UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

University of California
San Diego

Analyses of Dynamic Element Matching Techniques for Data Conversion

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

Doctor of Philosophy

in Electrical and Computer Engineering (Communication Theory & Systems)

by

Henrik Tholstrup Jensen

Committee in charge:

Professor Ian Galton, Chair
Professor LeRoy Dorman
Professor William Hodgkiss
Professor Larry Larson
Professor Paul Siegel

1997

UMI Number: 9732715

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

To my Parents...

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

PREFACE

This dissertation is a collection of three papers that are intended for separate publication in *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, each paper forming a separate chapter of the dissertation. Specifically, the first paper has been reviewed by the IEEE, revised, and accepted for publication, the second paper has been submitted to the IEEE for review, and the third paper is in preparation for review, and is in large ready for submission. Furthermore, the research covered in Chapter 1 has been presented at the *1996 IEEE International Symposium on Circuits and Systems*, May 12-15, 1996, Atlanta, Georgia, and the research covered in Chapters 2 and 3 is to be presented at the *1997 IEEE International Symposium on Circuits and Systems*, June 9-12, 1997, Hong Kong.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Ian Galton, my advisor and close friend, for his continued guidance, support, and encouragement throughout this endeavor. I have indeed been very fortunate to have him for my advisor and teacher during my years in graduate school. His engineering ingenuity, command of the technical topics, and abilities as a teacher are nothing less than exceptional. At the personal level, his concern about my well-being during my student career has gone far beyond the call of duty. No expenses have been spared in terms of financial support, state-of-the-art equipment, and funding for participation in relevant technical conferences, in order to provide me with the optimal environment to perform research and to give me the full doctoral student experience.

Also many thanks to the rest of the Galton family; wife Kerry, daughter Riley, and son Mitchell, for their warm friendship and for excellent company on the many dinner trips to the Souplantation and other fine dining establishments...

I am grateful to Professors LeRoy Dorman and William Hodgkiss of the Scripps Institution of Oceanography, UCSD, and to Professors Larry Larson and Paul Siegel of the Department of Electrical and Computer Engineering, UCSD, for serving on the doctoral committee. I am also highly appreciative of the competence and professional, positive attitude exhibited by the staff of the Department of Electrical and Computer Engineering, UCSD, throughout my stay here.

This work was supported in part by the National Science Foundation under Grants MIP-9309656 and EEC-9320381.

Finally I wish to thank the Fulbright Commission of Denmark for its services and advice provided in the years of 1991 and 1992 when I applied to graduate school in the U.S., and thanks goes to the Knud Højgaard Foundation, the Radio-Parts

Foundation, the Denmark-America Foundation, the International Services Office of Odense University, and the Jacobsen Foundation (Risingskolen, Odense) for their generous support during my initial years in the U.S..

Chapter Acknowledgment: The text of Chapter 1, in part or in full, consists of the material that will appear as a "Regular Paper" in a future edition of *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing.* The dissertation author was the primary researcher and the co-author listed in the publication directed and supervised the research which forms the basis for the chapter.

VITA

March 16, 1968	Born, Odense, Denmark
1992	Bachelor of Science, Odense University, Odense, Denmark
1992-1993	Fulbright Scholar and Teaching Assistant, Department of Electrical and Computer Engineering, University of California, Irvine
1993-1994	Research Assistant, Department of Electrical and Computer Engineering, University of California, Irvine
1994	Master of Science, University of California, Irvine
1994-1997	Research Assistant, Department of Electrical and Computer Engineering, University of California, Irvine, and Department of Electrical and Computer Engineering, University of California, San Diego
1997	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

H. T. Jensen, I. Galton, "Yield estimation of a second-order $\Delta\Sigma$ADC employing a noise-shaping DAC," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing,* in preparation.

H. T. Jensen, I. Galton, "An analysis of the partial randomization dynamic element matching technique," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing,* under review.

H. T. Jensen, I. Galton, "A low-complexity dynamic element matching DAC for direct digital synthesis,"*IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing,* accepted for publication.

I. Galton, H. T. Jensen, "Oversampling parallel delta-sigma modulator A/D conversion," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing,* vol. 43, no. 12, pp. 801-810, December 1996.

I. Galton, H. T. Jensen, "Delta-sigma modulator based A/D conversion without oversampling," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing,* vol. 42, no. 12, pp. 773-784, December 1995.

H. T. Jensen, I. Galton, "Yield estimation of a first-order noise-shaping D/A converter," *IEEE Proceedings of the International Symposium on Circuits and Systems 1997*, ISCAS 1997, Hong Kong.

H. T. Jensen, I. Galton, "A performance analysis of the partial randomization dynamic element matching DAC architecture," *IEEE Proceedings of the International Symposium on Circuits and Systems 1997*, ISCAS 1997, Hong Kong.

H. T. Jensen, I. Galton, "A Hardware-efficient DAC for Direct Digital Synthesis," *IEEE Proceedings of the International Symposium on Circuits and Systems 1996*, ISCAS 1996, Atlanta, GA.

I. Galton, D. A. Towne, J. Rosenberg, H. T. Jensen, "Clock Distribution Using Coupled Oscillators," *IEEE Proceedings of the International Symposium on Circuits and Systems 1996*, ISCAS 1996, Atlanta, GA.

H. T. Jensen, I. Galton, "A robust parallel delta-sigma A/D converter architecture," *IEEE Proceedings of the International Symposium on Circuits and Systems 1995*, ISCAS 1995, Seattle, WA.

ABSTRACT OF THE DISSERTATION

Analyses of Dynamic Element Matching Techniques for Data Conversion

by

Henrik Tholstrup Jensen
Doctor of Philosophy in Electrical and Computer Engineering
(Communication Theory & Systems)

University of California, San Diego, 1997

Professor Ian Galton, Chair

IDEALLY, a digital-to-analog converter (DAC) circuit converts a sequence of numbers represented in a digital format into exactly the same sequence of numbers represented in an analog format. As an example, the digital sequence

$$d[n] = \{\cdots, 2, 1, 3, 2, 1, \cdots\}$$

could ideally be converted into analog form as the sequence of voltages

$$a_i[n] = \{\cdots, 2\mathrm{V}, 1\mathrm{V}, 3\mathrm{V}, 2\mathrm{V}, 1\mathrm{V}, \cdots\}.$$

However, non-idealities associated with the DAC fabrication process result in circuit imperfections which cause non-ideal behavior of the DAC. As an example, a fabri-

cated DAC might convert the digital sequence d into analog form as the sequence of voltages

$$a_{ni}[n] = \{\cdots, 2.1\text{V}, 0.9\text{V}, 3.0\text{V}, 2.1\text{V}, 0.9\text{V}, \cdots\}.$$

Thus, a practical DAC introduces analog error, defined as

$$a_i[n] - a_{ni}[n] = \{\cdots, -0.1\text{V}, 0.1\text{V}, 0.0\text{V}, -0.1\text{V}, 0.1\text{V}, \cdots\}.$$

In general, the analog error is a non-linear function of the digital input, so the DAC is a non-linear device, and the analog error tends to be strongly correlated with the digital input. This unfortunate property causes the DAC to be the performance limiting component in many electronic systems. For example, such is the case with direct digital synthesizers, which—using mostly digital circuitry—generate high spectral-purity sinusoidal analog signals in wireless communications systems. Another application wherein the analog error has detrimental effects is in the *oversampling $\Delta\Sigma$ data converter.*

In this dissertation, recently developed digital signal processing algorithms—also known as *dynamic element matching (DEM) techniques*—designed to eliminate or minimize the detrimental effects of the analog error are presented and their performance analyzed. Rather than simply trying to minimize the analog error by improvements in the DAC fabrication process, the non-ideal behavior is accepted as inevitable, but its detrimental effects are mitigated using a DEM technique. Various DEM techniques are presented, each of which is optimal for a particular DAC application. In all cases, a drastic decrease of the detrimental effects of the analog error has been verified with theoretical analyses and demonstrated with simulation data.

Chapter 1

A Low-Complexity Dynamic Element Matching DAC for Direct Digital Synthesis[†]

Henrik T. Jensen, *Student Member, IEEE,* and Ian Galton, *Member, IEEE* [‡]

***Abstract*—This paper presents and analyzes a new dynamic element matching technique for low harmonic distortion digital-to-analog conversion. The benefit of the technique over the prior art is significantly reduced hardware complexity with no reduction in performance. It is particularly appropriate for applications such as direct digital synthesis in wireless communications systems, wherein low hardware complexity is essential in addition to low harmonic distortion.**

I. INTRODUCTION

AS a largely digital technique for generating high spectral-purity sinusoidal analog signals, direct digital synthesis (DDS) is increasingly used in wireless communications systems. The main limitation in most DDS systems is imposed by the front-end digital-to-analog converter (DAC) required to convert the digitally synthesized sinusoidal sequence into an analog waveform. In particular, non-ideal circuit behavior causes the DAC to introduce *DAC noise.* At least a component of the DAC noise is a non-linear function of the input sequence, so harmonic distortion

[†] Accepted for publication: IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING

[‡] The authors are with the Department of Electrical and Computer Engineering, UNIVERSITY OF CALIFORNIA, SAN DIEGO, La Jolla, CA 92093-0407, USA; email: htjensen@ece.ucsd.edu and galton@ece.ucsd.edu

is introduced that places an upper bound on the achievable spurious-free dynamic range (SFDR) of the overall system. As shown in [1] DDS applications typically require only moderate resolution (e.g., 5-12 bits) provided the harmonic distortion introduced by the DAC is low. For example, an extremely low-complexity digital portion of an 8-bit DDS system has been demonstrated that is capable of achieving a minimum SFDR of 90 dB provided the minimum SFDR of the DAC is 90 dB or greater.

Thus, a remaining problem is to develop moderate-resolution DACs that achieve such low levels of harmonic distortion. In the past, *dynamic element matching* (DEM) techniques have been successfully applied to decorrelate the DAC noise from the input signal in various DAC topologies. A particularly promising topology involves the use of a bank of 1-bit DACs, the outputs of which are summed together to yield a single multi-bit DAC [2], [3], [4]. For most digital input values, there are many possible input codes to the bank of 1-bit DACs that nominally yield the desired analog output value. Thus, the DAC noise arising from errors introduced by the 1-bit DACs can be "scrambled" by randomly selecting one of the appropriate codes for each digital input value. Although DACs based on this approach have been shown experimentally [2], [3] and through quantitative analysis [4] to achieve excellent SFDRs, the presented DACs suffer from excessive digital hardware complexity. For example, an 8-bit DAC based on the approach used in [2] requires 1024 binary switches and 1024 independent random control bits.

This paper presents a new DEM technique suitable for DACs applicable to DDS. The DEM technique scrambles the DAC noise such that conversion performance similar to that of the prior art is achieved, but with much lower hardware complexity. The proposed DEM technique allows for a varying degree of scrambling, providing a trade-off between harmonic distortion suppression and hardware complexity. Two versions of the architecture are considered separately: a version with the full degree of scrambling, referred to as *full randomization* DEM, and a version with a reduced degree of scrambling, referred to as *partial randomization* DEM. With full randomization DEM, the DAC noise is white and the SFDR is optimal (infinite, in principle). Theoretical results quantifying the performance of full randomization DEM are presented and closely supported by simulation results. Simulations indicate that very good SFDR performance is achieved with partial randomization DEM, and while both DEM versions have much lower hardware complexity than the prior art, the greatest hardware-efficiency is offered by partial randomization DEM. To illustrate these results, example 8-bit DACs with 0.5% static analog mismatch errors are considered in detail; 502 binary switches and 8 independent random bits are required to implement full randomization DEM, whereas merely 46 binary switches and 3 independent random bits are required with partial randomization DEM to provide more than 97 dB of SFDR.

The remainder of the paper is divided into sections as follows. Section II reviews the architectures of the low-harmonic distortion DACs presented in [2], [3], and [4] and presents the two versions of the proposed architecture. Section III presents performance details for full randomization DEM. Section IV provides an IC fabrication yield estimate for full randomization DEM based on the results of the theoretical analysis. In Section V it is demonstrated by means of simulation results how partial randomization DEM can significantly suppress harmonic distortion while offering additional hardware reductions. A quantitative discussion of the hardware requirements of full randomization DEM and partial randomization DEM is given in Section VI. The theoretical results stated in Section III are derived in detail in the appendices.

