for $k = j+1, j+2, \dots, 5$, where j is the number of the stage in which the nonlinearity coefficients are currently being estimated, by new se-

quences, $d_k[n]$, designed to cancel the unwanted correlations. In the example described in the previous section, the problem is that the HDC_1 block estimates $\alpha_{1,1}$ and $\alpha_{3,1}$ poorly when the pipelined ADC input signal is such that $e_{\text{ADC},2}[n]$ in (76) is correlated with either $c_{1,1}[n]$ or $c_{3,1}[n]$. In general the problem is that the HDC_j block estimates $\alpha_{1,j}$ and $\alpha_{3,j}$ poorly when the pipelined ADC input signal is such that $e_{\text{ADC},k}[n]$ is correlated with either $c_{1,j}[n]$ or $c_{3,j}[n]$ for any $k > j$. This problem can be avoided if the $d_k[n]$ sequences satisfy

$$\frac{1}{P} \sum_{n=m}^{m+P-1} (e_{\text{ADC},k}[n] + d_k[n]) c_{3,j}[n] \rightarrow 0 \quad (79)$$

and

$$\frac{1}{P} \sum_{n=m}^{m+P-1} (e_{\text{ADC},k}[n] + d_k[n]) c_{1,j}[n] \rightarrow 0, \quad (80)$$

in probability as $P \rightarrow \infty$ for $k = j+1, j+2, \dots, 5$.

A. Implementation Details

This sub-section describes the implementation details of the proposed solution. A detailed explanation of why it works, and, therefore, the motivation underlying its design, is deferred to Sub-Sections IV-B and IV-C.

Figure 18 shows the example pipelined ADC described previously with HDC applied to the first four stages for the case in which the HDC_1 block is in the process of estimating $\alpha_{1,1}$ and $\alpha_{3,1}$. The blocks labeled RD_k , for $k = 2, 3, 4$, and 5 , are called *residue decorrelator* (RD) blocks because they generate the above-mentioned $d_k[n]$ sequences. Whenever the HDC_j block for $j \neq 1$ is in the process of estimating $\alpha_{1,j}$ and

$\alpha_{3,j}$, the block diagram changes from that shown in Figure 18 only in that $c_k[n]$ is added to the output of flash ADC $_k$ in place of $d_k[n]$ for $k = 2, 3, \dots, j$.[†]

Figure 19 shows a block diagram of the RD $_k$ block configured for the case in which the HDC $_j$ block with $j < k$ is in the process of estimating $\alpha_{1,j}$ and $\alpha_{3,j}$. The function h shown in the figure is defined as

$$h(x) = \begin{cases} 1, & \text{if } x > 0, \\ -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0. \end{cases} \quad (81)$$

The HDC blocks are the same as described in Section II, except they use different correlation sequences than given by (63) as described below. The calibration sequences are as given by (60) with $N = 4$.

The modified correlation sequences used in both the HDC and RD blocks are

$$c_{1,j}[n] = \begin{cases} -\frac{1}{4} \frac{c_j[n]}{A}, & \text{if } c_j[n] \neq 0, \\ c_a[n], & \text{if } c_j[n] = 0 \text{ and } t_{1,j}[n] = t_{2,j}[n], \\ c_b[n], & \text{otherwise,} \end{cases} \quad (82)$$

and

$$c_{3,j}[n] = \begin{cases} -\frac{1}{4} \frac{s_j[n]}{A^3} & \text{if } c_j[n] \neq 0, \\ c_a[n], & \text{if } c_j[n] = 0 \text{ and } t_{1,j}[n] = t_{2,j}[n], \\ -c_b[n], & \text{otherwise,} \end{cases} \quad (83)$$

where

[†] While adding $c_k[n]$ to the output of flash ADC $_k$ for $k < j$ is not absolutely necessary, it is done anyway to dither the flash ADCs. This reduces the unwanted correlations, thereby slightly improving the accuracy of the HDC correlations.

$$\begin{aligned}
s_j[n] = & t_{1,j}[n]t_{2,j}[n]t_{3,j}[n] + t_{1,j}[n]t_{2,j}[n]t_{4,j}[n] \\
& + t_{2,j}[n]t_{3,j}[n]t_{4,j}[n] + t_{1,j}[n]t_{3,j}[n]t_{4,j}[n],
\end{aligned} \tag{84}$$

and $c_a[n]$ and $c_b[n]$ are two-level zero-mean pseudo-random sequences that take on values of $\pm B$ and are independent from each other and from $c_j[n]$. The constant B must satisfy:

$$B \geq \frac{4e_{ADC,max}}{\Delta} - 1. \tag{85}$$

Therefore, when $e_{ADC,max} = \Delta/2$ (i.e., in the absence of non-ideal circuit behavior), $B = 1$ is acceptable, but in practice B must be somewhat greater than one to accommodate the maximum anticipated non-ideal flash ADC errors. Any value of B that satisfies (85) will work, but increasing B increases the HDC convergence time, so B should be chosen as small as possible subject to the constraint of (85).

The RD block solution described above involves only digital circuitry, and the expected area and power dissipation of the circuitry are small compared to those of the digital circuitry required by the HDC technique without the RD block solution. Therefore, the RD block solution is not expected to contribute significantly to the overall circuit area or power dissipation of typical pipelined ADCs to which it would be applied.

B. Theory of Operation

As depicted in Figure 18, each RD_k block operates on the output of the HDC_k block and generates the $d_k[n]$ sequence. As explained below, it forms a feedback loop which adaptively adjusts $d_k[n]$ such that the correlations of the HDC_k block's output

sequence with $c_{1,j}[n]$ and $c_{3,j}[n]$ converge to zero. The output of the HDC_k block satisfies

$$r_k[n] \Big|_{\text{corrected}} \cong -e_{\text{ADC},k}[n] - d_k[n] \quad (86)$$

to a high degree of accuracy, so this causes (79) and (80) to be satisfied with a high degree of accuracy.

It follows from (86) and Figure 19 that the outputs of the top and bottom accumulators in the RD_k block can be written as

$$f_k[n] \cong -\sum_{i=0}^n (e_{\text{ADC},k}[i] + d_k[i]) (c_{1,j}[i] - c_{3,j}[i]) \quad (87)$$

and

$$g_k[n] \cong -\sum_{i=0}^n (e_{\text{ADC},k}[i] + d_k[i]) (c_{1,j}[i] + c_{3,j}[i]). \quad (88)$$

As explained below, the RD_k block chooses the $d_k[n]$ sequence to ensure that both $f_k[n]$ and $g_k[n]$ are bounded in probability for all n . This implies that

$$\frac{1}{n} (f_k[n] - g_k[n]) \rightarrow 0 \quad (89)$$

and

$$\frac{1}{n} (f_k[n] + g_k[n]) \rightarrow 0 \quad (90)$$

in probability as $n \rightarrow \infty$, and therefore that both (79) and (80) hold.

Table 1 shows the $t_{i,j}[n]/A$ values along with the corresponding values of $c_j[n]/A$, $c_{1,j}[n]$, $c_{3,j}[n]$, and $c_{1,j}[n] \pm c_{3,j}[n]$, for each of the 16 possible sets values that $t_{1,j}[n]$, $t_{2,j}[n]$, $t_{3,j}[n]$, $t_{4,j}[n]$ can take on. By definition, each of the 16 possible sets oc-

curs with a probability of 1/16. In the following, several observations are made from Table 1 to show that (89) and (90) hold in probability as $n \rightarrow \infty$.

Table 1 indicates that one or the other of $c_{1,j}[n] + c_{3,j}[n]$ and $c_{1,j}[n] - c_{3,j}[n]$ is guaranteed to be zero at each time index n , so it can be seen from Figure 19 that only one of $f_k[n]$ and $g_k[n]$ changes each time n is incremented. Consequently, for each value of n , $d_k[n]$ influences $f_k[n]$ but has no effect on $g_k[n]$ if $c_{1,j}[n] - c_{3,j}[n]$ is non-zero, or vice versa if $c_{1,j}[n] + c_{3,j}[n]$ is nonzero. This is why it is possible for $d_k[n]$ to keep both $f_k[n]$ and $g_k[n]$ bounded; each RD block implements two feedback loops that are interlaced with each other such that no crosstalk occurs between them.

It follows from Figure 19, (81), and (86) that

$$f_k[n] = f_k[n-1] - e_{ADC,k}[n](c_{1,j}[n] - c_{3,j}[n]) - \frac{\Delta}{4} \operatorname{sgn}\{f_k(n-1)\} |c_{1,j}[n] - c_{3,j}[n]|, \quad (91)$$

and

$$g_k[n] = g_k[n-1] - e_{ADC,k}[n](c_{1,j}[n] + c_{3,j}[n]) - \frac{\Delta}{4} \operatorname{sgn}\{g_k(n-1)\} |c_{1,j}[n] + c_{3,j}[n]|, \quad (92)$$

where $\operatorname{sgn}\{x\} = 1$ if $x \geq 0$ and $\operatorname{sgn}\{x\} = -1$ otherwise. The last terms in (91) and (92) correspond to $d_k[n](c_{1,j}[n] - c_{3,j}[n])$ and $d_k[n](c_{1,j}[n] + c_{3,j}[n])$, respectively. For each n , one of these terms is zero and the other has the opposite sign of the corresponding value of $f_k[n-1]$ or $g_k[n-1]$.

To show that this ensures $f_k[n]$ and $g_k[n]$ are bounded in probability for all n , it remains to show that in each of (91) and (92) the average magnitude of the last term is at least as large as that of the term proportional to $e_{ADC,k}[n]$. That is, for (91) it remains to show that

$$\frac{\Delta}{4} \mathbb{E} \left\{ \left| c_{1,j}[n] - c_{3,j}[n] \right| \middle| c_{1,j}[n] - c_{3,j}[n] \neq 0 \right\} \quad (93)$$

is at least as large as

$$\left| \mathbb{E} \left\{ e_{ADC,k}[n] (c_{1,j}[n] - c_{3,j}[n]) \middle| c_{1,j}[n] - c_{3,j}[n] \neq 0 \right\} \right| \quad (94)$$

and for (92) it remains to show that

$$\frac{\Delta}{4} \mathbb{E} \left\{ \left| c_{1,j}[n] + c_{3,j}[n] \right| \middle| c_{1,j}[n] + c_{3,j}[n] \neq 0 \right\} \quad (95)$$

is at least as large as

$$\left| \mathbb{E} \left\{ e_{ADC,k}[n] (c_{1,j}[n] + c_{3,j}[n]) \middle| c_{1,j}[n] + c_{3,j}[n] \neq 0 \right\} \right|. \quad (96)$$

As mentioned above each row of Table 1 corresponds to one of the 16 possible sets of values that $t_{1,j}[n]$, $t_{2,j}[n]$, $t_{3,j}[n]$, $t_{4,j}[n]$ can take on, so the rows correspond to mutually exclusive events that each occur with a probability of 1/16. Therefore Table 1 implies that (93) evaluates to $\frac{2}{3} \cdot \Delta(1+B)/4$ and (95) evaluates to $\Delta(1+B)/4$.

Table 1 indicates that the correlation sequences contain $c_a[n]$ and $c_b[n]$ only when the calibration sequence is zero. This implies that the input sequence to the flash ADC in the $(j+1)$ th stage is statistically independent of $c_a[n]$ and $c_b[n]$, so $e_{ADC,j+1}[n]$ is also statistically independent of $c_a[n]$ and $c_b[n]$. Since $c_a[n]$ and $c_b[n]$ are both zero-mean sequences, it follows that the correlations of $e_{ADC,j+1}[n]$ with the correlation sequences are both zero. Therefore, Table 1 implies that (94) and (96) must be no larger than $\frac{2}{3}e_{ADC,max}$ and $e_{ADC,max}$, respectively.

It follows that (93) and (95) are at least as large as both (94) and (96) if $\Delta(1+B)/4 \geq e_{ADC,max}$, which is equivalent to (85). Therefore, (79) and (80) are satisfied

in probability as $P \rightarrow \infty$ for $k = j+1, j+2, \dots, 5$ by the solution described in Section IV-A.

C. Calibration Sequence Choice

As described in Section II, the number, N , of two-level sequences added to form the calibration sequence $c_j[n]$ must be at least as large as the highest order of nonlinear distortion to be cancelled by the HDC algorithm. Therefore, the minimum possible value of N is 3 when the residue amplifiers are well-modeled by (58).

However, if $N = 3$ were used with the solution presented in Section IV-B, the magnitude of $d_k[n]$ would have had to be $e_{ADC,max}$ to ensure full cancellation of the unwanted correlation terms for all pipelined ADC input signals. Since $e_{ADC,max}$ is greater than $\Delta/2$ in practice, such $d_k[n]$ sequences would exceed the over-range margin of the pipeline stages.

The solution proposed in Section IV-B uses $N = 4$ to avoid this problem. With $N = 4$ the calibration sequence, $c_j[n]$, is zero 6/16 of the time (see Table 1), and whenever this happens $e_{ADC,j+1}[n]$ is uncorrelated with the correlation sequences as explained in Section IV-B. However, as shown in Section IV-B, the $d_{j+1}[n]$ sequence is correlated with the correlation sequences regardless of whether the calibration sequence is zero, which makes it possible to operate properly with $d_k[n]$ sequences that have a magnitude of only $\Delta/4$. Therefore, the $d_k[n]$ sequences use exactly the same portion of the over-range margin as the calibration sequences.

D. Simulation Results

Figure 20 shows computer simulation results that are identical to those which produced the data shown in Figure 17, except that the simulated pipelined ADC was enhanced with the modified calibration sequences and RD blocks as described above. As in Figure 17, the SNDR and SFDR values shown in Figure 11 correspond to a 0 dBFS sinusoidal input signal with frequency of $0.39f_s$. Each pair of SNDR and SFDR values were obtained by simulating the enhanced pipelined ADC with the sinusoidal input sequence but with the HDC blocks using $\alpha_{1,k}$ and $\alpha_{3,k}$ coefficients that were obtained previously by simulating the pipelined ADC with a constant input. Each SNDR and SFDR value is plotted versus the input sequence for which $\alpha_{1,k}$ and $\alpha_{3,k}$ coefficients were estimated.

A comparison of Figures 17 and 20 indicates that the proposed technique improves the worst-case SNDR and SFDR values by 12.6 dB and 15.8 dB, respectively. Numerous other simulation experiments performed by the authors have yielded quantitatively similar results. The proposed technique involves several approximations as described above, so it does not completely eliminate the variability of the SNDR and SFDR with the input signal, but it greatly reduces the variability as intended.

As explained in Section III, accurate nonlinearity coefficient estimation for all constant input sequences implies accurate coefficient convergence for all non-constant input sequences too. Therefore, the results support the assertion that the solution to the quantization-induced convergence error problem presented in this paper enables correct operation of the HDC_k blocks regardless of the pipelined ADC input

signal.