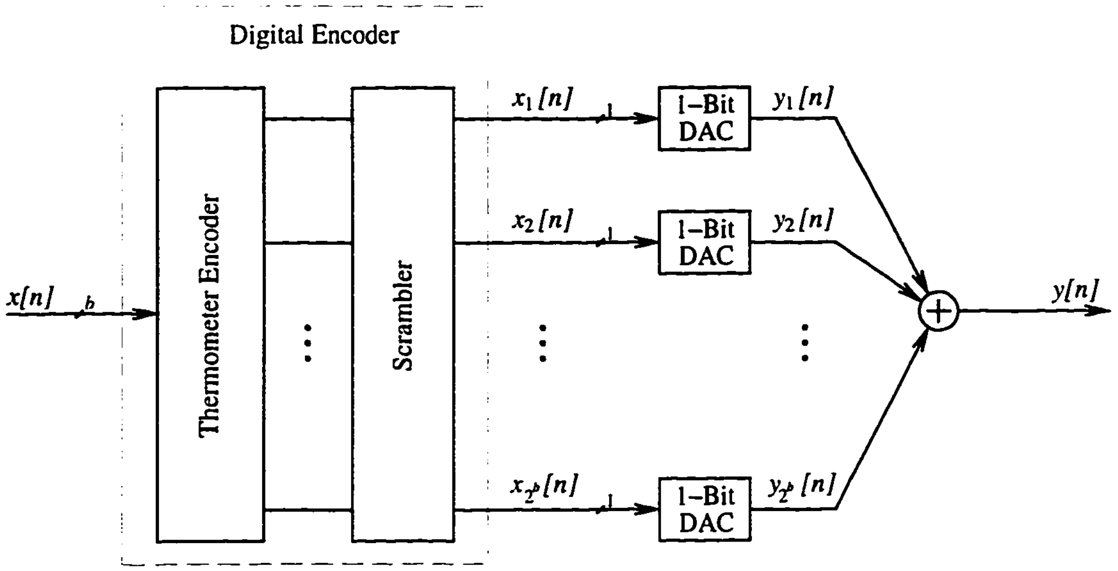


Figure 1.1: The high-level topology of the low-harmonic distortion DACs presented in [2] and [4].

II. LOW-HARMONIC DISTORTION DAC APPROACHES

A. Background and Prior Art

The high-level topology shared by the DACs presented in [2] and [4] is shown in Figure 1.1. The digital input, $x[n]$, is a sequence of unsigned b-bit binary numbers less than 2^b, i.e,

$$x[n] \in \{0, 1, \ldots, x_{max}\}, \quad x_{max} = 2^b - 1.$$

The DAC consists of a *digital encoder*, 2^b one-bit DACs referred to as *unit DAC-elements*, and an analog output summing node. At the high level of Figure 1.1, the digital encoder maps each input sample to 2^b output bits, $x_1[n] \ldots x_{2^b}[n]$, such that

$$\sum_{i=1}^{2^b} x_i[n] = x[n]. \tag{1}$$

The unit DAC-elements operate according to

$$y_r[n] = \begin{cases} 1 + e_{h_r}, & \text{if } x_r[n] = 1; \\ e_{l_r}, & \text{if } x_r[n] = 0; \end{cases} \tag{2}$$

where $y_r[n]$ denotes the analog output of the r^{th} unit DAC-element, and e_{h_r} and e_{l_r} are errors in the analog output levels arising from inevitable non-idealities in the IC fabrication process. Throughout the paper, these errors are assumed to be time-invariant, but otherwise arbitrary [2] and are referred to as *static DAC-element errors.* The r^{th} unit DAC-element is said to be *selected* when $x_r[n] = 1$. The DAC output $y[n]$ is formed by the analog output summing node such that

$$y[n] = \sum_{i=1}^{2^b} y_i[n]. \tag{3}$$

It follows from (1), (2), and (3) that $y[n] = x[n]$ in the absence of static DAC-element errors. However, as shown in [5], with non-zero static DAC-element errors, the DAC output has the form

$$y[n] = \alpha x[n] + \beta + e[n], \tag{4}$$

where α is a *constant gain,* β is a *DC offset,* and $e[n]$ is a conversion error term referred to as *DAC noise.* The purpose of the digital encoder is to scramble the DAC noise by randomly selecting the unit DAC-elements such that $e[n]$ is white and uncorrelated with $x[n]$. To accomplish this objective, the digital encoders of the prior art employ a *thermometer-encoder* and a *scrambler.* During each clock period, the thermometer encoder deterministically sets $x[n]$ of its 2^b output bits to "1" and the remaining $2^b - x[n]$ of its output bits to "0". The scrambler randomly permutes the resulting 2^b bits, thereby selecting $x[n]$ of the unit DAC-elements at random. As explained in [2], the effect is to randomly modulate the DAC noise without modulating the signal component of the DAC output. The random modulation effectively converts the harmonic distortion, i.e., spurious tones, into white noise.

The scrambler implements the random permutation using a network of *binary switches*, each controlled by a *random control bit.* The binary switch is a simple 2-input × 2-output device that, depending upon the value of the random control bit, either passes the inputs directly through to the outputs or connects the inputs to the outputs in reverse order. The random control bit of each binary switch is

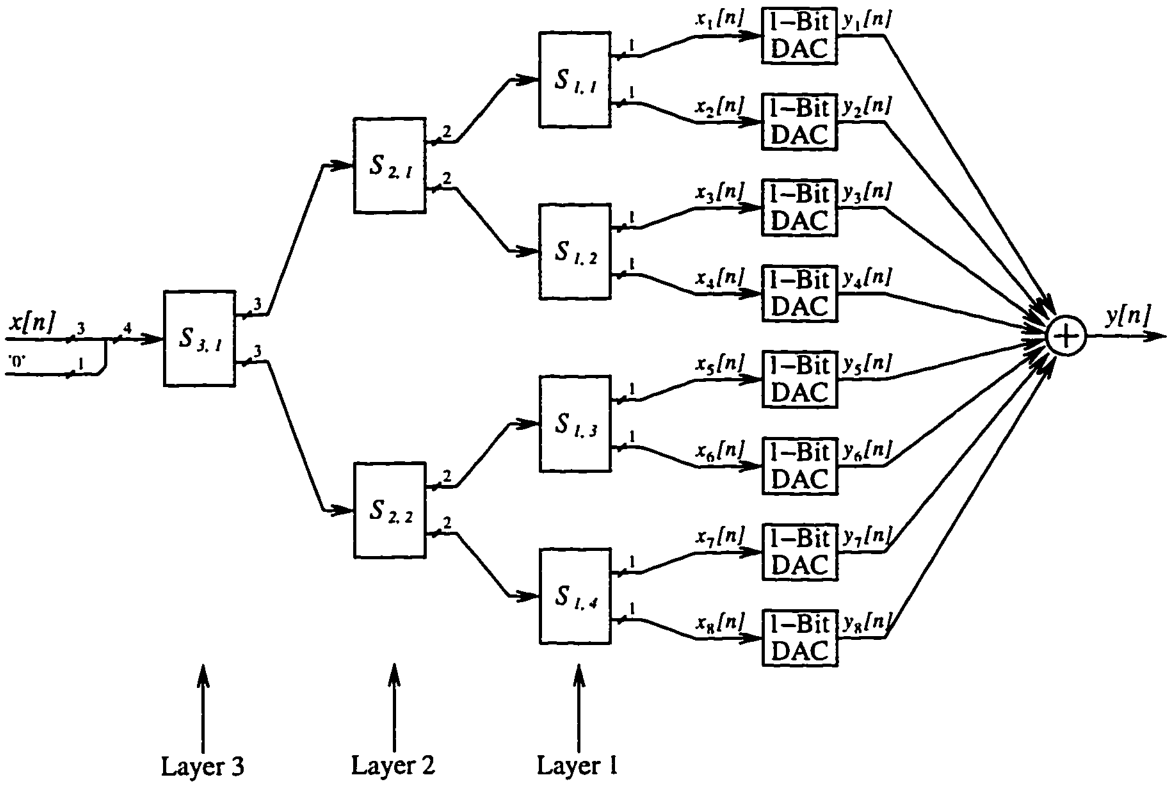
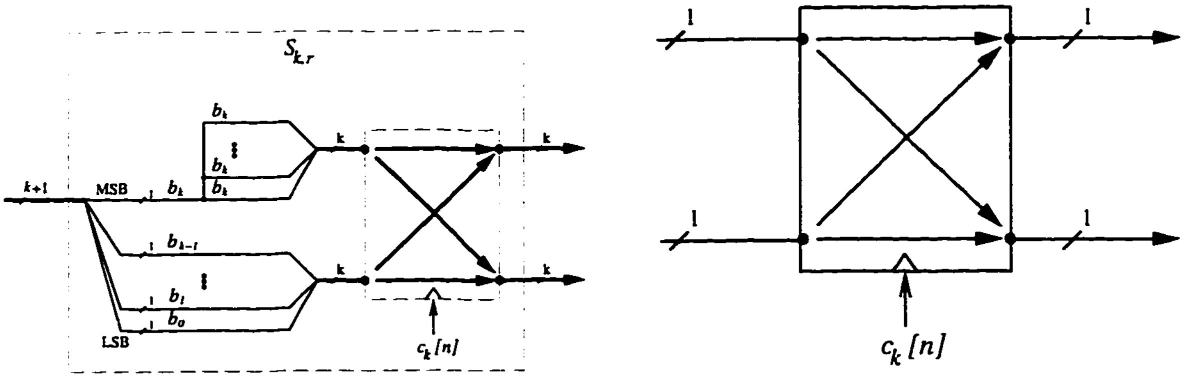


Figure 1.2: A 3-bit version of the proposed DAC architecture.

ideally a white random bit-sequence, statistically independent of the random control bits applied to the other binary switches. Thus, implementing the digital encoders presented in [2] and [4] requires as many random control bits as binary switches. The digital encoder in [4] is capable of randomly connecting its 2^b-bit inputs to its 2^b-bit outputs in any of the $2^b!$ possible combinations. The digital encoder in [2] implements only a subset of all combinations, being capable of randomly connecting its inputs to its outputs in $2^{b2^{b-1}}$ possible combinations. As will be seen, the digital encoder proposed in this paper implements significantly fewer random input-output mappings than the prior art, yet provides white DAC noise, nonetheless.

B. Proposed DAC topology

The proposed dynamic element matching DAC architecture is shown in Figure 1.2. To simplify the figure, a 3-bit example is shown. The DAC is of the general



(a) (b)

Figure 1.3: Details of (a) the switching block $S_{k,r}$, and (b) the binary switch.

topology introduced in [5]. The tree-structured digital encoder consists of three layers of *switching blocks*, each labeled $S_{k,r}$, where k denotes the layer number and r denotes the position of the switching block in the layer.

Figure 1.3a shows the functional details of the switching block $S_{k,r}$. The switching block has one $k+1$-bit input, two k-bit outputs, and a random control bit input, $c_k[n]$. The random control bit is common to *all* the switching blocks within the k^{th} layer (for clarity, the random control bits are not shown in Figure 1.2). The $S_{k,r}$ switching block operates such that when $c_k[n]$ is high, the most significant bit (MSB), b_k, of the input is mapped to all k bits of the top output, and the remaining k bits of the input are mapped directly to the k bits of the bottom output. When $c_k[n]$ is low, the situation is as above except that the mappings are interchanged. Thus, it follows that $S_{k,r}$ can be implemented using k binary switches, all controlled by $c_k[n]$. Figure 1.3b shows the binary switch controlled by $c_k[n]$. The process of randomly mapping the input to the outputs is referred to as *random switching.* At the outermost layer, i.e., $k = b$, the DAC input $x[n]$ is assigned to the $S_{b,1}$ input bits b_1 through b_b, and a zero is assigned to the input bit b_0, as indicated in Figure 1.2. It is shown in appendix 1.A that the digital encoder obtained by interconnecting the switching blocks of Figure 1.3a as shown in Figure 1.2 indeed satisfies (1).

Motivated by the results of the simulated performance presented in sections III and V, two versions of the proposed architecture are now defined. The term *full randomization* DEM refers to a DAC with random switching in *all* layers, i.e., layers 1 through b. The term *partial randomization* DEM refers to a DAC with random switching in a limited number of layers, i.e., in layers R through b, where $2 \leq R \leq b-1$. As an example of partial randomization DEM, consider the 8-bit DAC of Figure 1.4, where random switching is performed in layers 6 through 8. Layers 1 through 5 have no effect on the scrambling of the DAC noise, so it follows that these layers can be eliminated and substituted by eight nominally identical *DAC banks*, each with a 6-bit input. The details of the DAC bank are shown in Figure 1.5. The LSB of the input controls a unit DAC-element, whereas the remaining 5 bits control a 5-bit conventional DAC.

III. PERFORMANCE DETAILS FOR FULL RANDOMIZATION DEM

A. Simulation Results

The simulated performance of an example 8-bit DAC with the proposed architecture is presented in Figure 1.6. Each graph in the figure shows the simulated power spectral density (PSD) relative to x_{max}^2 of a particular signal of the DAC driven by a dithered and DC offset sinusoid. Specifically, $x[n]$ was formed by adding dither to the sequence $A\sin(\omega_0 n) + DC_x$, where $A = 126$, $\omega_0 = \frac{625}{2048}\pi$, and $DC_x = 127.5$, and then quantizing the result to 8 bits. The dither added to the sinusoidal input was a white sequence with a triangular probability density function supported on $(-1, 1)$, so the quantization error was white noise [6].