V. ACKNOWLEDGEMENTS

Chapter 2, in full, has been accepted for publication by the IEEE Transactions on Circuits and Systems I: Regular Papers. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

VI. FIGURES

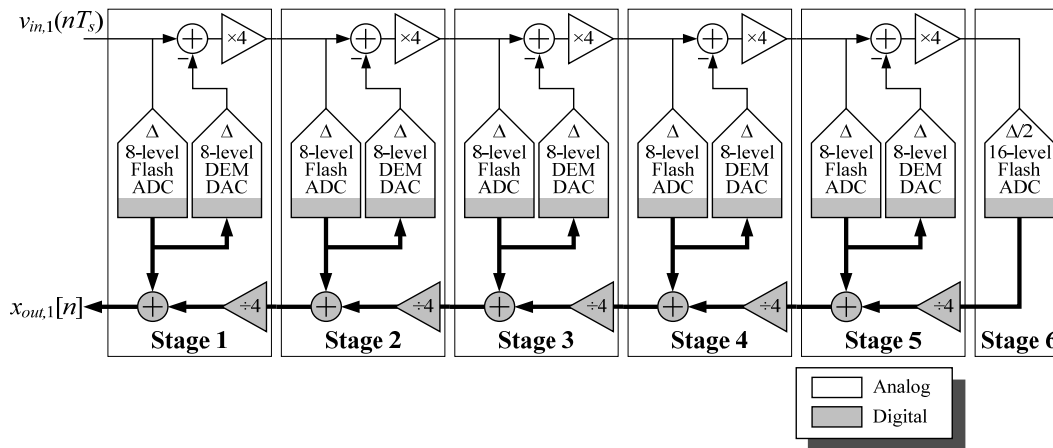
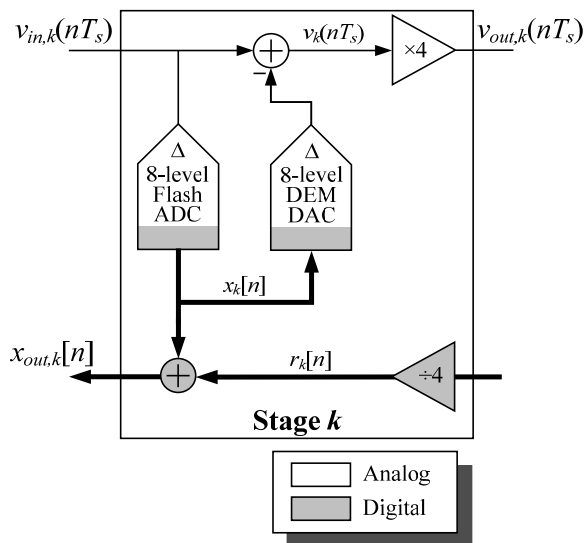


Figure 11 Example pipelined ADC.

Figure 12 The k th pipelined ADC stage for $k = 1, \dots, 5$ with labels that indicate variable names used throughout the paper.

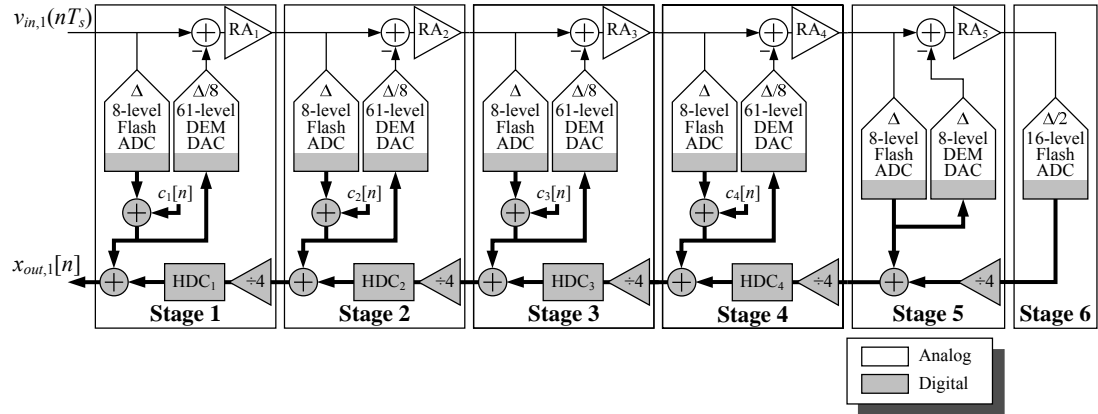


Figure 13 The example pipelined ADC with HDC applied to the first four stages.

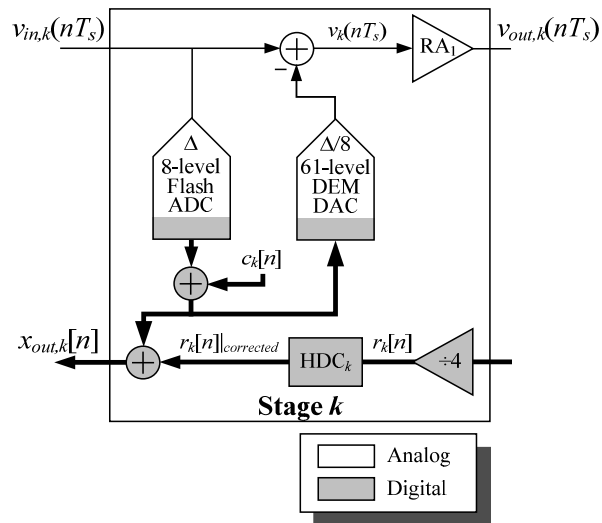


Figure 14 The k th pipelined ADC stage with HDC for $k = 1, 2, 3, 4$, with labels that indicate variable names used throughout the paper.

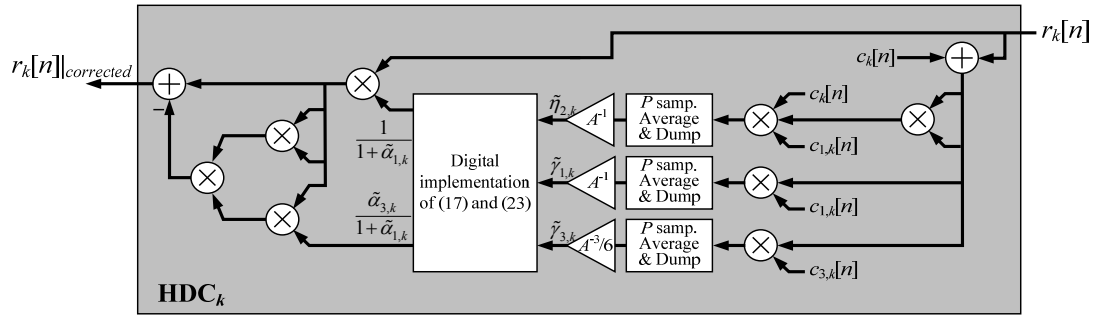


Figure 15 Details of the HDC_k block.

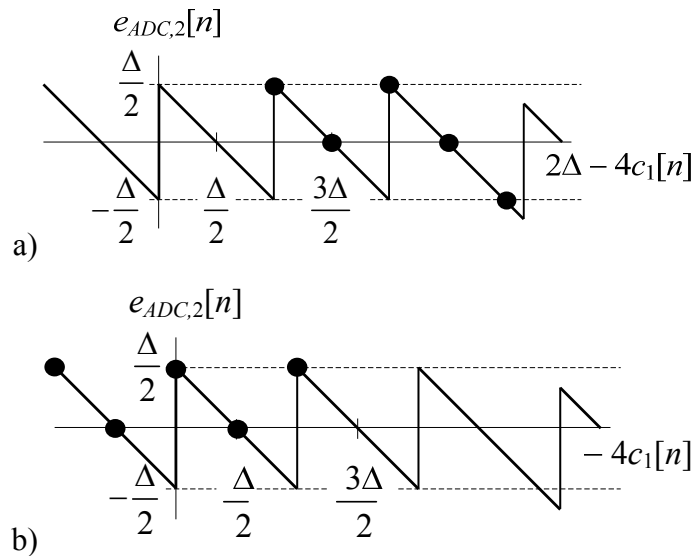


Figure 16 The five values of $e_{ADC,2}[n]$ that occur when the input to flash ADC_2 is (a) $2\Delta - 4c_1[n]$ and (b) $-4c_1[n]$ superimposed on a plot of the input-output characteristic of flash ADC_2 (which contains a single threshold error).

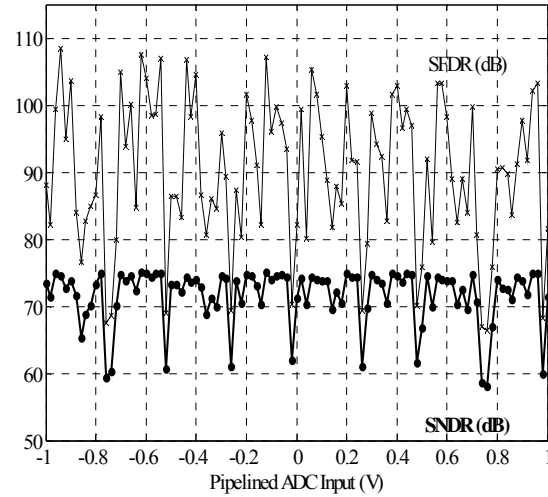


Figure 17 Simulation results for pipelined ADC shown in Figure 3 with $f_{in} = 0.39f_s$ and 0 dBFS sinusoidal input signal.

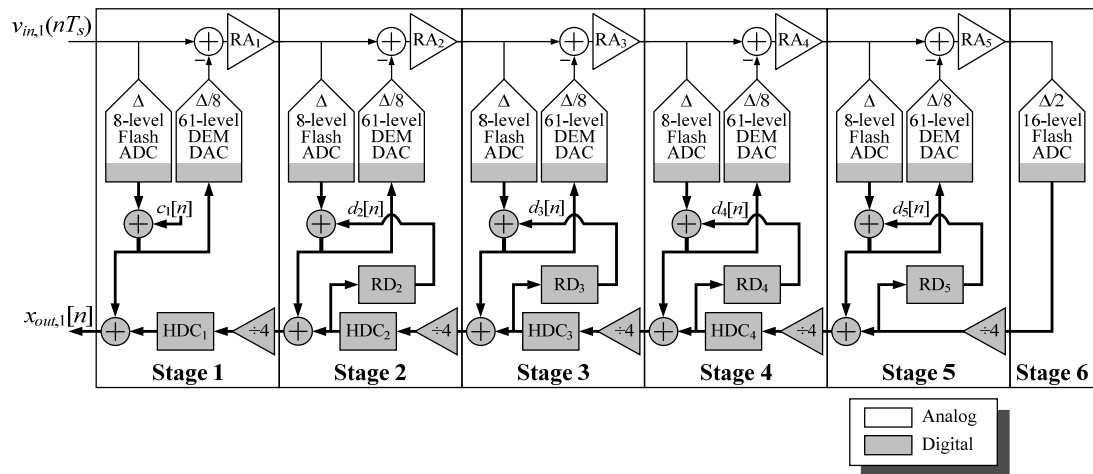


Figure 18 Pipelined ADC configuration during estimation of $\alpha_{1,1}$ and $\alpha_{3,1}$ with HDC in the first stage and RD applied to the remaining stages.

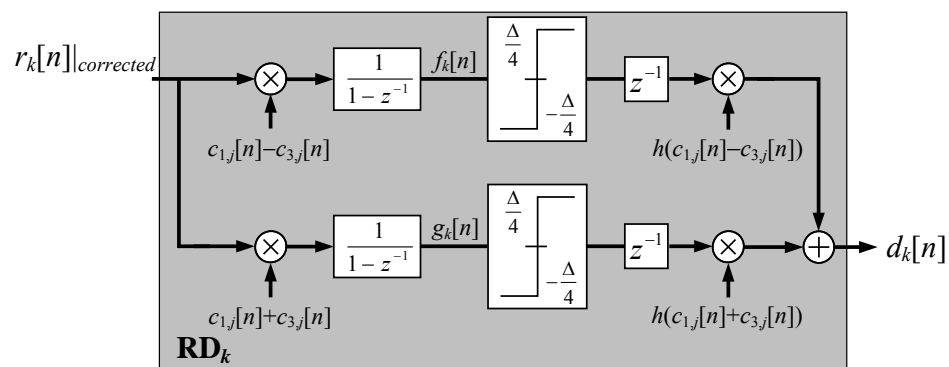


Figure 19 Details of the RD_k block.

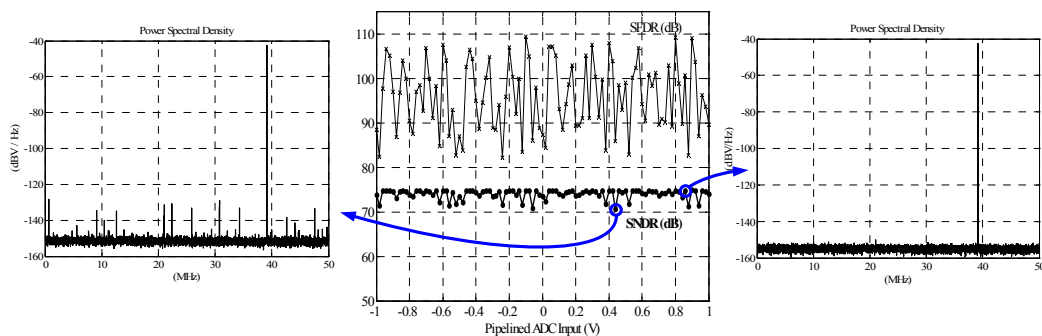


Figure 20 Simulation results for pipelined ADC shown in Figure 8 with $f_{in} = 0.39f_s$ and 0 dBFS sinusoidal input signal and with representative Power Spectral Density plots of the output.

VII. TABLES

Table 1 All possible sets of values that $t_{1,j}[n]/A$, $t_{2,j}[n]/A$, $t_{3,j}[n]/A$, $t_{4,j}[n]/A$ can take on along with the corresponding values of $c_j[n]/A$, $c_{1,j}[n]$, $c_{3,j}[n]$, and $c_{1,j}[n] \pm c_{3,j}[n]$.