Figure 1.6a corresponds to $y[n]$ of an ideal DAC (i.e., a DAC with no static DAC-element errors, so $y[n] = x[n]$), Figure 1.6b corresponds to $y[n]$ with no random

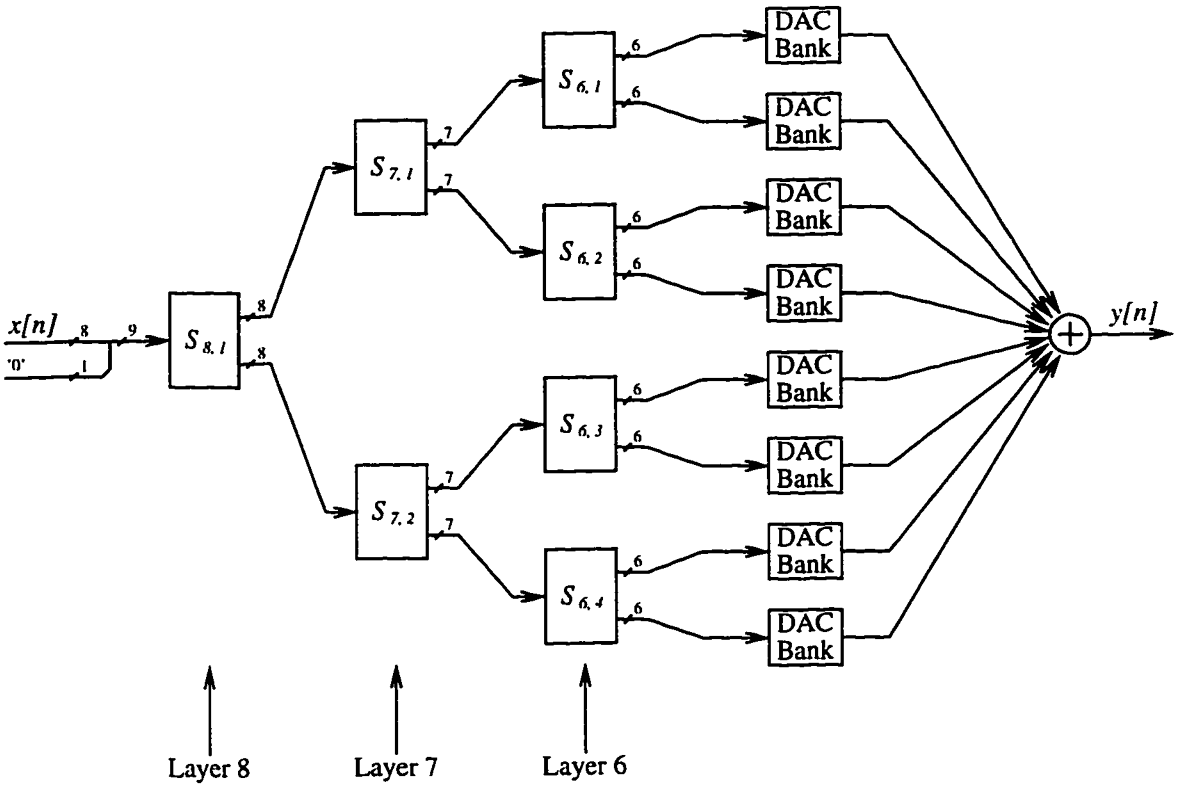


Figure 1.4: An 8-bit DAC with partial randomization DEM.

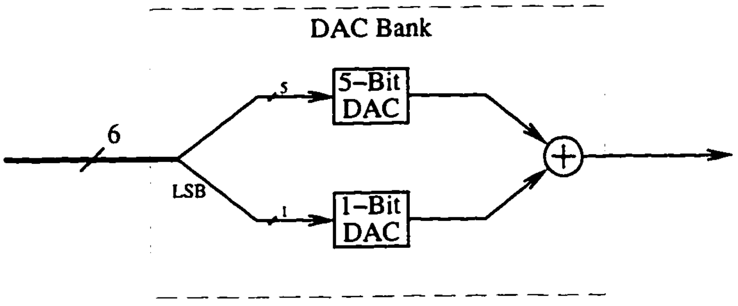
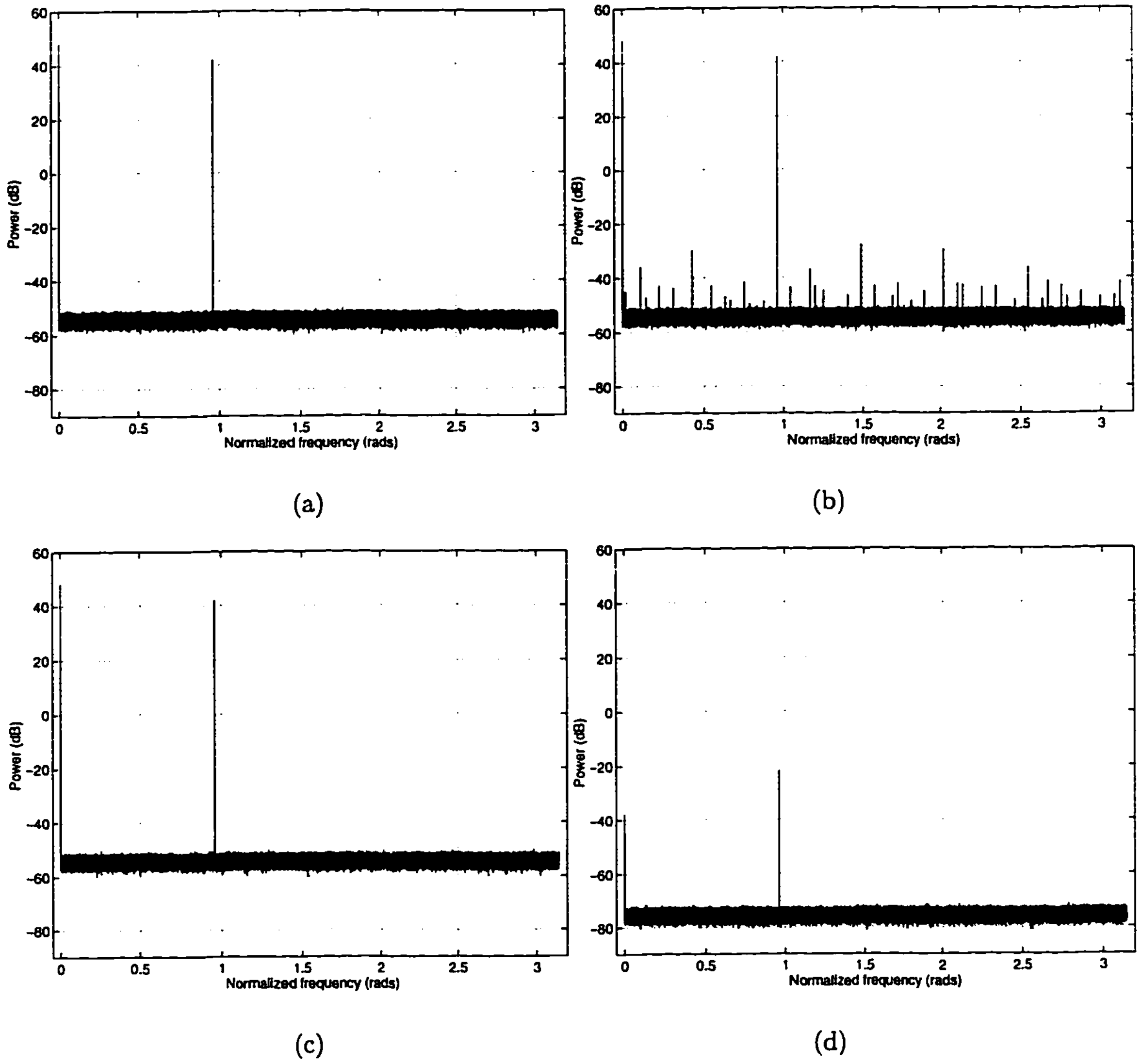


Figure 1.5: Details of the DAC bank.

switching (the digital encoder thus being equivalent to a thermometer encoder), Figure 1.6c corresponds to $y[n]$ with full randomization DEM, and Figure 1.6d corresponds to the signal $y[n] - x[n]$ with full randomization DEM. The static mismatch errors were chosen randomly from a normal distribution with a standard deviation of 0.5%. This represents a conservative estimate relative to the static DAC-element errors expected in practice, but serves to demonstrate the robustness



(a)

(b)

(c)

(d)

Figure 1.6: Simulated PSDs relative to x^2_{max} of an example 8-bit DAC. The plots show (a) the ideal DAC output $y[n]$, (b) the DAC output $y[n]$ with no random switching, (c) the output $y[n]$ with full randomization DEM, and (d) the signal $y[n] - x[n]$ with full randomization DEM.

of the proposed DEM technique [2], [7].

As is evident from the numerous spurs distributed across the spectrum in Figure 1.6b, rather severe harmonic distortion results from the static DAC-element errors in the absence of random switching. The maximum-amplitude spur occurs at a frequency of approximately 1.5π radians, and has power -69.86 dB below the power of the desired sinusoidal signal of frequency ω_0. Numerous additional simulations performed by the authors show that the DAC exhibits similar behavior when driven by inputs of different frequencies. It follows that merely 69.86 dB of SFDR is provided. The data in Figure 1.6c indicates that harmonic distortion is not visible with full randomization DEM. As demonstrated by the simulation results and confirmed in the following section, the DAC easily provides 90 dB of SFDR and is thus applicable to the DDS system mentioned in the Introduction.

Additional details of the simulation results are as follows. The PSDs were each estimated by averaging 16 length-2^{18} periodograms [8]. The frequency scales were normalized such that π corresponds to half the clock rate of the DAC.

B. Performance Equations

A detailed theoretical performance analysis of full randomization DEM is given in appendix 1.B and appendix 1.C. However, for the purpose of comparing simulation results and theory, the main results of the analysis will be stated in the following.

For a b-bit version of the proposed DAC architecture, let $x[n]$ be a deterministic input sequence and let $x^{(i)}[n]$ denote the i^{th} bit of $x[n]$, $1 \le i \le b$. In accordance with the usual definitions, let the *time-average means* of $x[n]$, $x^{(i)}[n]$, and $x^{(i)}[n]x^{(j)}[n]$ be defined as

$$\overline{M}_x = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n],$$

$$\overline{M}_{x^{(i)}} = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x^{(i)}[n],$$

and

$$\overline{M}_{x^{(i,j)}} = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x^{(i)}[n]x^{(j)}[n],$$

respectively, and let the *time-average autocorrelation* of $x[n]$ be defined as

$$\overline{R}_{xx}[k] = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n]x[n+k].$$

The time-average autocorrelation of $y[n]$ is defined analogously with x replaced by y in the above definition. The two main theoretical results of this paper can now be stated as follows:

Result 1. *The output of the proposed DAC with full randomization DEM can be written in the form*

$$y[n] = \alpha x[n] + \beta + e[n], \tag{5}$$

where $e[n]$ is a zero-mean, white random process of the form

$$e[n] = \sum_{i=1}^{b} w_i[n] x^{(i)}[n], \tag{6}$$

and each $w_i[n]$ is a zero-mean, white random process.

Appendix 1.A provides exact formulae for the constants α and β in (5), and expressions for $w_i[n]$ in (6) are developed in appendix 1.B. For now, it suffices to know that the random processes $w_i[n]$ depend only upon the static DAC-element errors and are zero-mean, white, and uncorrelated with $x[n]$. Notice that the above results hold for any underlying statistical distribution or correlation properties of the static DAC-element errors.

Result 2. *If $\overline{M}_x$ and $\overline{R}_{xx}[k]$ exist, then*

$$\overline{M}_y = \alpha \overline{M}_x + \beta \tag{7}$$

and

$$\overline{R}_{yy}[k] = \alpha^2 \overline{R}_{xx}[k] + \overline{\eta} + \overline{\sigma}^2 \delta[k], \tag{8}$$

with probability 1, where

$$\overline{\eta} = 2\alpha\beta \overline{M}_x + \beta^2, \tag{9}$$

and

$$\overline{\sigma}^2 = \sum_{i=1}^{b} \gamma_i \overline{M}_{x^{(i)}} + \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \gamma_{i,j} \overline{M}_{x^{(i,j)}}. \tag{10}$$

Appendix 1.B provides formulae for the constant coefficients γ_i and $\gamma_{i,j}$ in (10). For now, it suffices to know that these coefficients depend only upon the static DAC-element errors. As before, this result holds for any underlying statistical distribution or correlation properties of the static DAC-element errors.

C. Comparison of Simulation Results and Theory

To summarize, (8) states that $\overline{R}_{yy}[k]$ consists of a scaled version of $\overline{R}_{xx}[k]$, a DC offset and white DAC noise. This general conclusion is similar to the corresponding result in [4] and is clearly supported by the simulation results of Figure 1.6.