$t_{1,j}[n]/A$	$t_{2,j}[n]/A$	$t_{3,j}[n]/A$	$t_{4,j}[n]/A$	$c_j[n]/A$	$c_{1,j}[n]$	$c_{3,j}[n]$	$c_{1,j}[n] + c_{3,j}[n]$	$c_{1,j}[n] - c_{3,j}[n]$
-1	-1	-1	-1	-4	1	1	2	0
-1	-1	-1	1	-2	1/2	-1/2	0	1
-1	-1	1	-1	-2	1/2	-1/2	0	1
-1	-1	1	1	0	$c_a[n]$	$c_a[n]$	$2c_a[n]$	0
-1	1	-1	-1	-2	1/2	-1/2	0	1
-1	1	-1	1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
-1	1	1	-1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
-1	1	1	1	2	-1/2	1/2	0	-1
1	-1	-1	-1	-2	1/2	-1/2	0	1
1	-1	-1	1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
1	-1	1	-1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
1	-1	1	1	2	-1/2	1/2	0	-1
1	1	-1	-1	0	$c_a[n]$	$c_a[n]$	$2c_a[n]$	0
1	1	-1	1	2	-1/2	1/2	0	-1
1	1	1	-1	2	-1/2	1/2	0	-1
1	1	1	1	4	-1	-1	-2	0

VIII. REFERENCES

39. B. Murmann, B. Boser, "A 12b 75MS/s Pipelined ADC using Open-Loop Residue Amplification," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2040-2050, December 2003.
40. J. P. Keane, P. J. Hurst, S. H. Lewis, "Background Interstage Gain Calibration Technique for Pipelined ADCs," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 1, pp. 32-43, January 2005.
41. H. Van de Vel, B. A. J. Buter, H. van der Ploeg, M. Vertregt, G. J. G. M. Geelen, E. J. F. Paulus, "A 1.2-V 250-mW 14-b 100-MS/s Digitally Calibrated Pipeline ADC in 90-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1047-1056, April 2009.
42. A. Panigada, I. Galton, "Digital Background Correction of Harmonic Distortion in Pipelined ADCs," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 53, no. 9, pp. 1885-1895, September 2006.
43. A. Panigada, I. Galton, "A 130 mW 100 MS/s Pipelined ADC with 69 dB SNDR Enabled by Digital Harmonic Distortion Correction," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3314-3328, December 2009.
44. O. A. Horna, "A 150 Mbps A/D and D/A conversion system," *Comsat Technical Review*, vol. 2, no. 1, pp. 52-57, 1972.
45. S. S. Taylor, "High speed analog-to-digital conversion in integrated circuits," Ph.D. dissertation, Univ. of Calif., Berkeley, pp. 30-42, 1978.
46. S. Lewis, P. Gray, "A Pipelined 5MHz 9b ADC," *International Solid-State Circuits Conference, Digest of Technical Papers*, pp. 210-211, 405, February 1987.

Chapter 3

Convergence Time Analysis for Harmonic Distortion Correction

Used in Pipelined ADCs

Abstract—The Harmonic Distortion Correction (HDC) algorithm is a digital calibration algorithm used to eliminate errors that arise at the output of pipelined ADCs due to weak nonlinearities in the residue amplifiers. In order to eliminate these errors, the HDC algorithm has to estimate some of the coefficients that model residue amplifiers' weakly-nonlinear behavior. These estimates are obtained by considering one residue amplifier at the time and averaging many samples of the signal that correspond to its output. Prior to this work, the number of samples that needed to be averaged to obtain reliable coefficient estimates was determined ad-hoc, via few simulations. This paper provides a rigorous analysis of the relationship between the number of averaged samples and the accuracy of the HDC algorithm's estimates.

I. INTRODUCTION

Pipelined analog-to-digital converters (ADCs) are intended for applications requiring high-resolution (greater than 10 bit), high-speed (greater than 50Msps) ADCs, which can be found in communication receivers, ultrasound imaging systems and TV demodulators. As the power consumption of these systems is an ever-present concern, it is essential that pipelined ADCs consume minimal power while achieving desirable performance. If implemented in CMOS technology, the minimal power con-

straint usually entails reduction of current consumption in the pipelined ADCs' residue amplifiers which can cause a change in the residue amplifier's transfer function from linear to weakly-nonlinear [47]. This results in distortion terms at the amplifier's output, which often lower the signal-to-noise-and-distortion ratio (SNDR) of the converter.

In recent years, much of the research in the area of pipelined ADCs has focused on mitigating the distortion due to the residue amplifier's weakly nonlinear behavior. Instead of pursuing the traditional approach of optimizing the analog design, the focus has been on developing digital calibration techniques that successfully estimate and cancel the error due to residue amplifier's nonlinear terms [48] – [51]. The best among these techniques are all-digital, do not require additional or more precise analog components, fractionally contribute to overall power consumption, and have the ability to estimate and track the changes of the residue amplifier's distortion regardless of the pipelined ADC's input.

The most robust technique with these properties that is known to the authors is the harmonic distortion correction (HDC) algorithm [48, 51]. The basis of the HDC algorithm is a random sequence that is added to the calibration stage and which is used to estimate the coefficients describing the residue amplifier's weak nonlinearity. The estimates are obtained by averaging the sum of the scaled digital values corresponding to the residue amplifier's output signal. This information, along with the scaled version of the residue amplifier's output, is used to eliminate the residue amplifier's nonlinear terms from the pipelined ADC output.

The estimation of the coefficients by the HDC algorithm is performed by averaging many samples of the signal. The accuracy of the estimates, defined as the difference between the true value of the coefficient and its estimated value, depends on the number of samples used in the process. As previously mentioned, the HDC algorithm is the most robust residue amplifier calibration technique, so it is of interest to investigate the number of samples that need to be averaged so that the obtained coefficient estimates are reliable. This paper presents a rigorous analysis of how error in the estimation of residue amplifier's coefficients decreases with the number of averaged samples.

The paper is organized as follows. Section II of the paper provides necessary background information on the theory of pipelined ADCs and the HDC algorithm, similar to what is published in [48, 51, 52]. In Section III, relationship between the number of averaged samples and the accuracy of the residue amplifier's coefficient estimates is analyzed. Section IV contains simulation results which complement the analysis in Section III.

II. BACKGROUND INFORMATION

A. Pipelined ADC Overview

A block diagram of an ideal 6-stage pipelined ADC is shown in Figure 21. The first 5 stages, denoted as *Stage k* for $k = 1, \dots, 5$, are structurally identical 3-bit stages, while the last, *Stage 6*, is a 4-bit stage. Each k th stage, shown in Figure 22, consists of an 8-level flash ADC ($FADC_k$), an 8-level dynamic element matching

digital-to-analog converter (DEM DAC_k) and a residue amplifier with the gain of 4. The last stage is just a 16-level flash ADC.

At each sampling time nT_s , where T_s is the sampling period of the converter, the pipelined ADC converts its analog input sequence, $v_{in,1}(nT_s)$, into a digital output, $x_{out}[n]$. In the process of conversion, FADC_k quantizes the input to the stage, $v_{in,k}(nT_s)$, and represents it as 1 of 8 possible digital codes at its output. The FADC_k output $x_k[n]$, shown in Figure 22, is given by

$$x_k[n] = v_{in,k}(nT_s) + e_{ADC,k}[n], \quad (97)$$

where $e_{ADC,k}[n]$ is the flash ADC's quantization error. Nominally, the magnitude of $e_{ADC,k}[n]$ is at most $\Delta/2$, where Δ is the step size of the flash ADC. Once $x_k[n]$ is available, DEM DAC_k converts it into analog format. The output of DEM DAC_k is subtracted from the input to the stage, and from (97) and Figure 22 the difference, $v_k(nT_s)$, is

$$v_k(nT_s) = -e_{ADC,k}[n]. \quad (98)$$

The residue amplifier amplifies $v_k(nT_s)$ by 4, and its output, $v_{out,k}(nT_s)$, is an input to the following stage, i.e.

$$v_{out,k}(nT_s) = 4v_k(nT_s) = v_{in,k+1}(nT_s). \quad (99)$$

The process repeats until the last stage where FADC₆ quantizes $v_{in,6}(nT_s)$ as in (97), but with $e_{ADC,6}[n]$ bounded by $\Delta/4$. From Figure 21, (97) – (99) it follows that all the digital codes $x_{k-1}[n]$ to $x_6[n]$ are combined together to form the *digitized residue* of the k th stage $r_k[n]$. In the case of the pipelined ADC in Figure 21, the digitized

residue is ideal and given by

$$r_k[n] = r_k[n]_{ideal} = v_k(nT_s) + 4^{k-6} e_{ADC6}[n]. \quad (100)$$

From (100), it follows that the pipelined ADC output in Figure 21 can be written as

$$x_{out}[n] = x_{out}[n]_{ideal} = x_1[n] + r_1[n]_{ideal}. \quad (101)$$

Unfortunately, in practice, each component of the pipelined ADC behaves non-ideally and thus represents a source of error that causes $x_{out}[n]$ to deviate from (101). For the remainder of the discussion in this paper it is assumed that the only source of error in the k th stage is its residue amplifier. Such residue amplifier, denoted as RA_k , does not amplify by 4, and its behavior cannot be accurately modeled with a linear transfer function. In this case, a more realistic description of RA_k 's input-output characteristic is given by

$$v_{out,k}(nT_s) \cong 4\left((1 + \alpha_{1,k})v_k(nT_s) + \alpha_{3,k}v_k^3(nT_s)\right) \quad (102)$$

where $\alpha_{1,k}$ and $\alpha_{3,k}$ are the residue amplifier's gain error and third order distortion coefficients, respectively. From (97), (99) and Figures 21 and 22, it follows that

$$r_k[n] = \frac{1}{4}\left(r_{k+1}[n] + v_{out,k}(nT_s) + e_{ADC,k+1}[n]\right), \quad (103)$$

and thus from (97) – (103),

$$r_k[n] = r_k[n]_{ideal} + \sum_{l=k}^5 4^{k-l} \left(\alpha_{1,l}v_l(nTs) + \alpha_{3,l}v_l^3(nTs)\right). \quad (104)$$

By replacing $r_1[n]_{ideal}$ in (101) with $r_1[n]$ given by (104), it follows from (101) that

$$x_{out}[n] = x_{out}[n]_{ideal} + \sum_{k=1}^5 4^{1-k} \left(\alpha_{1,k}v_k(nTs) + \alpha_{3,k}v_k^3(nTs)\right). \quad (105)$$

Thus, the output of the pipelined ADC deviates from its ideal value due to the non-ideal residue amplifiers, i.e. the non-zero values of $\alpha_{1,k}$ and $\alpha_{3,k}$.

The objective of each HDC_k block is to estimate the $\alpha_{1,k}$ and $\alpha_{3,k}$ coefficients, and use these estimates to remove the terms which are proportional to $\alpha_{1,k}$ and $\alpha_{3,k}$ from (104) (and, as a consequence, also from (105)). A brief overview of the HDC algorithm, necessary for the derivations in the subsequent sections of the paper, is given next.

B. HDC Algorithm Overview

Figure 23 shows the pipelined ADC from Figure 21 which residue amplifiers are non-ideal and behave according to (102). The HDC algorithm is implemented in the first four stages, and the shown configuration corresponds to the state of the system during estimation of $\alpha_{1,1}$ and $\alpha_{3,1}$. Each of the four stages contains a digital block, HDC_k , which realizes all the necessary operations as determined by the algorithm design in [48, 52]. Next, a description of the HDC_k block's operation is presented. In particular, the operation of the HDC_1 block is described during estimation of $\alpha_{1,1}$ and $\alpha_{3,1}$.

The estimation of the coefficients is performed sequentially, starting with the last calibration stage first. In Figure 23, estimates of $\alpha_{1,j}$ and $\alpha_{3,j}$ coefficients for $j \in \{2, 3, 4\}$, denoted as $\tilde{\alpha}_{1,j}$ and $\tilde{\alpha}_{3,j}$ respectively, are obtained prior to the HDC_1 estimation, and are used to correct $r_j[n]$ so that

$$r_j[n]|_{corrected} = \frac{r_j[n]}{(1 + \tilde{\alpha}_{1,j})} - \frac{\tilde{\alpha}_{3,j}}{(1 + \tilde{\alpha}_{1,j})^4} r_j^3[n]. \quad (106)$$

In order to enable the HDC₁ estimation, the sequence $c_1[n]$ is added at the output of FADC₁. The sequence $c_1[n]$ is the sum of 4 independent, identically distributed (i.i.d) 2-level sequences $t_{i,1}[n]$, for $i \in \{1, 2, 3, 4\}$, such that

$$c_1[n] = \sum_{i=1}^4 t_{i,1}[n]. \quad (107)$$

The amplitude, A , of each $t_{i,1}[n]$, is $\Delta/16$, which ensures that $c_1[n]$ fits within the over-range margin of RA₁ [48, 51]. The addition of $c_1[n]$ necessitates an increase in the resolution of DEM DAC₁ from 8 to 61 levels as described in [48, 52], and results in the residue amplifier's input

$$v_1(nT_s) = -e_{ADC,1}[n] - c_1[n]. \quad (108)$$

The output of RA₁ is given by (102), and it follows from (102) and (108) that $v_{out,1}(nT_s)$ contains the terms $\alpha_{1,1}c_1[n]$ and $\alpha_{3,1}c_1^3[n]$. From Figures 22 and 23 it follows that these terms propagate into $r_1[n]$, and the HDC₁ block estimates $\alpha_{1,1}$ and $\alpha_{3,1}$ by multiplying $r_1[n]$ with the HDC sequences. The HDC sequences, $c_{1,1}[n]$ and $c_{3,1}[n]$, are equal to $-c_1[n]/(4A)$ and $-s_1[n]/(4A^3)$, respectively, during the samples that $c_1[n] \neq 0$, where $s_1[n]$ is given by

$$s_1[n] = t_{1,1}[n]t_{2,1}[n]t_{3,1}[n] + t_{1,1}[n]t_{2,1}[n]t_{4,1}[n] + t_{2,1}[n]t_{3,1}[n]t_{4,1}[n] + t_{1,1}[n]t_{3,1}[n]t_{4,1}[n]. \quad (109)$$

The correct estimates of $\alpha_{1,1}$ and $\alpha_{3,1}$ can be obtained if $v_{out,1}(nT_s)$ is the only term in $r_1[n]$ that is correlated with the HDC sequences. In general, however, a part of $r_2[n]|_{corrected}$ that does not perfectly recombine with $e_{ADC,2}[n]$ propagates into $r_1[n]$,

and is correlated with $c_{1,1}[n]$ and $c_{3,1}[n]$ [48, 52]. For this reason, the RD_j blocks are added to all the stages subsequent to the calibration stage, as shown in Figure 23. The RD_j blocks generate the sequences $d_j[n]$ which ensure that the leakage of the signals from the stages subsequent to *Stage 1* does not affect the estimation of $\alpha_{1,1}$ and $\alpha_{3,1}$. The amplitude of $d_j[n]$ is $4A$ (i.e. $\Delta/4$) and with $d_j[n]$ added to the system, from (103), and Figure 23

$$r_{j-1}[n] = \frac{1}{4} \left(r_j[n] \Big|_{\text{corrected}} + v_{\text{out},j-1}(nT_s) + e_{ADC,j}[n] + d_j[n] \right). \quad (110)$$

The sequences $d_j[n]$ require that the HDC sequences are non-zero when $c_1[n] = 0$, so that

$$c_{1,1}[n] = \begin{cases} -\frac{1}{4} \frac{c_1[n]}{A}, & \text{if } c_1[n] \neq 0, \\ c_a[n], & \text{if } c_1[n] = 0 \text{ and } t_{1,1}[n] = t_{2,1}[n], \\ c_b[n], & \text{otherwise,} \end{cases} \quad (111)$$

and

$$c_{3,1}[n] = \begin{cases} -\frac{1}{4} \frac{s_1[n]}{A^3} & \text{if } c_1[n] \neq 0, \\ c_a[n], & \text{if } c_1[n] = 0 \text{ and } t_{1,1}[n] = t_{2,1}[n], \\ -c_b[n], & \text{otherwise,} \end{cases} \quad (112)$$

where $c_a[n]$ and $c_b[n]$ are 2-level, zero mean, i.i.d. sequences with amplitude $B \geq 1$. The values that $c_1[n]$, $c_{1,1}[n]$ and $c_{3,1}[n]$ can take on are given in Table 2 for $j = 1$. For the details of the RD_j block and $d_j[n]$ sequence the reader can refer to [52].

All the results in (107) – (112) can be extended to any calibration stage q by replacing the subscript 1 with q in the equations. Generally speaking, in the process

of estimation of $\alpha_{1,q}$ and $\alpha_{3,q}$ coefficients, the HDC_q block obtains $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ which are given by

$$\tilde{\gamma}_{1,q} = \frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{1,q}[n] \quad (113)$$

$$\tilde{\gamma}_{3,q} = \frac{1}{6A^3} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{3,q}[n] \quad (114)$$

$$\tilde{\eta}_{2,q} = \frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])^2 c_q[n] c_{1,q}[n]. \quad (115)$$

From the law of large numbers and the properties of $c_q[n]$, $c_{1,q}[n]$ and $c_{3,q}[n]$, it follows that as $P \rightarrow \infty$,

$$\tilde{\gamma}_{1,q} = \alpha_{1,q} + \left(10A^2 - 3 \frac{\tilde{\eta}_{2,q}}{(1 + \alpha_{1,q})^2} \right) \tilde{\gamma}_{3,q} \quad (116)$$

$$\tilde{\gamma}_{3,q} = \alpha_{3,q} \quad (117)$$

$$\tilde{\eta}_{2,q} = -\frac{1}{P} \sum_{n=m}^{m+P-1} (1 + \alpha_{1,q})^2 e_{ADC,q}^2 [n] c_q[n] c_{1,q}[n]. \quad (118)$$

Thus, for sufficiently large P , the estimates of $\alpha_{1,q}$ and $\alpha_{3,q}$ are computed according to

$$\tilde{\alpha}_{3,q} = \tilde{\gamma}_{3,q} \quad (119)$$

and

$$\tilde{\alpha}_{1,q} = \tilde{\gamma}_{1,q} + \tilde{\gamma}_{3,q} \left(\frac{3\tilde{\eta}_{2,q}}{(1 + \tilde{\gamma}_{1,q})^2} - 10A^2 \right) \cdot \left(\begin{array}{l} 1 - \frac{6\tilde{\eta}_{2,q}}{(1 + \tilde{\gamma}_{1,q})^3} \tilde{\gamma}_{3,q} \\ + \frac{63\tilde{\eta}_{2,q}^2 - 90A^2 \tilde{\eta}_{2,q} (1 + \tilde{\gamma}_{1,q})^2}{(1 + \tilde{\gamma}_{1,q})^6} \tilde{\gamma}_{3,q}^2 \end{array} \right). \quad (120)$$

Note that in the limit of $P \rightarrow \infty$, (120) does not converge to $\alpha_{1,q}$, and therefore (120) does not follow from (116), when solved for $\alpha_{1,q}$. Estimating $\alpha_{1,q}$ by computing an inverse of (116) would be too computationally intensive and undesirable for implementation in a practical system. Instead, (120) provides a simpler expression for the estimate of $\alpha_{1,q}$. However, this simplification comes at the expense of error in $\alpha_{1,q}$ estimation. More information on how (120) approximates the inverse of (116) can be found in [52].

Once $\tilde{\alpha}_{1,q}$ and $\tilde{\alpha}_{3,q}$ are obtained, the HDC_q block corrects the digitized residue according to (106) for $j = q$. The correction of the digitized residue is performed under the assumption that the estimates of $\alpha_{1,q}$ and $\alpha_{3,q}$ are sufficiently accurate. If $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$, and $\tilde{\eta}_{2,q}$ differ from (116) and (118), then, in general, the estimates in (119) and (120) deviate from $\alpha_{3,q}$ and $\alpha_{1,q}$, respectively. In this case, either the unwanted terms in the digitized residue are only partially removed during correction or the correction introduces additional error into the pipelined ADC output, as is shown next.

III. HDC ALGORITHM CONVERGENCE TIME ANALYSIS

Consider an example in which the pipelined ADC in Figure 23 is ideal. The residue amplifier coefficients of its 1st stage are $\alpha_{1,1} = 0$, $\alpha_{3,1} = 0$, and its digitized residue is $r_1[n] = r_1[n]_{ideal}$. Suppose that the HDC_1 block computes $\tilde{\gamma}_{3,1}$ according to (114) by using a single sample at time aT_s during which $r_1[a] + c_1[a] \neq 0$ and $c_{3,1}[a] \neq$

0. It follows from (106), (114) and (119) that $\tilde{\alpha}_{3,1} \neq 0$ and that the corrected digitized residue $r_1[n]_{\text{corrected}}$ contains a third order distortion term. This distortion term propagates into $x_{\text{out}}[n]$ and, in this example, the correction of $r_1[n]$ by the HDC_1 block degrades rather than improves the pipelined ADC's performance.

If, in the example above, more samples were averaged, the estimate of $\alpha_{3,1}$ would be a more accurate one. This can be seen from (117), where $\tilde{\gamma}_{3,1}$ is evaluated as $P \rightarrow \infty$, in which case $\tilde{\alpha}_{3,1} = \alpha_{3,1}$ (from (119)). The above example emphasizes the fact that $\tilde{\gamma}_{3,1}$, and thus $\tilde{\alpha}_{3,1}$, is a reliable estimate only if enough samples are averaged, and only then it should be used for digitized residue correction. As in a practical system only a finite number of samples is averaged, it is of interest to study the relationship between the number of samples used to obtain $\tilde{\gamma}_{1,1}$, $\tilde{\gamma}_{3,1}$ and $\tilde{\eta}_{2,1}$ and the accuracy of $\alpha_{1,1}$ and $\alpha_{3,1}$ estimates.

The *convergence time* of the estimation corresponds to the minimum number of samples that need to be averaged by the HDC_q block so that the errors due to $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ do not limit the desired accuracy of $\tilde{\alpha}_{1,q}$ and $\tilde{\alpha}_{3,q}$. In order to analyze how the properties of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ change with the number of averaged samples, and, thus, how the accuracy of $\alpha_{1,q}$ and $\alpha_{3,q}$ estimates changes, it is convenient to view $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ as random variables at each sample instant n . In what follows, the relationship between the convergence time of the estimation and the statistical properties of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ is analyzed, as well as how these properties of

$\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ affect $\alpha_{1,q}$ and $\alpha_{3,q}$ estimates.

A. Convergence time Analysis

From estimation theory [53], it is desirable for an estimator to have its mean converge to the parameter that is being estimated, and its variance go to zero, as the number of samples under consideration increases. The mean and the variance of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ can change with each sample n , and it is of interest to evaluate how they change with the number of averaged samples, P . For example, if neither their mean nor variance are functions of P , averaging more than one sample in (113) – (115) would not lead to better behaved estimators. Intuitively, however, it should be expected that at least one of these properties changes as a function of P only, as (113) – (115) converge to (116) – (118), respectively, for $P \rightarrow \infty$.

The convergence time analysis is done for the system in Figure 23. In order to simplify the statistical analysis of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$, certain assumptions are made regarding the pipelined ADC. The first one is that $|\alpha_{1,q}|$ and $|\alpha_{3,q}|$ are no bigger than 0.2 and $0.6V^2$, respectively. These values are about two times larger than what has been typically observed in practice. Moreover, any larger value of $\alpha_{1,q}$ would have a serious impact on the resolution of the pipelined ADC, even with the HDC algorithm correction. The second assumption is that the leakages of $e_{ADC,j}[n]$ and $d_j[n]$ (for $6 > j > q$) into *Stage* q can be ignored, as they do not significantly change the convergence time. This is true for a system that has a well-behaved random number generator that generates the sequence $c_q[n]$ ensuring that in each set of 16 samples, all the combina-

tions of the sequences in Table 2 occur once, and that the order in which they occur changes with each set of 16. Additionally, it is assumed that there is enough noise in the system so that $e_{ADC,6}[n]$ can be modeled as a uniformly distributed, white sequence, uncorrelated from the sequences used for estimation in (113) – (115).

From Figure 23, (102), (106), (110) and the above assumptions, for the purpose of the derivations that follow, the digitized residue in (113) – (115) can be well approximated as

$$r_q[n] = (1 + \alpha_{1,q})v_q(nTs) + \alpha_{3,q}v_q^3(nTs) + 4^{q-6}e_{ADC,6}[n], \quad (121)$$

where $v_q(nTs)$ corresponds to (108) for $q = 1$. From (114), the expected value of $\tilde{\gamma}_{3,q}$, is given by

$$E[\tilde{\gamma}_{3,q}] = \frac{1}{6A^3} \frac{1}{P} \sum_{n=m}^{m+P-1} E[(r_q[n] + c_q[n])c_{3,q}[n]]. \quad (122)$$

Substituting (121) into (122), and using the assumptions above with the properties of the sequences in Table 2, it follows that

$$E[\tilde{\gamma}_{3,q}] = \alpha_{3,q}. \quad (123)$$

Thus, the expected value of $\tilde{\gamma}_{3,q}$ is constant across all samples n , and as such it has no effect on the convergence time of the estimation. Applying the same reasoning as above to (113), (i.e. substituting (121) into (113), and using the stated assumptions together with the properties of the sequences in Table 2), it follows that

$$E[\tilde{\gamma}_{1,q}] = \alpha_{1,q} + \alpha_{3,q} \left(10A^2 + \frac{3}{P} \sum_{n=m}^{m+P-1} E[e_{ADC,q}^2[n]] \right). \quad (124)$$

Unlike $E[\tilde{\gamma}_{3,q}]$, the expected value of $\tilde{\gamma}_{1,q}$ is constant across all samples only for a stationary $e_{ADC,q}[n]$ signal. In general, the value of $E[\tilde{\gamma}_{1,q}]$ can change due to the

$$\frac{1}{P} \sum_{n=R_k}^{R_k+P-1} E[e_{ADC,q}^2[n]] \text{ term, which is a necessary component in the estimation of } \alpha_{1,q}.$$

Changing the number of averaged samples, however, does not result in a more accurate expected value of $\tilde{\gamma}_{1,q}$ estimator. For example, if the distribution of the input signal remains the same across all samples, the term $\frac{1}{P} \sum_{n=R_k}^{R_k+P-1} E[e_{ADC,q}^2[n]]$ does not

change as a function of P . Therefore, $E[\tilde{\gamma}_{1,q}]$ does not determine the convergence time of the estimation. Using the same procedure as above to evaluate the mean of $\tilde{\eta}_{2,q}$, it follows that

$$E[\tilde{\eta}_{2,q}] = -(1 + \alpha_{1,q})^2 \frac{1}{P} \sum_{n=m}^{m+P-1} E[e_{ADC,q}^2[n]] + Bias. \quad (125)$$

where *Bias* is bias given by

$$Bias \approx \frac{1}{P} \sum_{n=m}^{m+P-1} -2\alpha_{3,k} (1 + \alpha_{1,k}) E[e_{ADCk}^4[n]] - 60\alpha_{3,k} (1 + \alpha_{1,k}) E[e_{ADCk}^2[n]] A^2 - 150\alpha_{3,k}^2 E[e_{ADCk}^4[n]] A^2 - 1904A^4 \alpha_{3,k}^2 E[e_{ADCk}^2[n]].$$

As in the case of $E[\tilde{\gamma}_{1,q}]$, the expected value of $\tilde{\eta}_{2,q}$ varies, in general, with each sample n . Following the reasoning presented for $\tilde{\gamma}_{1,q}$ estimator, the term

$$-(1 + \alpha_{1,q})^2 \frac{1}{P} \sum_{n=m}^{m+P-1} E[e_{ADC,q}^2[n]] \text{ is a necessary component in the estimation of } \alpha_{1,q},$$

and its accuracy does not improve as P increases. Furthermore, increasing the number

of averaged samples does not, in general, change the bias in (125) (if the distribution of the input signal is unchanged, the bias of (125) remains constant). Therefore, $E[\tilde{\eta}_{2,q}]$ does not determine the convergence time of the estimation.

Based on the results above, and the intuitive argument stated in the beginning of this section, it is expected that the convergence time of the simulation depends on errors introduced due to variance of the estimators $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ or $\tilde{\eta}_{2,q}$. Furthermore, as the values of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ or $\tilde{\eta}_{2,q}$ are obtained by averaging the same number of samples, the convergence time is determined by the largest error that propagates to $\tilde{\alpha}_{1,q}$ and/or $\tilde{\alpha}_{3,q}$ due to one of these estimators.

The variance of $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ at each sample n can be computed in the standard way, i.e. by evaluating the second moment of the estimator and subtracting from it the squared value of its mean. As the properties of $c_q[n]$ are pre-determined by design and do not change with n , and the first two moments of $e_{ADC,6}[n]$ are assumed to be unchanging across all n , from (108), (113) – (115) and (121) it follows that $e_{ADC,q}[n]$ is the only term which properties are not pre-determined and which can affect the variance calculation. Furthermore, as can be verified from the derivations in the appendix, the convergence time of the simulation is determined by variance evaluated for the worst-case scenario when $|e_{ADC,q}[n]| = e_{ADC,max}$, where $e_{ADC,max}$ is the maximum value of the quantization error in the calibration stage. Nominally, $e_{ADC,max} = \Delta/2$.

From the derivations shown in the appendix, the variance of each of the esti-

mators $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ can be well-bounded by

$$\text{Var}[\tilde{\gamma}_{1,q}] \leq \frac{(1 + \alpha_{1,q})^2 e_{ADC \max}^2 \frac{6}{16} \left(\frac{2}{3} + B^2 \right)}{A^2 P} \quad (126)$$

$$\text{Var}[\tilde{\gamma}_{3,q}] \leq \frac{(1 + \alpha_{1,q})^2 e_{ADC \max}^2 \frac{6}{16} \left(\frac{2}{3} + B^2 \right)}{36 A^6 P} \quad (127)$$

$$\text{Var}[\tilde{\eta}_{2,q}] \approx \frac{1}{A^4 P^2} \sum_{n=m}^{m+P-1} \left(E \left[\left[\begin{array}{l} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^4 \\ + \alpha_{3,q} \left(e_{ADC,q}[n] + c_q[n] \right)^3 \end{array} \right] c_q^4[n] \right] - \left(E \left[\left[\begin{array}{l} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^2 \\ + \alpha_{3,q} \left(e_{ADC,q}[n] + c_q[n] \right)^3 \end{array} \right] c_q^2[n] \right] \right)^2 \right) \leq \frac{7\Delta^4}{2P}. \quad (128)$$

where A denotes the amplitude of the calibration sequences, and B is the magnitude of the sequences $c_a[n]$ and $c_b[n]$ in (111) and (112) (as defined in Section II). From (127) – (128) and the value A (that is $\Delta/16$ and as such much smaller than 1), it follows that the variance of $\tilde{\gamma}_{3,q}$ is larger than the variance in (126), while the variance in (128) is insignificant comparing to (127) and (126) (and as such it does not affect the convergence time).