It follows from (5) that a non-unity value of α causes a term corresponding to $(\alpha - 1)x[n]$ to occur in the signal $y[n] - x[n]$. This scaled version of $x[n]$ occurring in $y[n] - x[n]$ therefore has power

$$\Delta P_x = 10\log_{10}\left[(\alpha - 1)^2\right] \text{ dB} \tag{11}$$

relative to the power of $x[n]$. Similarly, it follows from (5) that the DC component occurring in $y[n] - x[n]$ has power

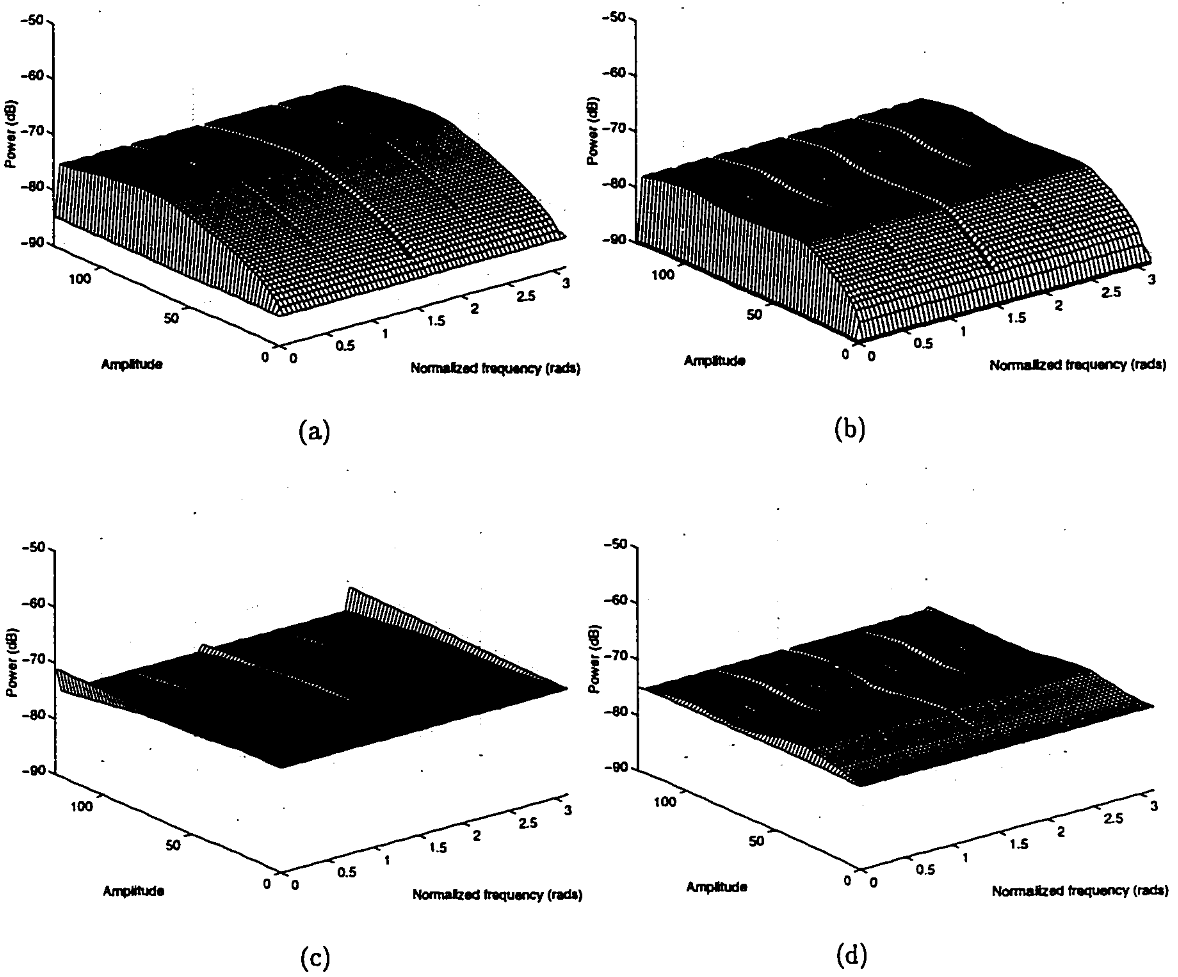
$$\Delta P_{DC} = 10\log_{10}\left[(\alpha - 1 + \frac{\beta}{\overline{M}_x})^2\right] \text{ dB} \tag{12}$$

relative to $(\overline{M}_x)^2$. To compare these predictions with the simulation results, the randomly chosen static DAC-element errors of the example 8-bit DAC were summed according to the formulae for α and β given in appendix 1.A, and resulted in $\alpha - 1 = -6.4361 \times 10^{-4}$ and $\beta = 7.5634 \times 10^{-2}$, respectively. Evaluating (11) and (12) with these values of α and β results in $\Delta P_x = -63.83$ dB and $\Delta P_{DC} = -85.95$ dB. Measuring the offsets corresponding to ΔP_x and ΔP_{DC} using the data of Figure 1.6a and Figure 1.6d yields -63.82 dB and -85.96 dB, respectively, in agreement with theory. Furthermore, evaluating (10) for the simulated values of static DAC-element errors yields a power of the DAC noise of $\overline{\sigma}^2 = -75.45$ dB relative to x_{max}^2. Numerically integrating the DAC noise component of the PSD of Figure 1.6d results in -75.44 dB, in agreement with theory. As an additional comment pertaining to the details of Figure 1.6, it is evident from a comparison of Figure 1.6c and Figure 1.6d that $\overline{\sigma}^2$ is negligible relative to the power of the white quantization error and dither term of $x[n]$.

D. An Interpretation of the Performance Equations

The most significant performance equations in the above are (8) and (10) which state that the DAC noise is white and give a formula for the power of the DAC noise, respectively. As is evident from (10), $\overline{\sigma}^2$ is a linear combination of the time-average means of the individual bits of $x[n]$ plus a linear combination of the time-average means of the products of pairs of bits of $x[n]$. If $x[n]$ is the quantized version of a sinusoid $A\sin(\omega n)$, it follows that $\overline{\sigma}^2$ depends on both amplitude A and frequency ω. This is different from the architecture presented in [4] for which $\overline{\sigma}^2$ only depends on signal amplitude. To demonstrate typical behavior of $\overline{\sigma}^2$, Figures 1.7a-d show plots of $\overline{\sigma}^2$ in dB relative to x_{max}^2 for four nominally identical 8-bit versions of the proposed DAC. The DACs differed only in the static DAC-element errors, which were randomly chosen from a normal distribution with a standard deviation of 0.5%. In each case, the DAC was driven by a sinusoidal input and the plot shows $\overline{\sigma}^2$ computed using (10) as a function of input amplitude and frequency. In general, only minor dependency on frequency is observed, whereas dependency on amplitude is stronger. Notice that there is no clear trend in $\overline{\sigma}^2$ as a function of amplitude, which is in contrast to the behavior of $\overline{\sigma}^2$ in [4] wherein $\overline{\sigma}^2$ decreases with increasing amplitude.

Figures 1.8a-d show plots of $\overline{\sigma}^2$ in dB relative to x_{max}^2 for the same four example 8-bit DACs as in Figure 1.7 when driven by the sum of two DC offset sinusoids, i.e., $x[n]$ being the quantized version of $x_1[n]+x_2[n]+x_{max}/2$, where $x_1[n] = A_1\sin(\omega_1 n)$



(a)

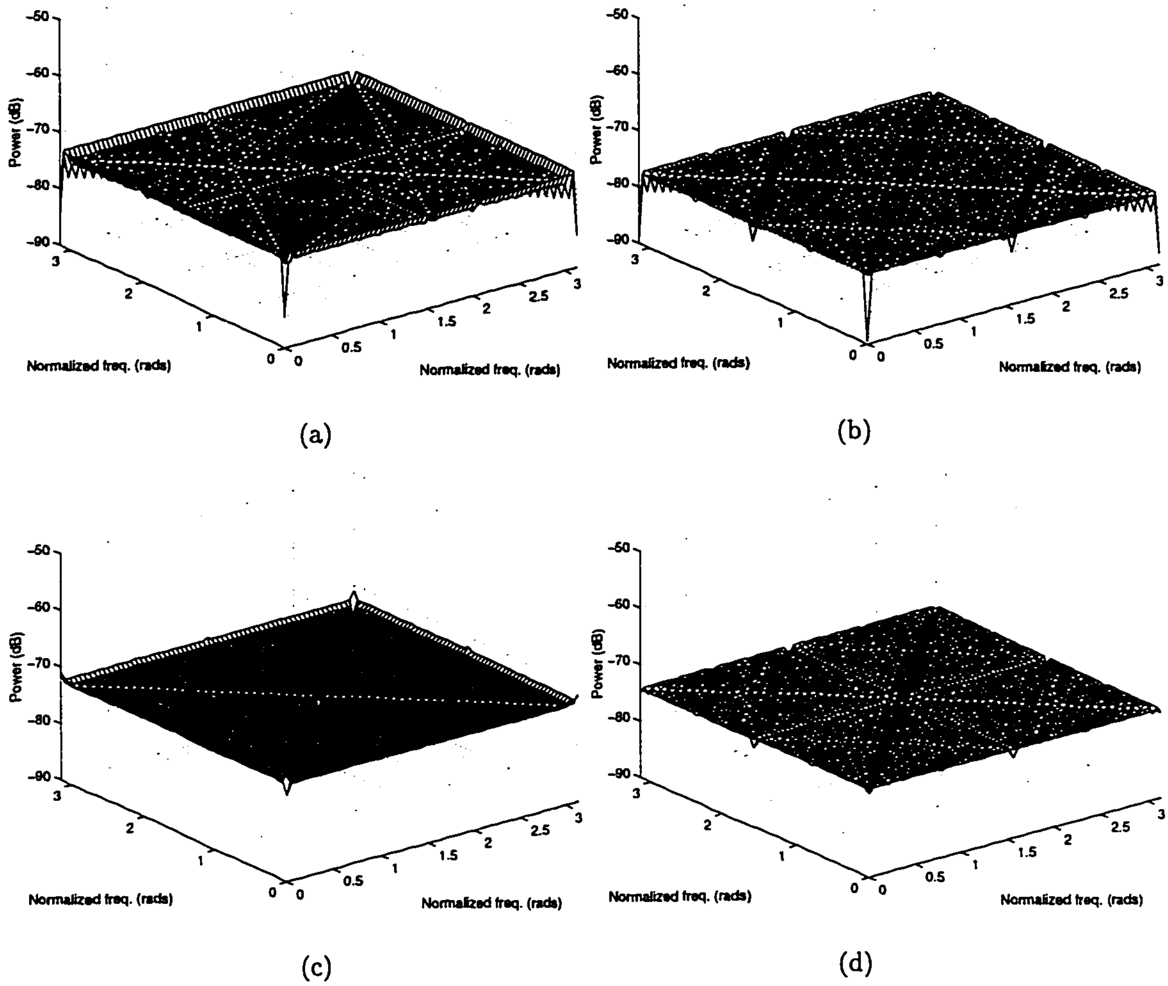
(b)

(c)

(d)

Figure 1.7: Plots of $\overline{\sigma}^2$ in dB relative to x^2_{max} for four nominally identical 8-bit versions of the proposed DAC. The DACs in (a) through (d) differed only in the static DAC-element errors, which were randomly chosen from a normal distribution with a standard deviation of 0.5%. In each case, the DAC was driven by a sinusoidal input and the plot shows $\overline{\sigma}^2$ computed using (10) as a function of input amplitude and frequency.

and $x_2[n] = A_2 \sin(\omega_2 n)$. In each plot, the amplitude of each sinusoidal component was fixed at $A_1 = A_2 = 63$, and $\overline{\sigma}^2$ computed using (10) is shown as a function of ω_1 and ω_2. Again, $\overline{\sigma}^2$ exhibits little dependency on input frequency and attains average values of −74.82 dB, −77.84 dB, −73.02 dB, and −74.33 dB, respectively. Thus, the random variation of the static DAC-element errors of the example DACs causes a spread in the average value of $\overline{\sigma}^2$ of 4.82 dB.