From (119) it follows that

$$\text{Var}[\tilde{\alpha}_{3,q}] = E[\tilde{\alpha}_{3,q} - E[\tilde{\alpha}_{3,q}]] = E[\tilde{\gamma}_{3,q} - E[\tilde{\gamma}_{3,q}]] = \text{Var}[\tilde{\gamma}_{3,q}]. \quad (129)$$

Thus, the error due to the variance of $\tilde{\gamma}_{3,q}$ directly propagates into $\tilde{\alpha}_{3,q}$ estimator and increasing the number of averaged samples P in (127) directly minimizes the maximum error due to the variance of $\alpha_{3,q}$ estimate. In order to analyze how the variance

of $\alpha_{1,q}$ changes due to the estimators $\tilde{\gamma}_{1,q}$, $\tilde{\gamma}_{3,q}$ and $\tilde{\eta}_{2,q}$ consider the following. As determined by the expression in (128), the variance of $\tilde{\eta}_{2,q}$ is negligible comparing to the variances of the other two estimators. Therefore, the error due to the variance of $\tilde{\eta}_{2,q}$ tends to be statistically far smaller than the errors in $\tilde{\gamma}_{1,q}$ and $\tilde{\gamma}_{3,q}$. For this reason, consider the case in which K samples have been averaged so that the variance of $\tilde{\eta}_{2,q}$ is negligible (from (128), for $\Delta = 0.25$ V, after averaging $\sim 2^{20}$ samples this condition is satisfied, while the variance of the other two estimators is still significant). In this case, from (120), the variance of $\alpha_{1,q}$ estimator can be approximated as

$$\begin{aligned} \text{Var}[\tilde{\alpha}_{1,q}] = E[\tilde{\alpha}_{1,q} - E[\tilde{\alpha}_{1,q}]] &\approx \text{Var}[\tilde{\gamma}_{1,q}] + \text{Var}[\tilde{\gamma}_{3,q}](3\tilde{\eta}_{2,q} - 10A^2)^2 + \\ &(3\tilde{\eta}_{2,q} - 10A^2)E[(\tilde{\gamma}_{1,q} - E[\tilde{\gamma}_{1,q}])(\tilde{\gamma}_{3,q} - E[\tilde{\gamma}_{3,q}])] \end{aligned} \quad (130)$$

As shown in the appendix, for the worst-case scenario, the term $E[(\tilde{\gamma}_{1,q} - E[\tilde{\gamma}_{1,q}])(\tilde{\gamma}_{3,q} - E[\tilde{\gamma}_{3,q}])]$ is about 2^{18} times smaller than the variance of $\tilde{\gamma}_{3,q}$, and the variance of $\tilde{\gamma}_{1,q}$. Furthermore, as follows from (126), $\text{Var}[\tilde{\gamma}_{1,q}]$ is about 960 times smaller than $\text{Var}[\tilde{\gamma}_{3,q}](3\tilde{\eta}_{2,q} - 10A^2)^2$, in the worst case scenario when $(3\tilde{\eta}_{2,q} - 10A^2) \approx \frac{1}{22}$. Therefore, the estimator $\tilde{\gamma}_{3,q}$ not only limits the accuracy of $\tilde{\alpha}_{3,q}$ but also of $\tilde{\alpha}_{1,q}$. Thus, the convergence time of the simulation is determined by $\tilde{\gamma}_{3,q}$ estimator, and depending on the desired accuracy of $\tilde{\alpha}_{3,q}$ and $\tilde{\alpha}_{1,q}$, it can be found using the variance expression in (127).

IV. SIMULATION RESULTS

In order to verify the bound of $\tilde{\gamma}_{3,q}$ given in (127), the system in Figure 23 is simulated during the calibration of *Stage 1* with calibration sequences such that $B = 1.5$ and a DC input such that $e_{ADC,q}[n] = e_{ADCmax} = 0.625\Delta$, for $\Delta = 0.25V$. Furthermore, the residue amplifier coefficients $\alpha_{1,1}$ and $\alpha_{3,1}$ are equal to -0.085 and $-0.3V^2$, respectively. All the stages following the calibration stage have residue amplifiers with gain-errors comparable to $\alpha_{1,1}$. These gain errors have been digitally estimated with accuracy typically seen after averaging 2^{32} samples, and the estimates are used to correct the digitized residue prior to entering *Stage 1*.

The solid line in Figure 24 corresponds to the theoretical standard deviation of $\tilde{\gamma}_{3,q}$ that follows from (127) for $e_{ADCmax} = 0.625\Delta$, plotted as a function of the number of averaged samples P . The dotted line corresponds to the standard deviation of $\tilde{\gamma}_{3,q}$ that is estimated experimentally, using 48 Monte Carlo runs which are not shown. There is an excellent match between the upper bound given by (127) and the upper bound deduced from the simulation. Additionally, these results do not contradict the claim that for well-designed random number generators, the leakage of the signals $e_{ADC,st}[n]$ and $d_{st}[n]$ into the calibration stage does not significantly affect the convergence time of the simulation.

V. ACKNOWLEDGEMENTS

Chapter 3, in part is currently being prepared for submission for publication of the material. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis of this paper.

VI. FIGURES

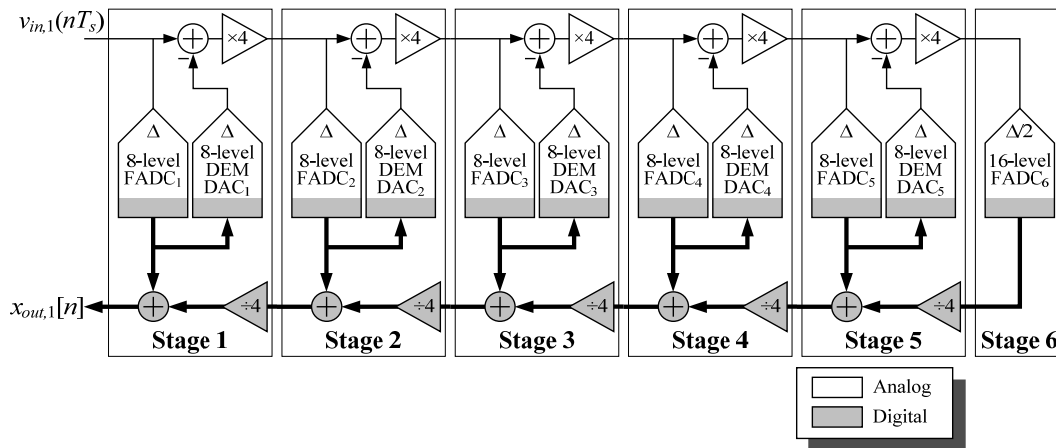
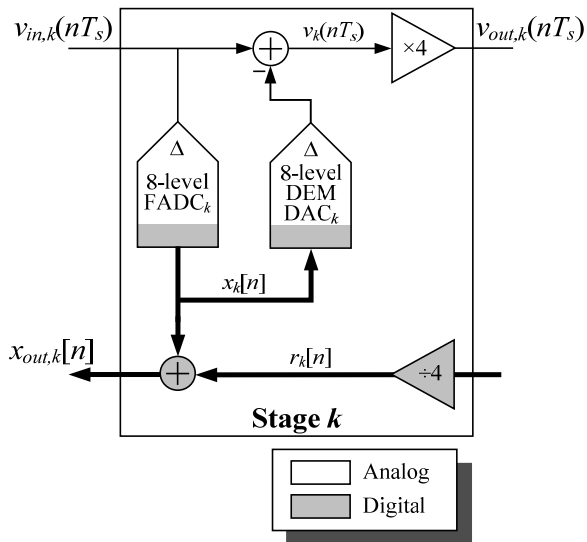


Figure 21 Example pipelined ADC.

Figure 22 The k th pipelined ADC stage for $k = 1, \dots, 5$ with labels that indicate variable names used throughout the paper.

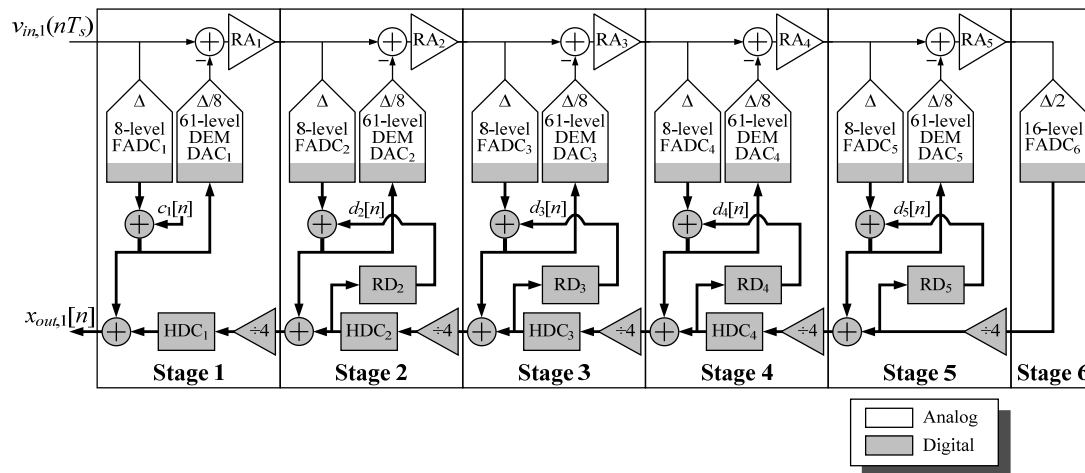


Figure 23 Pipelined ADC configuration during estimation of $\alpha_{1,1}$ and $\alpha_{3,1}$ with HDC in the first stage and RD applied to the remaining stages

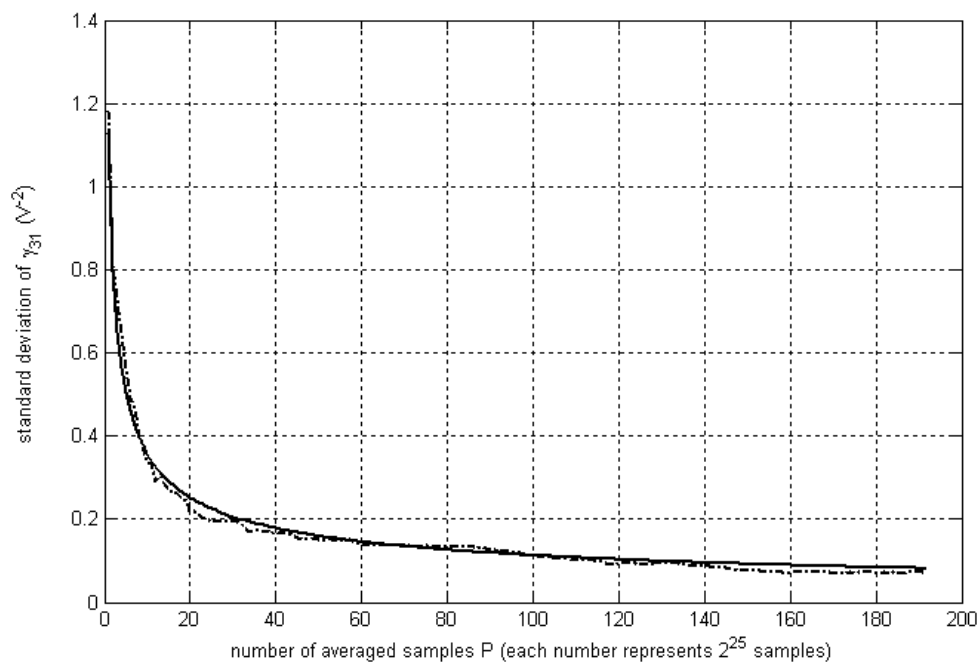


Figure 24 A comparison plot of calculated standard deviation of $\tilde{y}_{3,1}$ estimate (bold) vs. simulated standard deviation of $\tilde{y}_{3,1}$, plotted as a function of the number of averaged samples P .

VII. TABLES

Table 2 All possible sets of values that $t_{1,j}[n]/A$, $t_{2,j}[n]/A$, $t_{3,j}[n]/A$, $t_{4,j}[n]/A$ can take on along with the corresponding values of $c_j[n]/A$, $c_{1,j}[n]$, $c_{3,j}[n]$, and $c_{1,j}[n] \pm c_{3,j}[n]$.

$t_{1,j}[n]/A$	$t_{2,j}[n]/A$	$t_{3,j}[n]/A$	$t_{4,j}[n]/A$	$c_j[n]/A$	$c_{1,j}[n]$	$c_{3,j}[n]$	$c_{1,j}[n] + c_{3,j}[n]$	$c_{1,j}[n] - c_{3,j}[n]$
-1	-1	-1	-1	-4	1	1	2	0
-1	-1	-1	1	-2	1/2	-1/2	0	1
-1	-1	1	-1	-2	1/2	-1/2	0	1
-1	-1	1	1	0	$c_a[n]$	$c_a[n]$	$2c_a[n]$	0
-1	1	-1	-1	-2	1/2	-1/2	0	1
-1	1	-1	1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
-1	1	1	-1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
-1	1	1	1	2	-1/2	1/2	0	-1
1	-1	-1	-1	-2	1/2	-1/2	0	1
1	-1	-1	1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
1	-1	1	-1	0	$c_b[n]$	$-c_b[n]$	0	$2c_b[n]$
1	-1	1	1	2	-1/2	1/2	0	-1
1	1	-1	-1	0	$c_a[n]$	$c_a[n]$	$2c_a[n]$	0
1	1	-1	1	2	-1/2	1/2	0	-1
1	1	1	-1	2	-1/2	1/2	0	-1
1	1	1	1	4	-1	-1	-2	0

VIII. APPENDIX

The derivations for the equations (127) – (128) are shown in this section.

Using the information in (114) and (121), the upper bound on the variance of $\tilde{\gamma}_{3,q}$ can be computed as follows. First, the second moment of the estimator in (114) needs to be determined,

$$\begin{aligned}
E[\tilde{\gamma}_{3,q}^2] &= E\left[\left(\frac{1}{6A^3} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{3,q}[n]\right)^2\right] \\
&= \frac{1}{36A^6 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E\left[(r_q[n] + c_q[n])(r_q[s] + c_q[s]) c_{3,q}[n] c_{3,q}[s]\right] \\
E[\tilde{\gamma}_{3,q}^2] &= \frac{1}{36A^6 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E\left[\begin{aligned} &\left(\begin{aligned} &\left((1 + \alpha_{1,q})(-e_{ADC,q}[n]) - \alpha_{1,q} c_q[n] \right. \\ &\left. - \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 + 4^{q-6} e_{ADC,6}[n] \right) c_{3,q}[n] \\ &\left((1 + \alpha_{1,q})(-e_{ADC,q}[s]) - \alpha_{1,q} c_q[s] \right. \\ &\left. - \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 + 4^{q-6} e_{ADC,6}[s] \right) c_{3,q}[s] \end{aligned} \right) \\ &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[n] \right. \\ &\left. + \alpha_{1,q} c_q[n] - 4^{q-6} e_{ADC,6}[n] \right) \cdot c_{3,q}[n] c_{3,q}[s] \cdot \\ &\left((1 + \alpha_{1,q}) e_{ADC,q}[s] + \alpha_{1,q} c_q[s] - 4^{q-6} e_{ADC,6}[s] \right) + \\ &2\alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \cdot c_{3,q}[n] c_{3,q}[s] \cdot \\ &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[s] \right. \\ &\left. + \alpha_{1,q} c_q[s] - 4^{q-6} e_{ADC,6}[s] \right) + \alpha_{3,q}^2 (e_{ADC,q}[n] + c_q[n])^3 \cdot \\ &(e_{ADC,q}[s] + c_q[s])^3 c_{3,q}[n] c_{3,q}[s] \end{aligned} \right) \end{aligned} \right] \\
&= \frac{1}{36A^6 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E\left[\begin{aligned} &\left(\begin{aligned} &\left((1 + \alpha_{1,q})(-e_{ADC,q}[n]) - \alpha_{1,q} c_q[n] \right. \\ &\left. - \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 + 4^{q-6} e_{ADC,6}[n] \right) c_{3,q}[n] \\ &\left((1 + \alpha_{1,q})(-e_{ADC,q}[s]) - \alpha_{1,q} c_q[s] \right. \\ &\left. - \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 + 4^{q-6} e_{ADC,6}[s] \right) c_{3,q}[s] \end{aligned} \right) \\ &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[n] \right. \\ &\left. + \alpha_{1,q} c_q[n] - 4^{q-6} e_{ADC,6}[n] \right) \cdot c_{3,q}[n] c_{3,q}[s] \cdot \\ &\left((1 + \alpha_{1,q}) e_{ADC,q}[s] + \alpha_{1,q} c_q[s] - 4^{q-6} e_{ADC,6}[s] \right) + \\ &2\alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \cdot c_{3,q}[n] c_{3,q}[s] \cdot \\ &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[s] \right. \\ &\left. + \alpha_{1,q} c_q[s] - 4^{q-6} e_{ADC,6}[s] \right) + \alpha_{3,q}^2 (e_{ADC,q}[n] + c_q[n])^3 \cdot \\ &(e_{ADC,q}[s] + c_q[s])^3 c_{3,q}[n] c_{3,q}[s] \end{aligned} \right) \end{aligned} \right]
\end{aligned}$$

From the properties of the calibration sequences in (107), (111) and (112), and the fact that $e_{ADC,6}[n]$ is independent from $e_{ADC,q}[s]$ and $c_{3,q}[s]$ for $n \neq s$, the expected value of the first and the second term above can be non-zero only when $s = q$. Thus,

$$\begin{aligned}
& E \left[\left(\begin{array}{l} (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ \alpha_{1,q} c_q[n] - 4^{q-6} e_{ADC,6}[n] \end{array} \right)^2 c_{3,q}^2[n] \right] + \\
E[\tilde{\gamma}_{3,q}^2] &= \frac{1}{A^6 P^2} \sum_{n=m}^{m+P-1} \frac{\alpha_{3,q}}{0.5} E \left[\left(e_{ADC,q}[n] + c_q[n] \right)^3 \begin{array}{l} (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ \alpha_{1,q} c_q[n] - \\ 4^{q-6} e_{ADC,6}[n] \end{array} c_{3,q}^2[n] \right] \quad (131) \\
& + \left(\sum_{s=m}^{m+P-1} E \left[\begin{array}{l} \alpha_{3,q}^2 (e_{ADC,q}[n] + c_q[n])^3 \cdot \\ (e_{ADC,q}[s] + c_q[s])^3 c_{3,q}[n] c_{3,q}[s] \end{array} \right] \right)
\end{aligned}$$

The square of the expected value of $\tilde{\gamma}_{3,q}$ can be found as follows

$$\begin{aligned}
E[\tilde{\gamma}_{3,q}]^2 &= \left(E \left[\frac{1}{6A^3 P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{3,q}[n] \right] \right)^2 \\
&= \frac{1}{36A^6 P} \left(\sum_{n=m}^{m+P-1} E \left[(r_q[n] + c_q[n]) c_{3,q}[n] \right] \right) \left(\sum_{n=s}^{m+P-1} E \left[(r_q[s] + c_q[s]) c_{3,q}[n] \right] \right) \\
&= \frac{1}{36A^6 P^2} \left(\sum_{n=m}^{m+P-1} E \left[\begin{array}{l} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] - 4^{q-6} e_{ADC,6}[n] \right) \\ + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 + \alpha_{1,q} c_q[n] \end{array} c_{3,q}[n] \right] \right) \\
&\quad \left(\sum_{n=s}^{m+P-1} E \left[(r_q[s] + c_q[s]) c_{3,q}[n] \right] \right) \\
&= \frac{1}{36A^6 P^2} \left(\sum_{n=m}^{m+P-1} E \left[\alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 c_{3,q}[n] \right] \right) \\
&\quad \left(\sum_{n=s}^{m+P-1} E \left[(r_q[s] + c_q[s]) c_{1,q}[s] \right] \right)
\end{aligned}$$

Subtracting the above value from (131) and simplifying the resulting expression using the properties of $e_{ADC,q}[n]$, assumptions in Section III regarding $e_{ADC,6}[n]$, and the sequences $c_{3,q}[n]$ and $c_q[n]$ defined in (107) and (112),

$$\begin{aligned}
Var[\tilde{y}_{3,q}] = & \frac{1}{A^6 P^2} \sum_{n=m}^{m+P-1} \alpha_{1,q} E \left[e_{ADC,q}^2[n] c_q^2[n] c_{3,q}^2[n] \right] + 2\alpha_{3,q} \alpha_{1,q} E \left[c_q^4[n] c_{3,q}^2[n] \right] \\
& - 6\alpha_{3,q} (1 + \alpha_{1,q}) E \left[e_{ADC,q}^2[n] c_q^2[n] c_{3,q}^2[n] \right] - \\
& 2 \cdot 4^{q-6} E \left[e_{ADC,6}[n] \left(3\alpha_{3,q} e_{ADC,q}[n] c_q^2[n] + \alpha_{3,q} c_q^3[n] + \alpha_{3,q} e_{ADC,q}^3[n] \right) c_{3,q}^2[n] \right] + \\
& \alpha_{3,q}^2 E \left[e_{ADC,q}^6[n] c_{3,q}^2[n] \right] + 15\alpha_{3,q}^2 E \left[e_{ADC,q}^4[n] c_q^2[n] c_{3,q}^2[n] \right] \\
& + 15\alpha_{3,q}^2 E \left[e_{ADC,q}^2[n] c_q^4[n] c_{3,q}^2[n] \right] + \alpha_{3,q}^2 E \left[c_q^6[n] c_{3,q}^2[n] \right]
\end{aligned} \tag{132}$$

It can be verified by inspection that each term is at its maximum value if $|e_{ADC,q}[n]| = e_{ADCmax}$. Therefore, maximum variance is achieved when $|e_{ADC,q}[n]| = e_{ADCmax}$. Furthermore, under this condition, the most dominant among the above terms is the first one,

$$E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^2 c_{3,q}^2[n] + \frac{e_{ADC,6}^2[n]}{4^{12-2q}} c_{3,q}^2[n] \right] \Bigg|_{e_{ADC,q}[n]=e_{ADC,max}} \tag{133}$$

The second largest term is $2\alpha_{3,q} (1 + \alpha_{1,q}) E \left[e_{ADC,q}^4[n] c_{3,q}^2[n] \right]$, which is

$\frac{1}{2\alpha_{3,q} (1 + \alpha_{1,q}) e_{ADC,max}^2}$ times smaller than the most dominant one. For the pipelined

ADC in Figure 23 implemented in a modern CMOS technology, this means that the second largest term is ~ 30 times smaller than (133). As such, it affects the maximum variance calculation by at most 2%, and thus, it can be considered negligible. Furthermore, since

$$E[c_{1,q}^2[n]] = \frac{6}{16} \left(\frac{2}{3} + B^2 \right) \quad (134)$$

the term

$$\left| 2 \cdot 4^{q-6} E \left[e_{ADC,6}[n] \cdot \alpha_{3,q} \left(e_{ADC,q}[n] + c_q[n] \right)^3 c_{3,q}^2[n] \right] \right| \leq \frac{e_{ADCmax}^3 \cdot \Delta_6}{16} \alpha_{3,q} \frac{6}{16} \left(\frac{2}{3} + B^2 \right), \text{ for}$$

$1 \geq B \geq 1.5$ (reasonable values of B that are needed in the system, refer to [52] for more information) is about 30 times smaller than $2\alpha_{3,q} (1 + \alpha_{1,q}) E[e_{ADC,q}^4[n] c_{3,q}^2[n]]$ when $|e_{ADC,q}[n]| = e_{ADCmax}$. In practice, $e_{ADC,6}[n]$ is significantly dithered with the noise in the system, and it is not expected that its correlation is ever going to be as high as the set bound. Using the properties of the sequences in Table 2, it can be verified that all the other terms in (132) are significantly smaller than the first term when $|e_{ADC,q}[n]| = e_{ADCmax}$. Thus, the upper bound on the variance can be well approximated by the first term in (132). From the assumptions stated in Section III, $e_{ADC,6}[n]$ is uncorrelated with $c_{1,q}[n]c_q[n]$ and also from $c_{3,q}[n]^2$ (which is just a scaled version of $c_{1,q}[n]c_q[n]$ when $c_q[n] \neq 0$), and the properties of the sequences in Table 2, it follows that the variance expression can be bounded as

$$\begin{aligned} Var[\tilde{y}_{3,q}] &\leq \frac{1}{36A^6P^2} \sum_{n=m}^{m+P-1} E \left[\begin{array}{l} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^2 c_{3,q}^2[n] \\ + 4^{2q-12} e_{ADC,6}[n] c_{3,q}^2[n] \\ - \frac{2}{4^{6-q}} \cdot (1 + \alpha_{1,q}) e_{ADC,q}[n] e_{ADC,6}[n] c_{3,q}^2[n] \end{array} \right] \Big|_{e_{ADC,q}[n]=e_{ADCmax}} \\ &\leq \frac{1}{36A^6P^2} \sum_{n=m}^{m+P-1} E \left[\begin{array}{l} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^2 c_{3,q}^2[n] \\ - \frac{2}{4^{6-q}} \cdot (1 + \alpha_{1,q}) E[e_{ADC,q}[n] e_{ADC,6}[n] c_{3,q}^2[n]] \end{array} \right] \Big|_{e_{ADC,q}[n]=e_{ADCmax}} \end{aligned}$$

The correlation $\left| E\left[e_{ADC,q}[n]e_{ADC,6}[n]c_{3,q}^2[n] \right] \right|$ can be bounded by $\frac{e_{ADC\max}^2}{2} E\left[c_{3,q}^2[n] \right]$ so that the term $2 \cdot 4^{q-6} (1 + \alpha_{1,q}) E\left[e_{ADC,q}[n]e_{ADC,6}[n]c_{3,q}^2[n] \right]$ is at least 16 times smaller than the first term when $e_{ADC,q}[n] = e_{ADC\max}$ (in the worst case scenario for $q = 4$). In practice, however, the correlation due to $E\left[e_{ADC,q}[n]e_{ADC,6}[n]c_{3,q}^2[n] \right]$ is expected to always be smaller than the bound $\frac{e_{ADC\max}^2}{2} E\left[c_{3,q}^2[n] \right]$, as there are multiple noise sources prior to the *Stage 6* (which signals are not correlated with $e_{ADC,q}[n]$) that help in minimizing the correlation of $e_{ADC,q}[n]$ with $e_{ADC,6}[n]$. Additionally, as $E\left[\alpha_{1,q}^2 c_q^2[n]c_{3,q}^2[n] \right]$ is at least 150 times smaller than $(1 + \alpha_{1,q})^2 e_{ADC\max}^2 E\left[c_{3,q}^2[n] \right]$, from (134) it follows that the variance of $\tilde{\gamma}_{3,q}$ can be well-bounded by

$$\text{Var}\left[\tilde{\gamma}_{3,q} \right] \leq \frac{(1 + \alpha_{1,q})^2 e_{ADC\max}^2 \left(\frac{6}{16} \left(\frac{2}{3} + B^2 \right) \right)}{36A^6P}.$$

The variance of $\tilde{\gamma}_{1,q}$ can be computed in a similar way as above. Substituting (121) into (113),

$$\begin{aligned} E\left[\tilde{\gamma}_{1,q}^2 \right] &= E\left[\left(\frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])c_{1,q}[n] \right)^2 \right] \\ &= \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E\left[(r_q[n] + c_q[n])(r_q[s] + c_q[s])c_{1,q}[n]c_{1,q}[s] \right], \end{aligned}$$

which can be expanded as

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E \left[\begin{array}{l} \left(\begin{array}{l} (1+\alpha_{1,q})(-e_{ADC,q}[n]) - \alpha_{1,q}c_q[n] \\ -\alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 + 4^{q-6}e_{ADC,6}[n] \end{array} \right) c_{1,q}[n] \\ \left(\begin{array}{l} (1+\alpha_{1,q})(-e_{ADC,q}[s]) - \alpha_{1,q}c_q[s] \\ -\alpha_{3,q}(e_{ADC,q}[s] + c_q[s])^3 + 4^{q-6}e_{ADC,6}[s] \end{array} \right) c_{1,q}[s] \end{array} \right],$$

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E \left[\begin{array}{l} \left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] + \\ \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \end{array} \right) c_{1,q}[n] \cdot \\ \left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{1,q}c_q[s] + \\ \alpha_{3,q}(e_{ADC,q}[s] + c_q[s])^3 \end{array} \right) c_{1,q}[s] - \\ \frac{2}{4^{6-q}}e_{ADC,6}[n] \left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{1,q}c_q[s] \\ + \alpha_{3,q}(e_{ADC,q}[s] + c_q[s])^3 \end{array} \right) \cdot \\ c_{1,q}[n]c_{1,q}[s] + 4^{2q-12}e_{ADC,6}[n]e_{ADC,6}[s]c_{1,q}[n]c_{1,q}[s] \end{array} \right]$$