(a)

(b)

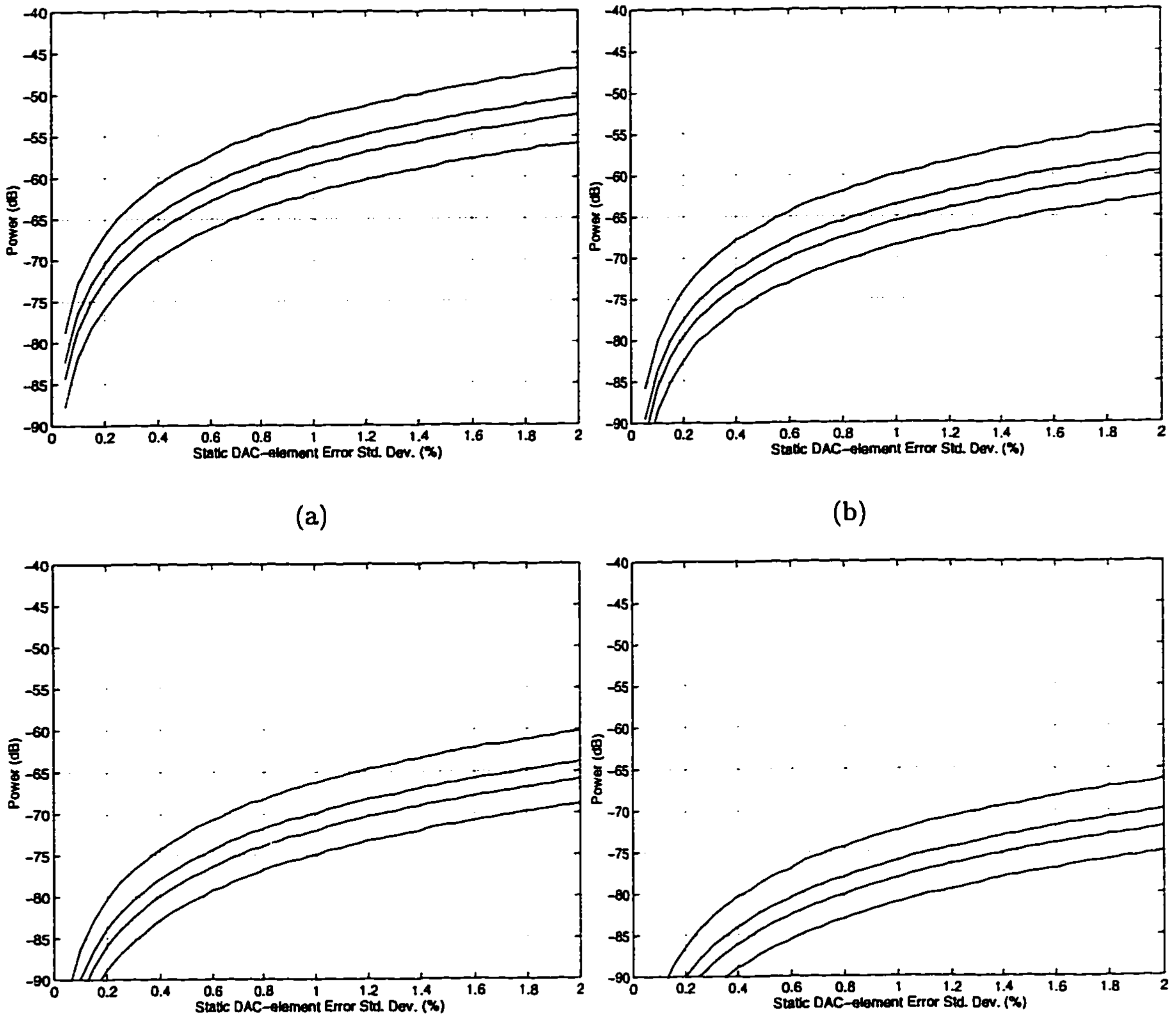
(c)

(d)

Figure 1.8: Plots of $\overline{\sigma}^2$ in dB relative to x_{max}^2 for the same four example 8-bit DACs as in Figure 1.7 when driven by the sum of two sinusoids. In each plot, the amplitude of each sinusoidal component was fixed, and $\overline{\sigma}^2$ computed using (10) is shown as a function of the frequencies of the sinusoidal components.

IV. IC FABRICATION YIELD ESTIMATION

With knowledge of the statistical distribution of the static DAC-element errors, an IC fabrication yield estimate of the proposed DAC architecture with full randomization DEM can be performed using (10). IC fabrication yield estimation data provides a means by which to estimate the percentage of fabricated DACs that will result in a value of $\overline{\sigma}^2$ less than any value of interest. The IC fabrication



(a)

(b)

(c)

(d)

Figure 1.9: IC fabrication yield estimation data for (a) 4-bit, (b) 6-bit, (c) 8-bit, and (d) 10-bit versions of the proposed DAC. The static DAC-element errors were chosen from a normal distribution with a standard deviation ranging from 0.05% to 2%. In each case, from top to bottom, the curves show the largest of the smallest 95%, 65%, 35%, and 5% of $\overline{\sigma}^2$ values in dB relative to x^2_{max}, respectively, when driving the DAC by a sinusoidal input.

yield estimation procedure used in the following was first introduced in [4] and is based upon the idea of computing a large number of samples of $\overline{\sigma}^2$ for a given level of static DAC-element errors, thereby generating data that closely resemble the corresponding statistical distribution of $\overline{\sigma}^2$.

For example, Figures 1.9a-d show IC fabrication yield estimation data corresponding to 4-bit, 6-bit, 8-bit, and 10-bit DACs, respectively. In each case, from top to bottom, the curves show the largest of the smallest 95%, 65%, 35%, and 5% of $\overline{\sigma}^2$ values in dB relative to x_{max}^2, respectively, when driving the DAC by a maximum-amplitude, DC offset sinusoidal input of frequency $\omega_0 = \frac{625}{2048}$. Each figure shows $\overline{\sigma}^2$ versus increasing standard deviation of the static DAC-element errors, and each yield estimation is based upon 5000 calculated values. The static DAC-element errors were chosen as samples of independent, normally distributed random variables with a standard deviation ranging from 0.05% to 2%. This particular choice of static DAC-element errors was made for demonstration purposes only; any other distribution could have been used without changing the yield estimation procedure.

For example, with a standard deviation of 0.5%, the data of Figure 1.9c predicts that 95% of all 8-bit DACs will satisfy $\overline{\sigma}^2 \leq -72$ dB relative to x_{max}^2 and that 5% will satisfy $\overline{\sigma}^2 \leq -81$ dB relative to x_{max}^2. Thus, 90% of all 8-bit DACs fabricated satisfy

$$-81 \text{ dB} \leq \overline{\sigma}^2 \leq -72 \text{ dB}$$

relative to x_{max}^2. This conclusion is supported by the data of the simulated example 8-bit DAC of Figure 1.6, for which $\overline{\sigma}^2 = -75.44$ dB relative to x_{max}^2, and by the four example 8-bit DACs of Figure 1.7, for which $\overline{\sigma}^2$ equals -75.53 dB, -78.35 dB, -73.91 dB, and -75.57 dB, respectively, all relative to x_{max}^2.

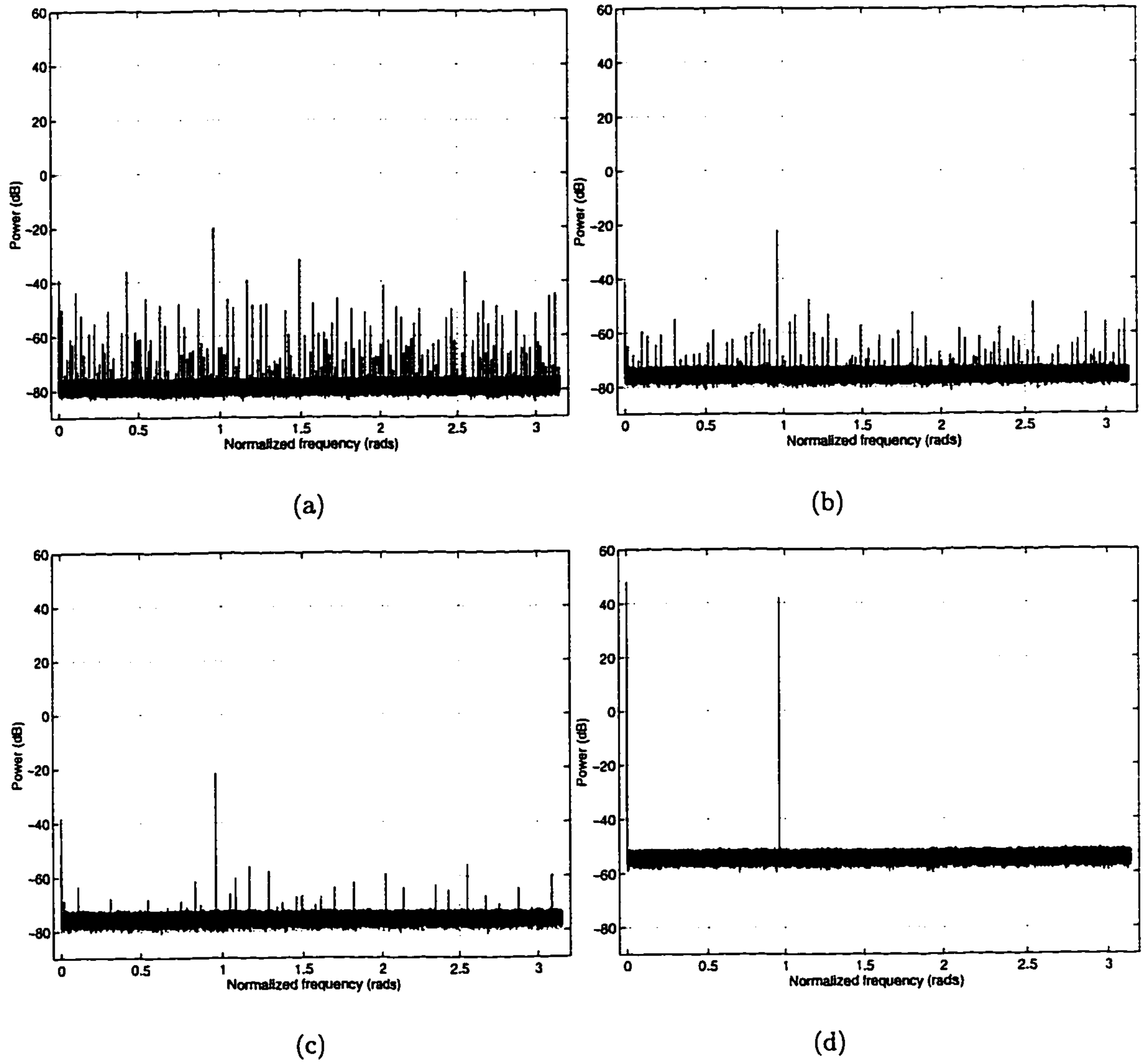
As mentioned previously, when driving the DAC by a sinusoidal input, $\overline{\sigma}^2$ depends on both amplitude and frequency. With a strong dependency, this property would limit the usefulness of the IC fabrication yield estimation technique in that the resulting data only would be applicable to DACs driven by a particular sinusoid. However, as was demonstrated in Figure 1.7, $\overline{\sigma}^2$ is largely independent of sinusoidal frequency, and repeating the yield estimate calculations with maximum-amplitude sinusoidal inputs of several different frequencies gives results very close to the data presented in Figure 1.9. As was also demonstrated with the data in Figure 1.7, no clear trend in $\overline{\sigma}^2$ as a function of sinusoid amplitude is observed, and repeating the yield estimate calculations with sinusoids of different amplitudes gives results very close to the data in Figure 1.9. Consequently, when computing a large number of values of $\overline{\sigma}^2$, the spread of $\overline{\sigma}^2$ caused by varying amplitude is largely absorbed in the spread of $\overline{\sigma}^2$ caused by varying random mismatch errors. It follows that Figure 1.9 represents IC fabrication yield estimation data valid for sinusoidal inputs of any amplitude and frequency.

V. PERFORMANCE DETAILS FOR PARTIAL RANDOMIZATION DEM

In practice, a number of factors other than the static DAC-element errors limit the SFDR achievable by the DAC. Non-ideal circuit behavior such as clock-skew, clock coupling, and finite slew-rates inevitably contributes to harmonic distortion of the DAC output. Thus, the total amount of harmonic distortion of the DAC can be viewed as the effects of two components, namely a component caused by the static DAC-element errors and a component caused by all other non-ideal circuit behavior. To the extent that the latter component is below—or can be attenuated to—the maximum allowable level of harmonic distortion for a given DAC application, a better "engineering solution" to mitigate the effects of the static DAC-element errors might be to merely attenuate the resulting harmonic distortion to the maximum allowable level, thereby possibly reducing the hardware requirement of the DEM technique. The simulation results presented in the following indicate that partial randomization DEM indeed offers such an option.