From the properties of the calibration sequences in (107), (111) and (112), and the fact that $e_{ADC,6}[n]$ is independent from $e_{ADC,q}[s]$ and $c_{1,q}[s]$ for $n \neq s$, the expected value of the third and fourth terms above can be non-zero only when $s = q$. Thus,

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \left(\begin{array}{l} \left(\begin{array}{l} \sum_{s=m}^{m+P-1} E \left[\left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] \\ + \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \end{array} \right) c_{1,q}[n] \cdot \right. \\ \left. \left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{1,q}c_q[s] \\ + \alpha_{3,q}(e_{ADC,q}[s] + c_q[s])^3 \end{array} \right) c_{1,q}[s] \end{array} \right) \\ - \frac{2}{4^{6-q}} \cdot E \left[e_{ADC,6}[n] \left(\begin{array}{l} (1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] \\ + \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \end{array} \right) c_{1,q}^2[n] \right] \end{array} \right) + E \left[\frac{e_{ADC,6}^2[n]}{4^{12-2q}} \cdot c_{1,q}^2[n] \right]$$

From the properties of the system, $e_{ADC,6}[n]$ can be correlated with $c_{1,q}[n]$ only during the samples that $c_q[n] \neq 0$. Furthermore, from the assumptions stated in Section III,

$e_{ADC,6}[n]$ is uncorrelated with $c_{1,q}[n]^2$ (as $e_{ADC,6}[n]$ is uncorrelated with $c_q[n]c_{1,q}[n]$, and $c_{1,q}[n]^2$ is just a scaled version of this sequence when $c_q[n] \neq 0$). Using this information together with the variance of $e_{ADC,6}[n]$ given by

$$E[e_{ADC,6}^2[n]] = \left(\frac{\Delta_6^2}{12} \right), \quad (135)$$

where Δ_6 is the step size of FADC₆, from (134) and (135), it follows that

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \left(\begin{aligned} & \sum_{s=m}^{m+P-1} E \left[\begin{aligned} & \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] + \right. \\ & \left. \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \right) c_{1,q}[n] \end{aligned} \right) \\ & \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{1,q}c_q[s] + \right. \\ & \left. \alpha_{3,q}(e_{ADC,q}[s] + c_q[s])^3 \right) c_{1,q}[s] \end{aligned} \right) \end{aligned} \right] + \frac{\Delta_6^2 \left(\frac{2}{3} + B^2 \right)}{4^{12-2q}} \\ & - \frac{2}{4^{6-q}} E \left[e_{ADC,6}[n] \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] \right) \\ & + \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \end{aligned} \right) c_{1,q}^2[n] \right] \end{aligned} \right)$$

Expanding the first term and using the properties of $c_q[n]$, $c_{1,q}[n]$, $c_{3,q}[n]$ and $e_{ADC,q}[n]$,

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \cdot \left(\begin{aligned} & \frac{\Delta_6^2 \left(\frac{2}{3} + B^2 \right)}{12 \cdot 4^{12-2q}} + E \left[\begin{aligned} & \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{3,q}c_q^3[n] + 3\alpha_{3,q}e_{ADC,q}^2[n]c_q[n] + \right. \\ & \left. \alpha_{3,q}(e_{ADC,q}^3[n] + 3e_{ADC,q}[n]c_q^2[n]) + \alpha_{1,q}c_q[n] \end{aligned} \right) c_{1,q}[n] \end{aligned} \right] \\ & \sum_{n=m}^{m+P-1} \left(\begin{aligned} & \sum_{\substack{s=m \\ s \neq n}}^{m+P-1} E \left[\begin{aligned} & \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{3,q}c_q^3[s] + 3\alpha_{3,q}e_{ADC,q}^2[s]c_q[s] + \right. \\ & \left. \alpha_{3,q}(e_{ADC,q}^3[s] + 3e_{ADC,q}[s]c_q^2[s]) + \alpha_{1,q}c_q[s] \end{aligned} \right) c_{1,q}[s] \end{aligned} \right) \\ & E \left[\begin{aligned} & \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[s] + \alpha_{3,q}c_q^3[s] + 3\alpha_{3,q}e_{ADC,q}^2[s]c_q[s] + \right. \\ & \left. \alpha_{3,q}(e_{ADC,q}^3[s] + 3e_{ADC,q}[s]c_q^2[s]) + \alpha_{1,q}c_q[s] \end{aligned} \right) c_{1,q}^2[s] \end{aligned} \right] - \frac{2}{4^{6-q}} \\ & E \left[e_{ADC,6}[n] \left(\begin{aligned} & \left((1+\alpha_{1,q})e_{ADC,q}[n] + \alpha_{1,q}c_q[n] + \alpha_{3,q}(e_{ADC,q}[n] + c_q[n])^3 \right) c_{1,q}^2[n] \end{aligned} \right) \end{aligned} \right) \end{aligned} \right)$$

and using the properties of the sequences given in Table 2, the equation above can be further simplified to

$$E[\tilde{\gamma}_{1,q}^2] = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \left(E \left[\begin{aligned} & \left[\begin{aligned} & \left(\alpha_{1,q} c_q[n] c_{1,q}[n] + \alpha_{3,q} c_q^3[n] c_{1,q}[n] + \right) \\ & \left(3\alpha_{3,q} e_{ADC,q}^2[n] c_q[n] c_{1,q}[n] \right) \end{aligned} \right] \\ & \sum_{\substack{s=m \\ s \neq n}}^{m+P-1} E \left[\begin{aligned} & \left(\alpha_{1,q} c_q[s] c_{1,q}[s] + \alpha_{3,q} c_q^3[s] c_{1,q}[s] + \right) \\ & \left(3\alpha_{3,q} e_{ADC,q}^2[s] c_q[s] c_{1,q}[s] \right) \end{aligned} \right] + \\ & E \left[\begin{aligned} & \left(\alpha_{1,q} c_q[n] c_{1,q}[n] + \alpha_{3,q} c_q^3[n] c_{1,q}[n] + \right)^2 \\ & \left(3\alpha_{3,q} e_{ADC,q}^2[n] c_q[n] c_{1,q}[n] \right) \end{aligned} \right] + \frac{\Delta_6^2 \left(\frac{2}{3} + B^2 \right)}{12 \cdot 4^{12-2q}} \\ & + E \left[\begin{aligned} & \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \right. \\ & \left. \alpha_{3,q} (e_{ADC,q}^3[n] + 3e_{ADC,q}[n] c_q^2[n]) \right)^2 \\ & c_{1,q}^2[n] \end{aligned} \right] - 2 \cdot \\ & E \left[\begin{aligned} & 4^{q-6} e_{ADC,6}[n] \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \right) \\ & \left(\alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) c_{1,q}^2[n] \end{aligned} \right] \end{aligned} \right) \quad (136)$$

In order to find the variance of $\tilde{\gamma}_{1,q}$, the expected value squared of this estimator needs to be determined,

$$\begin{aligned} E[\tilde{\gamma}_{1,q}^2] &= \left(E \left[\frac{1}{AP} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{1,q}[n] \right] \right)^2 \\ &= \frac{1}{A^2 P^2} \left(\sum_{n=m}^{m+P-1} E \left[(r_q[n] + c_q[n]) c_{1,q}[n] \right] \right)^2 \\ &= \frac{1}{A^2 P^2} \left(\sum_{n=m}^{m+P-1} E \left[\begin{aligned} & \left(\alpha_{1,q} c_q[n] c_{1,q}[n] + \alpha_{3,q} c_q^3[n] c_{1,q}[n] + \right) \\ & \left(3\alpha_{3,q} e_{ADC,q}^2[n] c_q[n] c_{1,q}[n] \right) \end{aligned} \right] \right. \\ & \quad \left. \left(\sum_{n=s}^{m+P-1} E \left[(r_q[s] + c_q[s]) c_{1,q}[s] \right] \right) \right) \end{aligned}$$

Using (136), the squared mean value of $\tilde{\gamma}_{1,q}$ obtained above, and the properties of

$c_q[n]$, the variance of $\tilde{\gamma}_{1,q}$ is given by

$$\begin{aligned}
& \text{Var}[\tilde{\gamma}_{1,q}] = \\
& = \frac{1}{A^2 P^2} \sum_{n=m}^{m+P-1} \left(E \left[\left(\begin{aligned} & \alpha_{1,q} c_q[n] c_{1,q}[n] + \\ & 3\alpha_{3,q} e_{ADC,q}^2[n] c_q[n] c_{1,q}[n] + \alpha_{3,q} c_q^3[n] c_{1,q}[n] \end{aligned} \right)^2 \right] - \right. \\
& \left. E \left[\left(\begin{aligned} & \alpha_{1,q} c_q[n] c_{1,q}[n] + \\ & 3\alpha_{3,q} e_{ADC,q}^2[n] c_q[n] c_{1,q}[n] + \alpha_{3,q} c_q^3[n] c_{1,q}[n] \end{aligned} \right) \right]^2 \right) + \\
& E \left[\left(\begin{aligned} & (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ & \alpha_{3,q} (e_{ADC,q}^3[n] + 3e_{ADC,q}[n] c_q^2[n]) \end{aligned} \right)^2 c_{1,q}^2[n] \right] + \frac{\Delta_6^2 \left(\frac{2}{3} + B^2 \right)}{12 \cdot 4^{12-2q}} \\
& - \frac{2}{4^{6-q}} E \left[e_{ADC,6}[n] \left(\begin{aligned} & (1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \\ & \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{aligned} \right) c_{1,q}^2[n] \right] \\
& \left. \begin{aligned} & \left(1.5\alpha_{1,q}^2 + \alpha_{3,q}^2 A^2 \left(22.5 \cdot E[e_{ADC,q}^4[n]] - 9 \left(E[e_{ADC,q}^2[n]] \right)^2 \right) + 48\alpha_{1,q} \cdot \right. \\ & \alpha_{3,q} A^4 + (144\alpha_{3,q}^2 A^2 + 9\alpha_{1,q} \alpha_{3,q}) E[e_{ADC,q}^2[n]] A^2 + 420 A^6 \alpha_{3,q}^2 \\ & + E \left[\left(\begin{aligned} & (1 + \alpha_{1,q})^2 e_{ADC,q}^2[n] + \\ & \alpha_{3,q}^2 (e_{ADC,q}^3[n] + 3e_{ADC,q}[n] c_q^2[n])^2 + \\ & 2\alpha_{3,q} (1 + \alpha_{1,q}) e_{ADC,q}[n] \cdot \\ & (e_{ADC,q}^3[n] + 3e_{ADC,q}[n] c_q^2[n]) \end{aligned} \right) c_{1,q}^2[n] \right] + \frac{\Delta_6^2 \left(\frac{2}{3} + B^2 \right)}{12 \cdot 4^{12-2q}} \\ & \left. - \frac{2}{4^{6-q}} E \left[e_{ADC,6}[n] \left(\begin{aligned} & (1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \\ & \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{aligned} \right) c_{1,q}^2[n] \right] \right) \end{aligned} \right) \quad (137)
\end{aligned}$$

For $|e_{ADC,q}[n]| = e_{ADC,max}$, the variance above is maximum, considering that the only

term that can reduce the variance in this case is $9A^2 \left(E[e_{ADC,q}^2[n]] \right)^2 \alpha_{3,q}^2$, which is

about $2 \cdot 10^4$ times smaller than the most dominant term, i.e.

$E\left[\left(1+\alpha_{1,q}\right)^2 e_{ADC,q}^2[n]c_{1,q}^2[n]\right]_{e_{ADC,q}[n]=e_{ADC,max}}$. Thus, the term $9A^2\left(E\left[e_{ADC,q}^2[n]\right]\right)^2\alpha_{3,q}^2$

can be neglected. The next largest term in (137) when $|e_{ADC,q}[n]|=e_{ADC,max}$ is

$2\left(1+\alpha_{1,q}\right)\alpha_{3,q}E\left[e_{ADC,q}^4[n]c_{1,q}^2[n]\right]$, which is $\frac{1}{2\alpha_{3,q}\left(1+\alpha_{1,q}\right)e_{ADC,max}^2}$ times smaller than

the most dominant one. For the pipelined ADC in Figure 23 implemented in a modern CMOS technology, this term is about 30 times smaller than

$E\left[\left(1+\alpha_{1,q}\right)^2 e_{ADC,q}^2[n]c_{1,q}^2[n]\right]_{e_{ADC,q}[n]=e_{ADC,max}}$. As such, $2\left(1+\alpha_{1,q}\right)\alpha_{3,q}E\left[e_{ADC,q}^4[n]c_{1,q}^2[n]\right]$ af-

fects the maximum variance by at most 2%, and thus can be considered negligible.

The next largest term is

$$2E\left[4^{q-6}e_{ADC,6}[n]\left(\left(1+\alpha_{1,q}\right)e_{ADC,q}[n]+\alpha_{1,q}c_q[n]+\alpha_{3,q}\left(e_{ADC,q}[n]+c_q[n]\right)^3\right)c_{1,q}^2[n]\right],$$

which magnitude is smaller than or equal to $\sim e_{ADC,max}^4$ in the worst case scenario ($q=4$). This calculation assumes that the output of *Stage 4* is entirely correlated with $e_{ADC,6}[n]$, and as such it is about 30 times smaller than the most dominant term. However, in a practical system, $e_{ADC,6}[n]$ is significantly dithered by noise (thermal, DAC noise, etc.), so that its correlation with

$$\left(\left(1+\alpha_{1,q}\right)e_{ADC,q}[n]+\alpha_{1,q}c_q[n]+\alpha_{3,q}\left(e_{ADC,q}[n]+c_q[n]\right)^3\right)$$

is expected to be far smaller than $e_{ADC,max}^4$. It can be shown that all the other terms in

(137) are also significantly smaller than $E\left[\left(1+\alpha_{1,q}\right)^2 e_{ADC,q}^2[n]c_{1,q}^2[n]\right]$ for

$|e_{ADC,q}[n]|=e_{ADC,max}$, so from the above discussion and the properties of $c_{1,q}[n]$ se-

quence, the variance of $\tilde{\gamma}_{1,q}$ can be well-bounded by

$$\text{Var}[\tilde{\gamma}_{1,q}] \leq \frac{(1 + \alpha_{1,q})^2 e_{ADC \max}^2 \frac{6}{16} \left(\frac{2}{3} + B^2 \right)}{A^2 P}.$$