A. Simulation Results

Simulation results for partial randomization DEM is shown in Figure 1.10. In



(a)

(b)

(c)

(d)

Figure 1.10: Simulated PSDs relative to x_{max}^2 of an example 8-bit DAC with partial randomization DEM. The plots show (a) $y[n] - x[n]$ with random switching in layer 8, (b) $y[n] - x[n]$ with random switching in layers 7 and 8, (c) $y[n] - x[n]$ with random switching in layers 6 through 8, and (d) $y[n]$ with random switching in layers 6 through 8.

particular, Figures 1.10a-c correspond to the signal $y[n] - x[n]$ with random switching in layer 8, layers 7 and 8, and layers 6 through 8, respectively. Figure 1.10d corresponds to $y[n]$ with random switching in layers 6 through 8. In all cases, the DAC input and static DAC-element errors were identical to those used for the full randomization DEM example of Figure 1.6. The simulation results indicates that the harmonic distortion is gradually attenuated as the number of layers with random switching is increased. The maximum-amplitude spurs of Figures 1.10a-c have powers −73.55 dB, −89.47 dB, and −97.49 dB, respectively, relative to the power of the desired sinusoidal signal of frequency ω_0. Several simulations using other sinusoid frequencies and amplitudes similarly support these findings. Consequently, the SFDRs provided by the DAC are 73.55 dB, 89.47 dB, and 97.49 dB, respectively. The parameters used to compute the PSDs were identical to the parameters used to compute the PSDs of Figure 1.6.

To summarize the simulation results of Figure 1.10, partial randomization DEM increasingly suppresses harmonic distortion as the number of layers with random switching is increased, and may suffice to provide the necessary dynamic range for a given application. For example, three layers of random switching would suffice to provide the desired minimum 90 dB of SFDR for an 8-bit DAC applicable to the DDS system mentioned in the Introduction. As quantified in the next section, the hardware complexity of the digital encoder is greatly reduced with partial randomization DEM.

Additional research is needed to theoretically quantify the performance of partial randomization DEM. Among the goals for such research would be a determination of a guaranteed minimum SFDR given a specific degree of randomization.

VI. HARDWARE COMPLEXITY OF THE DIGITAL ENCODER

The hardware complexity of the digital encoder is a function of both the required number of binary switches and the required number of random control bits. As will be shown in the following, the proposed architecture has much lower hardware complexity than the prior art.

DAC Bits	6	7	8	9	10	11	12
Digital Encoder Presented in [2]	192 / 192	448 / 448	1024 / 1024	2304 / 2304	5120 / 5120	11264/11264	24576/24576
Proposed Digital Encoder	120 / 6	247 / 7	502 / 8	1013 / 9	2036 / 10	4083 / 11	8178 / 12

Figure 1.11: Digital hardware required to implement the digital encoders presented in [2] and of the proposed architecture versus the DAC bit-resolution. The table entries are given as pairs (x/y), where x is the number of binary switches and y is the number of random control bits.

A. Full Randomization DEM

To determine the number of required binary switches, recall that the switching block $S_{k,r}$ requires k binary switches. From this, it can be shown that the total number of binary switches required by the digital encoder of a b-bit DAC is $2^{b+1} - b - 2$. It can furthermore be shown that the numbers of required binary switches of the digital encoders presented in [2] and [4] are $b2^{b-1}$ and $(b - \frac{1}{2})2^b$, respectively. The number of random control bits required for the proposed digital encoder is simply b, whereas the digital decoders in [2] and [4] require the same number of random control bits as binary switches.

It follows that the number of required random control bits has been reduced *exponentially* in b compared to the prior art, and that the number of required binary switches has been reduced *linearly* in b. As an example, an 8-bit DAC with the digital encoder architecture presented in [2] requires approximately twice as many binary switches as the proposed architecture, whereas 128 times as many random control bits are required.

A detailed comparison of the hardware complexity of moderate-resolution DACs is shown in Figure 1.11. The figure shows the hardware complexity of the digital encoder presented in [2] and the proposed architecture for bit-resolutions 6 through 12. The table entries are given as pairs (x/y), where x is the number of binary switches and y is the number of random control bits, respectively.

Layers with Random Switching	8	7–8	6–8	5–8	4–8	3–8	2–8
Hardware Complexity	8 / 1	22 / 2	46 / 3	86 / 4	150 / 5	246 / 6	374 / 7

Figure 1.12: Hardware complexity versus the range of layers with random switching of an example 8-bit digital encoder. The table entries are given as pairs (x/y), where x is the number of binary switches and y is the number of random control bits.

B. Partial Randomization DEM

As discussed previously, very low hardware complexity is achievable with partial randomization DEM. To obtain a precise count of the hardware requirement, suppose that the digital encoder implements random switching in layers R through b. It can be shown that the number of required binary switches is $(R+1)2^{b-R+1}-b-2$. The required number of random control bits is simply $b-R+1$ (i.e., the number of layers with random switching).

As an example, it follows that the 8-bit DAC with random switching in layers 6 through 8 requires $7 \times 8 - 8 - 2 = 46$ binary switches and merely 3 random control bits. This should be compared to the requirement of 502 binary switches and 8 random control bits for full randomization DEM and the requirement of 1024 binary switches and 1024 random control bits for the digital encoder in [2]. To further illustrate the reduction of hardware complexity when using partial randomization DEM, the hardware complexity of an 8-bit example DAC versus the range of layers with random switching is tabulated in Figure 1.12.

Finally, it should be mentioned that the reduction in hardware complexity obtained with the proposed digital encoder architecture also yields a major simplification in VLSI layout; generating and routing 1024 random control bits is significantly more difficult than generating and routing 8 or fewer random control bits.

VII. CONCLUSION

A new hardware-efficient dynamic element matching DAC architecture appropriate for DDS has been presented. The proposed architecture is significantly more hardware efficient than the prior art, yet provides similar performance with respect to suppression of harmonic distortion.

For full randomization DEM, quantitative results giving the power of the white conversion noise have been stated and proven, and yield estimates have been presented for selected bit-resolutions and VLSI process statistics. Computer simulation results have been presented that fully support the theoretical results for an example 8-bit DAC applicable to a certain DDS system.

Simulation results show that harmonic distortion is greatly suppressed with partial randomization DEM, which offers a considerable additional reduction in hardware complexity. It has been shown that for an example 8-bit DAC with partial randomization DEM, merely 3 layers of random switching suffice to provide greater than 90 dB of SFDR, as desired for the DDS application in question. Additional research is needed to theoretically quantify the performance of partial randomization DEM. Of particular interest would be the determination of a guaranteed minimum SFDR given a specific degree of randomization.

Non-ideal circuit behavior such as clock-skew, clock-coupling, and finite slew rates inevitably contributes to harmonic distortion of the DAC output. Such non-ideal circuit behavior is typically quite implementation dependent, and research to quantify and mitigate its effects must likely be performed on a case-by-case basis. Nevertheless, the results presented in this paper are still applicable is such situations. In particular, partial randomization DEM promises to offer the option of reducing the hardware complexity of the DEM technique to a minimum while still attenuating harmonic distortion resulting from static DAC-element errors below the level of inevitable harmonic distortion.

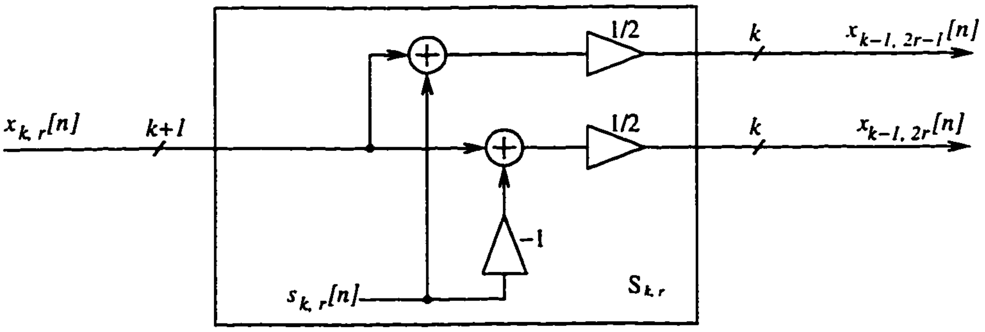


Figure 1.13: The signal processing equivalent of the switching block $S_{k,r}$.

APPENDIX 1.A

The purpose of this appendix is to verify that the output $y[n]$ of the proposed DAC architecture with full randomization DEM or partial randomization DEM is of the general form stated in [5], which will be repeated shortly for convenience. Then, in appendix 1.B, the general form of the DAC noise $e[n]$ given here is re-written to the form stated in Result 1 of Section III, and an expression for the variance of $e[n]$ is derived.

Before stating the general form of the DAC output as derived in [5], a few definitions are first presented. The DACs considered in [5] have switching blocks that perform the signal processing operations depicted in Figure 1.13, and as is shown below, the switching blocks of the DAC architecture proposed in this paper can also be viewed as shown in Figure 1.13. The $k+1$-bit input of $S_{k,r}$ is denoted $x_{k,r}[n]$, and the two k-bit outputs are denoted $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$, respectively. The i^{th} bit of $x_{k,r}[n]$ is denoted $x_{k,r}^{(i)}[n]$. The sequence $s_{k,r}[n]$ is generated within the switching block, and, as can be verified from the figure,

$$s_{k,r}[n] = x_{k-1,2r-1}[n] - x_{k-1,2r}[n]. \tag{13}$$

The results in [5] giving the general form of the DAC output can now be stated as follows:

Claim A. *The output $y[n]$ of a b-bit version of the proposed DAC architecture* *with full randomization DEM or partial randomization DEM is of the form*

$$y[n] = \alpha x[n] + \beta + e[n], \tag{14}$$

where

$$\alpha = 1 + \frac{1}{2^b}\sum_{i=1}^{2^b}(e_{h_i} - e_{l_i}), \tag{15}$$

$$\beta = \sum_{i=1}^{2^b} e_{l_i}, \tag{16}$$

and

$$e[n] = \sum_{k=1}^{b}\sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n]. \tag{17}$$

In (17),

$$\Delta_{k,r} = \frac{1}{2^k}\sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} \left[(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})\right] \tag{18}$$

and $s_{k,r}[n]$ *is defined by* (13). *These results do not depend upon any particular form or statistical properties of the static DAC-element errors.*

Proof. As shown in [5], to prove the above claim for the DAC with full randomization DEM, it suffices to verify

$$x_{b,1}[n] = x[n], \tag{19}$$

$$s_{k,r}[n] = \begin{cases} \text{even if } x_{k,r}[n] \text{ is even;} \\ \text{odd if } x_{k,r}[n] \text{ is odd;} \end{cases} \tag{20}$$

and

$$|s_{k,r}[n]| \leq \min\{x_{k,r}[n], 2^k - x_{k,r}[n]\}. \tag{21}$$

To accomplish this, a definition of the numerical value of $x_{k,r}[n]$ is needed.

Definition. *The numerical value of the input and outputs of the switching block proposed in this paper must be interpreted according to*

$$x_{k,r}[n] = \sum_{i=1}^{k} 2^{i-1} x_{k,r}^{(i)}[n] + x_{k,r}^{(0)}[n]. \tag{22}$$

Thus, $x_{k,r}[n]$ is the sum of a conventional k-bit unsigned binary number and an "extra LSB", $x_{k,r}^{(0)}[n]$.

First, to verify (19), recall that the input to the switching block $S_{b,1}$ was defined in Section II according to

$$x_{b,1}^{(0)}[n] = 0 \qquad \text{and} \qquad x_{b,1}^{(i)}[n] = x^{(i)}[n], \quad 1 \le i \le b.$$

Inserting this in (22) yields (19).

Next, it will be shown that the switching blocks presented in this paper perform signal processing according to Figure 1.13 such that

$$s_{k,r}[n] = \begin{cases} d_{k,r}[n], & \text{if } c_k[n] = 0; \\ -d_{k,r}[n], & \text{if } c_k[n] = 1; \end{cases} \tag{23}$$

where

$$d_{k,r}[n] = x_{k,r}[n] - 2^k x_{k,r}^{(k)}[n], \tag{24}$$

and $c_k[n]$ is the random control bit of the k^{th} layer. Then, (23) and (24) will be used to verify (20) and (21).

Suppose $c_k[n] = 0$. It follows from Figure 1.3, (13), and (22) that

$$s_{k,r}[n] = \sum_{i=1}^{k-1} 2^{i-1} x_{k,r}^{(i)}[n] + x_{k,r}^{(0)}[n] - \left(\sum_{i=1}^{k-1} 2^{i-1} + 1 \right) x_{k,r}^{(k)}[n]. \tag{25}$$

Collecting and rearranging terms using (22) yields $s_{k,r}[n] = d_{k,r}[n]$. Similarly, it follows that $s_{k,r}[n] = -d_{k,r}[n]$ when $c_k[n] = 1$. This verifies (23) and (24).

To verify (20), notice that the term $2^k x_{k,r}^{(k)}[n]$ in (24) is even because k is a positive integer. Thus, if $x_{k,r}[n]$ is even, $d_{k,r}[n]$ is even, and if $x_{k,r}[n]$ is odd, $d_{k,r}[n]$ is odd.

To verify (21), notice that if $x_{k,r}^{(k)}[n] = 1$, (22) implies $2^{k-1} \le x_{k,r}[n] \le 2^k$. Thus, from (23) and (24),

$$\left| s_{k,r}[n] \right| = 2^k - x_{k,r}[n] \le x_{k,r}[n].$$

Similarly, if $x_{k,r}^{(k)}[n] = 0$, (22) implies $0 \leq x_{k,r}[n] \leq 2^{k-1}$, and thus

$$\left|s_{k,r}[n]\right| = x_{k,r}[n] \leq 2^k - x_{k,r}[n],$$

which verifies Claim A for full randomization DEM.

The digital encoder with partial randomization DEM employing random switching in layers R through b is equivalent to the digital encoder of full randomization DEM for which $c_k[n] = 1$, $k = 1, 2, \ldots, R-1$. Thus, it follows that Claim A also holds for partial randomization DEM. ■

APPENDIX 1.B

The purpose of this appendix is to verify that the DAC noise $e[n]$ has the form stated in Result 1 of Section III, and to provide an expression for the variance of $e[n]$.

Claim B1. *For a b-bit version of the DAC architecture with full randomization DEM, the DAC noise is a zero-mean, white random process of the form*

$$e[n] = \sum_{i=1}^{b} w_i[n] x^{(i)}[n], \tag{26}$$

where each $w_i[n]$ is a zero-mean, white random process of the form

$$w_i[n] = w_{i,h_i[n]}. \tag{27}$$

In (27),

$$w_{i,j} = \sum_{m=(j-1)2^{i-1}+1}^{j2^{i-1}} (e_{h_m} - e_{l_m}) - 2^{i-1}(\alpha - 1) \tag{28}$$

and

$$h_i[n] = \sum_{j=1}^{b-i} 2^j c_{i+j}[n] + (1 - c_i[n]) + 1. \tag{29}$$

The above results do not depend upon any particular form or statistical properties of the static DAC-element errors.

Proof. By virtue of Claims A and B1, it is sufficient to show that (17) is equivalent to (26), which will be accomplished by induction. First, notice that substituting (22) into (23) and (24) with $k = b$ and using $x_{b,1}[n] = 0$ gives

$$s_{b,1}[n] = (-1)^{(1-c_b[n])}(2^{b-1}x^{(b)}[n] - \sum_{i=1}^{b-1} 2^{i-1}x^{(i)}[n]),$$

which, combined with (18), yields

$$\Delta_{b,1}s_{b,1}[n] = \left[\frac{1}{2^{b-1}} \sum_{m=(1-c_b[n])2^{b-1}+1}^{(1-c_b[n])2^{b-1}+2^{b-1}} (e_{h_m} - e_{l_m} - \alpha + 1)\right]\left[2^{b-1}x^{(b)}[n] - \sum_{i=1}^{b-1} 2^{i-1}x^{(i)}[n]\right]. \tag{30}$$

To establish the induction basis, let $b = 1$. Then, from (17),

$$e[n] = \Delta_{1,1}s_{1,1}[n], \tag{31}$$

and inserting (30) yields

$$e[n] = \left[\sum_{m=(1-c_1[n])+1}^{(1-c_1[n])+1} (e_{h_m} - e_{l_m} - \alpha + 1)\right] x^{(1)}[n].$$

This can be written as

$$e[n] = w_1[n]x^{(1)}[n] \tag{32}$$

where

$$w_1[n] = w_{1,h_1[n]} \qquad \text{and} \qquad h_1[n] = (1 - c_1[n]) + 1,$$

so $e[n]$ has the form of (26) for $b = 1$. Since $c_1[n]$ is a white random process with possible values 0 and 1, it follows that $h_1[n]$ is a white random process with possible values 1 and 2. But $h_1[n]$ determines the value of $w_1[n]$ to be either $w_{1,1}$ or $w_{1,2}$, and thus $w_1[n]$ is a white random process. It follows from (32) that $e[n]$ is a white random process. Furthermore, (23) and (31) show that $e[n]$ is zero-mean, and it follows from (32) that $w_1[n]$ is zero-mean.

Next, suppose the claim holds for $1 \leq b \leq b'$. It will be shown that the claim holds for $b' + 1$. Notice that (17) may be written as

$$e[n] = \Delta_{b'+1,1} s_{b'+1,1}[n] + e_1[n] + e_2[n], \tag{33}$$

where

$$e_1[n] = \sum_{k=1}^{b'} \sum_{r=1}^{2^{b'-k}} \Delta_{k,r} s_{k,r}[n], \tag{34}$$

and

$$e_2[n] = \sum_{k=1}^{b'} \sum_{r=1}^{2^{b'-k}} \Delta_{k,r+2^{b'-k}} s_{k,r+2^{b'-k}}[n]. \tag{35}$$

It will next be argued that $e_1[n] = 0$ if $c_{b'+1}[n] = 1$, and that $e_2[n] = 0$ if $c_{b'+1}[n] = 0$. Suppose $c_{b'+1}[n] = 1$. Since $x^{(b'+1)}[n]$ is either 0 or 1, it follows from Figure 1.3 and (22) that $x_{b',1}[n]$ is either 0 or $2^{b'}$. Consequently, all the $x_{k,r}[n], 1 \leq k \leq b', 1 \leq r \leq 2^{b'-k}$, are either 0 or 2^k. Thus, by (24) and (34), $e_1[n] = 0$. Similar reasoning verifies that $e_2[n] = 0$ if $c_{b'+1}[n] = 0$.

Using (30) and invoking the induction hypothesis, (33) may be written as

$$e[n] =$$
$$\left[\sum_{m=(1-c_{b'+1}[n])2^{b'}+1}^{(1-c_{b'+1}[n])2^{b'}+2^{b'}} (e_{h_m} - e_{l_m} - \alpha + 1) \right] \left[x^{(b'+1)}[n] - \sum_{i=1}^{b'} \frac{2^{i-1}}{2^{b'}} x^{(i)}[n] \right] + \sum_{i=1}^{b'} w_i[n] x^{(i)}[n]. \tag{36}$$

From (33), it follows that

$$\sum_{i=1}^{b'} w_i[n] x^{(i)}[n] \;=\; e_1[n] + e_2[n] \;=\; \begin{cases} e_1[n], & \text{if } c_{b'+1}[n] = 0; \\ e_2[n], & \text{if } c_{b'+1}[n] = 1, \end{cases}$$

where $w_i[n]$ is calculated from (27) with $b = b'$. The desired result can now be shown by comparing the coefficients of $x^{(i)}[n]$, $1 \leq i \leq b' + 1$, in (36) with the coefficients determined from (26) with $b = b' + 1$. The coefficient of $x^{(b'+1)}[n]$ in (36) is

$$\sum_{m=(1-c_{b'+1}[n])2^{b'}+1}^{(1-c_{b'+1}[n])2^{b'}+2^{b'}} (e_{h_m} - e_{l_m} - \alpha + 1), \tag{37}$$

which equals $w_{b'+1}[n]$, as asserted. Next, suppose $c_{b'+1}[n] = 0$. The coefficient of $x^{(i)}[n]$, $1 \leq i \leq b'$, in (36) is then

$$w_i[n] - \frac{2^{i-1}}{2^{b'}} \sum_{m=2^{b'}+1}^{2^{b'+1}} (e_{h_m} - e_{l_m} - \alpha + 1).$$

Inserting the definition of $w_i[n]$ with $b = b'$ and rearranging terms yields

$$\sum_{m=(h_i[n]-1)2^{i-1}+1}^{h_i[n]2^{i-1}} (e_{h_m} - e_{l_m} - \alpha + 1),$$

which equals $w_i[n]$ with $b = b' + 1$. For $c_{b'+1}[n] = 1$, it can similarly be shown that the coefficient of $x^{(i)}[n]$ in (36) is

$$\sum_{m=(2^{b'}+h_i[n]-1)2^{i-1}+1}^{(2^{b'}+h_i[n])2^{i-1}} (e_{h_m} - e_{l_m} - \alpha + 1),$$

as asserted. It follows from (37) that $w_{b'+1}[n]$ is a white random process. Thus, by the induction hypothesis, $e[n]$ is a white random process. Also by the induction hypothesis, $e_1[n]$ and $e_2[n]$ in (33) are each zero-mean. It then follows from (23) and (24) that $e[n]$ and $w_{b'+1}[n]$ are zero-mean random processes. ■

Claim B2. *The variance of* $e[n]$ *is given by*

$$\mathrm{Var}\{e[n]\} = \sum_{i=1}^{b} \gamma_i\, x^{(i)}[n] + \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \gamma_{i,j}\, x^{(i)}[n] x^{(j)}[n],$$

where

$$\gamma_i = \frac{1}{2^{b-i+1}} \sum_{j=1}^{2^{b-i+1}} (w_{i,j})^2 \tag{38}$$

and

$$\gamma_{i,j} = \frac{1}{2^{b-j-1}} \sum_{k=1}^{2^{b-i}} w_{i,2k-1} w_{i,2k}. \tag{39}$$

Proof. Since $e[n]$ is zero-mean, $\text{Var}\{e[n]\} = \text{E}\{(e[n])^2\}$, where $\text{E}\{\cdot\}$ denotes the statistical expectation operator. Using (26) and rearranging terms results in

$$\text{Var}\{e[n]\} = \sum_{i=1}^{b} \text{E}\left\{(w_i[n])^2\right\} x^{(i)}[n] + 2\sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \text{E}\left\{w_i[n]w_j[n]\right\} x^{(i)}[n]x^{(j)}[n], \tag{40}$$

where use was made of the equality $(x^{(i)}[n])^2 = x^{(i)}[n]$. To evaluate (40), consider first $\text{E}\left\{(w_i[n])^2\right\}$. By the definition in (29), $h_i[n]$ can be viewed as the value associated with a $(b-i+1)$-bit binary number, offset by 1, where the values of the bits are determined by $c_k[n]$, $k = i, i+1, \ldots, b$. The two possible values of $c_k[n]$ are equally probable and $c_k[n]$ is independent of $c_j[n]$, $k \neq j$, so it follows that the 2^{b-i+1} different values of $h_i[n]$ are equally probable. It then follows from (27) that the 2^{b-i+1} different values of $w_i[n]$ are equally probable, i.e., $w_i[n] = w_{i,j}$ with probability $\frac{1}{2^{b-i+1}}$. Thus,

$$\text{E}\left\{(w_i[n])^2\right\} = \frac{1}{2^{b-i+1}} \sum_{j=1}^{2^{b-i+1}} (w_{i,j})^2, \tag{41}$$

as asserted.

It will next be shown that

$$\text{E}\left\{w_i[n]w_j[n]\right\} = \frac{1}{2^{b-j}} \sum_{k=1}^{2^{b-i}} w_{i,2k-1} w_{i,2k}. \tag{42}$$

As derived in the above, $w_i[n]$ depends on $c_k[n]$, $k = i, i+1, \ldots, b$, and $w_j[n]$ depends on $c_k[n]$, $k = j, j+1, \ldots, b$. Thus, it follows that $w_j[n]$ assumes one of 2^{i-j} equally probable values depending upon the value of $w_i[n]$. Specifically, for a given value of $w_i[n]$, (27) may be rewritten as

$$w_i[n] = w_{i,2k-1+(1-r)}; \qquad k \in \left\{1, 2, \ldots, 2^{b-i}\right\}, \qquad r \in \{0, 1\}, \tag{43}$$

where r is the value of $c_i[n]$. As can be verified from (29), the possible values of $w_j[n]$ can then be specified in terms of k, r, and an integer parameter m according to

$$w_j[n] \in \left\{ w_{j,2^{i-j}(2k-2+r)+m} \; : m = 1, 2, \ldots, 2^{i-j} \right\}, \tag{44}$$

where each value occurs with probability $\frac{1}{2^{i-j}}$. Combining (43) and (44) yields

$$\mathrm{E}\left\{w_i[n]w_j[n]\right\} = \frac{1}{2^{b-i+1}} \frac{1}{2^{i-j}} \sum_{r=0}^{1} \left(\sum_{k=1}^{2^{b-i}} w_{i,2k-1+(1-r)} \left(\sum_{m=1}^{2^{i-j}} w_{j,2^{i-j}(2k-2+r)+m} \right) \right). \tag{45}$$

To proceed with the verification of (42), it will be shown in the following that

$$\sum_{m=1}^{2^{i-j}} w_{j,2^{i-j}(2k-2+r)+m} = w_{i,2k-1+r}. \tag{46}$$

From (28) it follows that (46) can be verified by establishing the appropriate limits for the summation of the terms $(e_{h_m} - e_{l_m} - \alpha + 1)$. The lower summation limit on the left-hand side of (46) can easily be found to be

$$[(2k - 1 + r) - 1]2^{i-1} + 1. \tag{47}$$

Similarly, the upper summation limit on the left-hand side of (46) can be found to be

$$(2k - 1 + r)2^{i-1}. \tag{48}$$

Then (46) follows from (28) using (47) and (48). Furthermore,

$$\sum_{r=0}^{1} w_{i,2k-1+(1-r)} w_{i,2k-1+r} = 2w_{i,2k-1}w_{i,2k},$$

and (45) reduces to (42). Claim B2 follows from (40), (41), (42), and the definitions of γ_i and $\gamma_{i,j}$. ■

APPENDIX 1.C

The purpose of this appendix is to verify the time-average properties of the DAC output with full randomization DEM as stated in Section III by Result 2, given here in its complete form as:

Claim C. *If $\overline{M}_x$ and $\overline{R}_{xx}[k]$ exist, then*

$$\overline{M}_y = \alpha\overline{M}_x + \beta \tag{49}$$

and

$$\overline{R}_{yy}[k] = \alpha^2\overline{R}_{xx}[k] + \overline{\eta} + \overline{\sigma}^2\delta[k], \tag{50}$$

with probability 1, where α is given by (15), *β is given by* (16),

$$\overline{\eta} = 2\alpha\beta\overline{M}_x + \beta^2, \tag{51}$$

and

$$\overline{\sigma}^2 = \sum_{i=1}^{b} \gamma_i \overline{M}_{x(i)} + \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \gamma_{i,j} \overline{M}_{x(i,j)}. \tag{52}$$

In (52), *γ_i and $\gamma_{i,j}$ are given by* (38) *and* (39), *respectively.*

Proof. From Result 1 it follows that

$$\mathrm{E}\left\{y[n]\right\} = \alpha x[n] + \beta,$$

and consequently,

$$\lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} \mathrm{E}\left\{y[n]\right\} = \alpha\overline{M}_x + \beta.$$

To deduce that (49) holds with probability 1, it suffices to show that $y[n]$ obeys the strong law of large numbers. By the Kolmogorov Criterion, it suffices to show that $y[n]$ has finite variance. This follows immediately because $x[n]$ and the static mismatch errors are bounded.