Following the same procedure as above, the variance of $\tilde{\eta}_{2,q}$ can be found from (115) and (121),

$$\begin{aligned} E[\tilde{\eta}_{2,q}^2] &= \frac{1}{A^2} \frac{1}{P^2} E \left[\sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])^2 c_{1,q}[n] c_q[n] \sum_{q=m}^{m+P-1} (r_q[s] + c_q[s])^2 c_{1,q}[s] c_q[s] \right] \\ &= \frac{1}{A^4 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E \left[\begin{aligned} &\left(\begin{aligned} &-(1 + \alpha_{1,q}) e_{ADC,q}[n] - \alpha_{1,q} c_q[n] - \\ &\alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 + 4^{q-6} e_{ADC,6}[n] \end{aligned} \right)^2 c_q^2[n] \\ &\left(\begin{aligned} &-(1 + \alpha_{1,q}) e_{ADC,q}[s] - \alpha_{1,q} c_q[s] - \\ &\alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 + 4^{q-6} e_{ADC,6}[s] \end{aligned} \right)^2 c_q^2[s] \end{aligned} \right] \\ &= \frac{1}{A^4 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E \left[\begin{aligned} &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right)^2 + 4^{2q-12} e_{ADC,6}^2[n] \\ &+ \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{aligned} \right) c_q^2[n] \\ &\left(\begin{aligned} &-\frac{2e_{ADC,6}[n]}{4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \right. \\ &\left. \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \end{aligned} \right) c_q^2[n] \\ &\left(\begin{aligned} &\left((1 + \alpha_{1,q}) e_{ADC,q}[s] + \alpha_{1,q} c_q[s] \right)^2 + 4^{2q-12} e_{ADC,6}^2[s] \\ &+ \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 \end{aligned} \right) c_q^2[s] \\ &\left(\begin{aligned} &-\frac{2e_{ADC,6}[s]}{4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[s] + \right. \\ &\left. \alpha_{1,q} c_q[s] + \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 \right) \end{aligned} \right) c_q^2[s] \end{aligned} \right] \end{aligned}$$

Expanding this expression,

$$\begin{aligned}
E[\tilde{\eta}_{2,q}^2] = & \\
& E \left[\left(\begin{array}{l} (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{array} \right)^2 c_q^2[n] \right] + 2 \cdot \\
& E \left[\left(\begin{array}{l} (1 + \alpha_{1,q}) e_{ADC,q}[s] + \\ \alpha_{1,q} c_q[s] + \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 \end{array} \right)^2 c_q^2[s] \right] \\
& \frac{1}{A^4 P^2} \sum_{n=m}^{m+P-1} \sum_{s=m}^{m+P-1} E \left[\left(\begin{array}{l} (4^{q-6} e_{ADC,6}[n])^2 - \frac{2}{4^{6-q}} e_{ADC,6}[n] \cdot \\ (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{array} \right) c_q^2[n] \right] \cdot \\
& E \left[\left(\begin{array}{l} (1 + \alpha_{1,q}) e_{ADC,q}[s] + \\ \alpha_{1,q} c_q[s] + \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 \end{array} \right)^2 c_q^2[s] \right] \\
& + E \left[\left(\begin{array}{l} (4^{q-6} e_{ADC,6}[n])^2 - \frac{2}{4^{6-q}} e_{ADC,6}[n] \cdot \\ (1 + \alpha_{1,q}) e_{ADC,q}[n] + \\ \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \end{array} \right) c_q^2[n] \right] \\
& E \left[\left(\begin{array}{l} (4^{q-6} e_{ADC,6}[s])^2 - \frac{2}{4^{6-q}} e_{ADC,6}[s] \cdot \\ (1 + \alpha_{1,q}) e_{ADC,q}[s] + \\ \alpha_{1,q} c_q[s] + \alpha_{3,q} (e_{ADC,q}[s] + c_q[s])^3 \end{array} \right) c_q^2[s] \right] \quad (138)
\end{aligned}$$

Using the properties of the calibration sequences in Table 2, the square of $\tilde{\eta}_{2,q}$ mean value can be found as

$$E[\tilde{\eta}_{2,q}^2] = \left(\frac{1}{A} \frac{1}{P} E \left[\sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])^2 c_q[n] c_{1,q}[n] \right] \right)^2$$

Subtracting $E[\tilde{\eta}_{2,q}]^2$ from $E[\tilde{\eta}_{2,q}^2]$,

$$\begin{aligned}
\text{Var}[\tilde{\eta}_{2,q}] &= \frac{1}{16A^4P^2} \cdot \\
& E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^4 c_q^4[n] \right] - \\
& \left(E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^2 c_q^2[n] \right] \right)^2 + \\
& 2E \left[\left[\left(4^{q-6} e_{ADC,6}[n] \right)^2 - \frac{2e_{ADC,6}[n]}{4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right) \right. \right. \\
& \left. \left. + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right] - \\
& \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^2 c_q^4[n] \right] \\
& \sum_{n=m}^{m+P-1} \left(E \left[\left[\left(4^{q-6} e_{ADC,6}[n] \right)^2 - \frac{2e_{ADC,6}[n]}{4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right) \right. \right. \right. \\
& \left. \left. + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right] c_q^2[n] \right] \\
& \cdot E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^2 c_q^2[n] \right] \\
& + E \left[\left[\left(4^{q-6} e_{ADC,6}[n] \right)^2 - \frac{2e_{ADC,6}[n]}{4^{6-q}} \left((1 + \alpha_{1,k}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right) \right. \right. \\
& \left. \left. + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right] c_q^4[n] \right] - \\
& \left(E \left[\left[\left(4^{q-6} e_{ADC,6}[n] \right)^2 - \frac{2}{4^{6-q}} e_{ADC,6}[n] \left((1 + \alpha_{1,k}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] \right) \right. \right. \right. \\
& \left. \left. + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right] c_q^2[n] \right] \right)^2
\end{aligned}$$

By inspection it can be determined that the maximum variance of $\tilde{\eta}_{2,q}$ is negligible comparing the bounds established for $\tilde{\gamma}_{1,q}$ and $\tilde{\gamma}_{3,q}$ as, from the equation above, it follows that each term is multiplied by either $c_q^4[n]$ (where $E[c_q^4[n]] = 40A^4$), $c_q^2[n]$ (which expected value is squared $(E[c_q^2[n]])^2 = 16A^4$), or higher powers of $c_q[n]$. Without much effort, an upper bound can be established by applying the inequalities below which use the information that input into residue amplifier (in order to avoid

clipping) should not be bigger than Δ ,

$$E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^4 c_q^4[n] \right] \leq \Delta^4 E[c_q^4[n]],$$

$$E \left[\left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right)^2 c_q^2[n] \right] \leq \Delta^2 E[c_q^2[n]]$$

$$2 \left[E \left[\left(\frac{e_{ADC,6}^2[n]}{4^{12-2q}} - \frac{e_{ADC,6}[n]}{0.5 \cdot 4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right) c_q^4[n] \right] \right] \leq \frac{\Delta^3}{8} e_{ADC\max} E[c_q^4[n]]$$

$$2 \left[E \left[\left(\frac{e_{ADC,6}^2[n]}{4^{12-2q}} - \frac{2e_{ADC,6}[n]}{4^{6-q}} \left((1 + \alpha_{1,q}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right) c_q^2[n] \right] \right] \leq \frac{\Delta^3}{8} e_{ADC\max} (E[c_q^2[n]])^2$$

$$E \left[\left(\frac{e_{ADC,6}^2[n]}{4^{12-2q}} - \frac{e_{ADC,6}[n]}{0.5 \cdot 4^{6-q}} \left((1 + \alpha_{1,k}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right) c_q^4[n] \right] \leq \frac{e_{ADC\max}^2 \Delta^2}{16^2} E[c_q^4[n]]$$

$$E \left[\left(\frac{e_{ADC,6}^2[n]}{4^{12-2q}} - \frac{e_{ADC,6}[n]}{0.5 \cdot 4^{6-q}} \left((1 + \alpha_{1,k}) e_{ADC,q}[n] + \alpha_{1,q} c_q[n] + \alpha_{3,q} (e_{ADC,q}[n] + c_q[n])^3 \right) \right) c_q^2[n] \right] \leq \left(\frac{e_{ADC\max} \Delta}{16} \right) E[c_q^2[n]]$$

As the upper bound for the first two terms in $Var[\tilde{\eta}_{2,q}]$ is the largest (their magnitude is about 16 times bigger than the magnitude of the following two terms), the variance of $\tilde{\eta}_{2,q}$ can be well-bounded with

$$\begin{aligned} Var[\tilde{\eta}_{2,q}] &\leq \frac{1}{16A^4P^2} \sum_{n=m}^{m+P-1} \left[E \left[\left(\begin{aligned} &\left((1+\alpha_{1,q})e_{ADC\max}[n] + \alpha_{1,q}c_q[n] \right)^4 \\ &+ \alpha_{3,q} \left(e_{ADC\max}[n] + c_q[n] \right)^3 \end{aligned} \right) c_q^4[n] \right] - \right. \\ &\quad \left. \left(E \left[\left(\begin{aligned} &\left((1+\alpha_{1,q})e_{ADC\max}[n] + \alpha_{1,q}c_q[n] \right)^2 \\ &+ \alpha_{3,q} \left(e_{ADC\max}[n] + c_q[n] \right)^3 \end{aligned} \right) c_q^2[n] \right] \right)^2 \right] \\ &\leq \frac{1}{16A^4P} \left(\Delta^4 E[c_q^4[n]] + \Delta^4 \left(E[c_q^2[n]] \right)^2 \right) = \frac{56A^4\Delta^4}{16A^4\Delta^4} = \frac{3.5\Delta^4A^2}{A^2P} \end{aligned}$$

Using the definitions in (113), (114) and (121), the correlation between $\tilde{\gamma}_{1,q}$ and $\tilde{\gamma}_{3,q}$ equals to

$$\begin{aligned} E[\tilde{\gamma}_{1,q}\tilde{\gamma}_{3,q}] &= E \left[\frac{1}{A} \frac{1}{P} \sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{1,q}[n] \frac{1}{6A^3} \frac{1}{P} \sum_{s=m}^{m+P-1} (r_q[s] + c_q[s]) c_{3,q}[s] \right] \\ &= \frac{1}{6A^4P^2} E \left[\sum_{n=m}^{m+P-1} (r_q[n] + c_q[n]) c_{1,q}[n] \right] E \left[\sum_{s=m}^{m+P-1} (r_q[s] + c_q[s]) c_{3,q}[s] \right] + \\ &\quad \frac{1}{6A^4P^2} E \left[\sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])^2 c_{3,q}[n] c_{1,q}[n] \right] - \\ &\quad \frac{1}{6A^4P^2} \sum_{n=m}^{m+P-1} E \left[(r_q[n] + c_q[n]) c_{1,q}[n] \right] E \left[(r_q[n] + c_q[n]) c_{3,q}[n] \right] \\ &= E[\tilde{\gamma}_{1,q}] E[\tilde{\gamma}_{3,q}] + \frac{1}{6A^4P^2} E \left[\sum_{n=m}^{m+P-1} (r_q[n] + c_q[n])^2 c_{3,q}[n] c_{1,q}[n] \right] \\ &\quad - \frac{\alpha_{3,q} \cdot \left(P\alpha_{1,q} + \alpha_{3,q} \left(\sum_{n=m}^{m+P-1} e_{ADC,q}^2[n] + 10A^2 \cdot P \right) \right)}{P^2} \end{aligned}$$

Neglecting the last term in the expression above (as this term goes very quickly to

zero, the difference between the mean of $\tilde{\gamma}_{1,q}\tilde{\gamma}_{3,q}$ and $E[\tilde{\gamma}_{1,q}]E[\tilde{\gamma}_{3,q}]$ is well-

approximated by the term $\frac{1}{6A^4P^2}E\left[\sum_{n=m}^{m+P-1}(r_q[n]+c_q[n])^2c_{3,q}[n]c_{1,q}[n]\right]$. That is,

$$\begin{aligned} & \frac{1}{6A^4P^2}E\left[\sum_{n=m}^{m+P-1}(r_q[n]+c_q[n])^2c_{3,q}[n]c_{1,q}[n]\right] = \frac{1}{6A^4P^2}\sum_{n=m}^{m+P-1}E\left[(r_q[n]+c_q[n])^2c_{3,q}[n]c_{1,q}[n]\right] \\ & = \frac{1}{6A^4P^2}\sum_{n=m}^{m+P-1}E\left[\left(\begin{aligned} & (1+\alpha_{1,q})e_{ADC,q}[n]+\alpha_{1,q}c_q[n]+ \\ & \alpha_{3,q}(e_{ADC,q}[n]+c_q[n])^3-4^{q-6}e_{ADC,6}[n] \end{aligned}\right)^2c_{3,q}[n]c_{1,q}[n]\right] \\ & = \frac{1}{6A^4P^2}\sum_{n=m}^{m+P-1}E\left[\left(\begin{aligned} & \alpha_{1,q}^2c_q^2[n]+6(1+\alpha_{1,q})\alpha_{3,q}e_{ADC,q}^2[n]c_q^2[n]+ \\ & 6\alpha_{1,q}\alpha_{3,q}e_{ADC,q}^2[n]c_q^2[n]+ \\ & \alpha_{3,q}^2(15e_{ADC,q}^4[n]c_q^2[n]+15e_{ADC,q}^2[n]c_q^4[n]+c_q^6[n]) \\ & -2\cdot 4^{q-6}e_{ADC,6}[n]\cdot(\alpha_{1,q}c_q[n]+\alpha_{3,q}(e_{ADC,q}[n]+c_q[n])^3) \end{aligned}\right)c_{3,q}[n]c_{1,q}[n]\right] \end{aligned}$$

where, in the worst case scenario when $|e_{ADC,q}[n]| = e_{ADC,max}$ (when the correlation is

the largest), this term can be well-bounded by $\frac{9}{6A^2P}(1+\alpha_{1,q})\alpha_{3,q}e_{ADC,max}^2[n]$, which is

approximately 218 times smaller than the variance of $\tilde{\gamma}_{3,q}$.

IX. REFERENCE

47. P. Wambacq, W. Sansen, *Distortion Analysis of Analog Integrated Circuits*, Kluwer Academic Publishers
48. A. Panigada, I. Galton, "A 130 mW 100 Ms/s Pipelined ADC with 69 dB SNDR Enabled by Digital Harmonic Distortion Correction," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3314-3328, December 2009
49. B. Murmann, B. Boser, "A 12b 75Ms/s Pipelined ADC using Open-Loop Residue Amplification," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2040-2050, December 2003
50. H. Van de Vel, B. A. J. Butler, H. van der Ploeg, M. Vertregt, G. J. G. M. Gleen, E. J. F. Paulus, "A 1.2-V 250-mW 14-b 100-Ms/s Digitally Calibrated Pipeline ADC in 90-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1047-1056, April 2009
51. A. Panigada, I. Galton, "Digital Background Correction of Harmonic Distortion in Pipelined ADCs," *IEEE Transactions on Circuits and Systems – I: Regular Papers*, vol. 53, no. 9, pp. 1885 – 1895, September 2006.
52. N. Rakuljic, I. Galton, "Suppression of Quantization-Induced Convergence Error in Pipelined ADCs with Harmonic Distortion Correction", accepted for publication in *IEEE Transactions on Circuits and Systems – I: Regular Papers*
53. D. G. Manolakis, et al, "Statistical and Adaptive Signal Processing, Spectral", Artech House