To verify (50), consider first the statistical autocorrelation of $y[n]$ defined as $R_{yy}[n,k] = \mathrm{E}\left\{y[n]y[n+k]\right\}$. From Result 1 it follows that

$$R_{yy}[n,k] = \mathrm{E}\left\{(\alpha x[n] + \beta + e[n])\,(\alpha x[n+k] + \beta + e[n+k])\right\}.$$

Expanding, collecting terms, and making use of the facts that e$[n]$ is a zero-mean, white random process and $x[n]$ is deterministic, results in

$$R_{yy}[n,k] = \alpha^2 x[n]x[n+k] + \alpha\beta(x[n] + x[n+k]) + \beta^2 + \sigma^2[n]\delta[k],$$

where

$$\sigma^2[n] = \mathrm{E}\left\{e^2[n]\right\} = \mathrm{Var}\left\{e[n]\right\}.$$

Then using Claim B and the definitions of $\overline{\eta}$ and $\overline{\sigma}^2$,

$$\lim_{P\to\infty} \frac{1}{P}\sum_{n=1}^{P} R_{yy}[n,k] = \alpha^2\overline{R}_{xx}[k] + \overline{\eta} + \overline{\sigma}^2\delta[k].$$

An argument identical to that presented for the corresponding result in [4] establishes (50) with probability 1. ■

REFERENCES

1. M. J. Flanagan, G. A. Zimmerman, "Spur-reduced digital sinusoid synthesis," *IEEE Trans. on Communications*, vol. 43, no. 7, pp. 2254-2262, July 1995.
2. L. R. Carley, "A noise shaping coder topology for 15+ bits converters," *IEEE J. Solid-State Circuits*, vol. SC-24, pp. 267-273, April 1989.
3. L. R. Carley and J. Kenney, "A 16-bit 4'th order noise-shaping D/A converter," *IEEE Proc. CICC*, pp. 21.7.1-21.7.4, 1988.
4. I. Galton, P. Carbone, "A rigorous error analysis of D/A conversion with dynamic element matching," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 763-772, Dec. 1995.

5. I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, to appear.

6. R. M. Gray, T. G. Stockham Jr., "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 805-812, May 1993.

7. R. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters.* Boston, MA: Kluwer Academic Publishers, 1994.

8. A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1989.

CHAPTER ACKNOWLEDGMENT

The text of Chapter 1, in part or in full, consists of the material that will appear as a "Regular Paper" in a future edition of *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing.*

The dissertation author was the primary researcher and the co-author listed in this publication directed and supervised the research which forms the basis for this chapter.

Chapter 2

An Analysis of the Partial Randomization Dynamic Element Matching Technique[†]

Henrik T. Jensen, *Student Member, IEEE,* and Ian Galton, *Member, IEEE* [‡]

***Abstract*—Partial randomization DEM was recently introduced as a promising DEM technique for low harmonic distortion digital-to-analog conversion. The DEM technique is well-suited for applications such as direct digital synthesis in wireless communication systems for which low hardware complexity is essential in addition to low harmonic distortion. Previously reported simulation results demonstrate that partial randomization DEM greatly attenuates harmonic distortion resulting from static errors in the analog output levels of the DAC, while offering considerable savings in hardware compared to other DEM techniques. This paper presents the first quantitative performance analysis of partial randomization DEM. As a main result, the minimum spurious-free dynamic range provided by the digital-to-analog converter has been quantified as a function of its hardware complexity and the analog output level errors.**

I. INTRODUCTION

DIRECT digital synthesis (DDS) has emerged as an efficient and flexible method of generating analog signals of high spectral purity [1]. A key component of a DDS system is the digital-to-analog converter (DAC), which must introduce very little harmonic distortion to honor the spectral purity of the synthesized digital

[†] Under review: IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING

[‡] The authors are with the Department of Electrical and Computer Engineering, UNIVERSITY OF CALIFORNIA, SAN DIEGO, La Jolla, CA 92093-0407, USA; email: htjensen@ece.ucsd.edu and galton@ece.ucsd.edu

signal. Unfortunately, non-ideal circuit behavior in practical DACs inevitably gives rise to *DAC noise* that consists largely of harmonic distortion and ultimately limits the achievable DDS performance.

Dynamic element matching (DEM) has emerged as a means of mitigating the deleterious effects of non-ideal circuit behavior in DACs by essentially causing a large portion of the DAC noise to be broken into white noise instead of harmonic distortion. Partial randomization DEM was recently introduced as a promising DEM technique for low harmonic distortion DACs [2]. As will be shown in this paper, the technique offers a trade of spur suppression for hardware complexity.

The partial randomization DEM DAC incorporates a bank of coarse DACs, referred to as *DAC-elements*, the outputs of which are summed together to yield a composite DAC. Inevitable non-ideal circuit behavior results in analog output errors of the DAC-elements, giving rise to DAC noise. The DAC noise can be viewed as consisting of two components, namely a component caused by the *static* part of the analog output errors, and a component caused by the *dynamic* part of the analog output errors. To the extent that the latter component is below—or can be attenuated to—the maximum allowable level of harmonic distortion for a given DAC application, a good "engineering solution" to mitigate the effects of the static analog output errors is to merely attenuate the resulting harmonic distortion to the maximum level that can be tolerated. Simulations reported in [2] indicate that partial randomization DEM can be used to achieve this result, while offering a significant reduction in hardware complexity over other DEM techniques. However, this finding was supported by simulation results only.

This paper provides a rigorous analysis of the performance of partial randomization DEM. In particular, given a desired minimum SFDR and knowledge of the statistics of the static analog output errors, the analysis can be used to quantify the hardware requirement of the DEM technique for any DAC bit-resolution of interest. As an example, the theoretical results are applied in an IC fabrication yield estimation analysis of the minimum SFDR provided by the 8-bit version of the DAC.

The remainder of the paper is divided into sections as follows. Section II reviews the proposed DAC architecture. Section III provides performance details of partial randomization DEM with simulation results and performance equations. Section IV presents the IC fabrication yield estimation technique for partial randomization DEM based upon the results of the theoretical analysis. The appendices provide derivations of the theoretical results stated in Section III.

II. ARCHITECTURE

To review the architecture of the proposed DAC, consider the example 8-bit version shown in Figure 2.1. The digital input, $x[n]$, is a sequence of unsigned 8-bit numbers less than 256. The DAC consists of 3 layers of digital devices, each referred to as a *switching block* and collectively referred to as the *digital encoder*, followed by an array of 8 *DAC-banks*, each labeled DB_i, and an analog output summing node. The number of layers is referred to as the *randomization index*, I, (i.e., $I = 3$ in Figure 2.1) and the layers are numbered 6 through 8. Each switching block is labeled $S_{k,r}$, where k denotes the layer number and r denotes the position of the switching block in the layer.

Figure 2.2 shows the functional details of the switching block $S_{k,r}$. The switching block has one $k+1$-bit input, two k-bit outputs, and an input for a *random control bit*, $c_k[n]$. The random control bit $c_k[n]$ is common to all the switching blocks in the k^{th} layer, and is ideally a white random bit-sequence, statistically independent of the random control bits applied to the other layers. When $c_k[n]$ is high, the MSB of

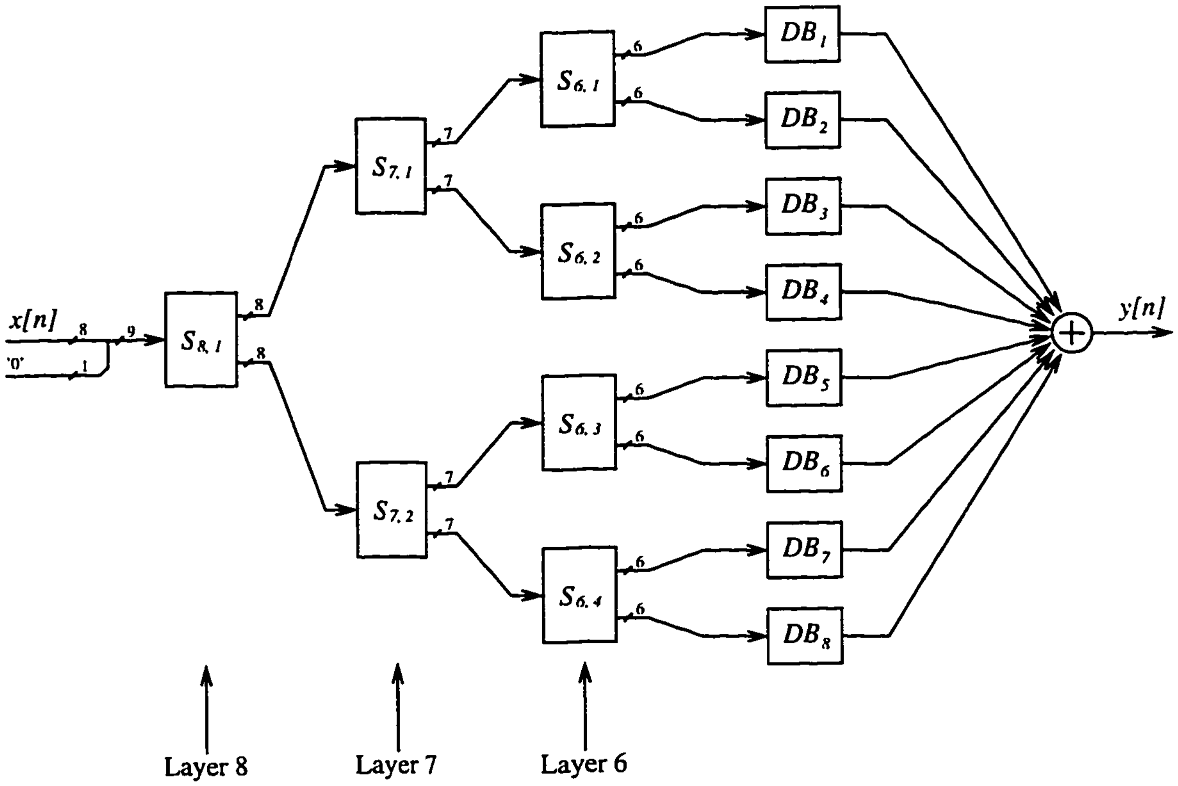


Figure 2.1: An 8-bit version of the proposed DAC architecture with randomization index $I = 3$. The layers are numbered 6 through 8.

the input is mapped to all k bits of the top output, and the remaining k bits of the input are mapped directly to the k bits of the bottom output. When $c_k[n]$ is low, the mappings are interchanged. This process of randomly mapping the input to the outputs is referred to as *random switching*. As indicated in Figure 2.1, the DAC input $x[n]$ is assigned to the $S_{b,1}$ input bits b_1 through b_b, and a zero is assigned to the input bit b_0.

Figure 2.3 shows the functional details of the i^{th} DAC-bank. As shown in Figure 2.3a, the DAC-bank has a 6-bit input, $x_i[n]$, and an analog output, $y_i[n]$. It is functionally equivalent to a 5-bit *conventional DAC* and a 1-bit conventional DAC, with $y_i[n]$ formed as the sum of the outputs of the two conventional DACs. Notice that $x_i[n]$ corresponds to the sequence $x_{5,i}[n]$ with the notation in Figure 2.2, where

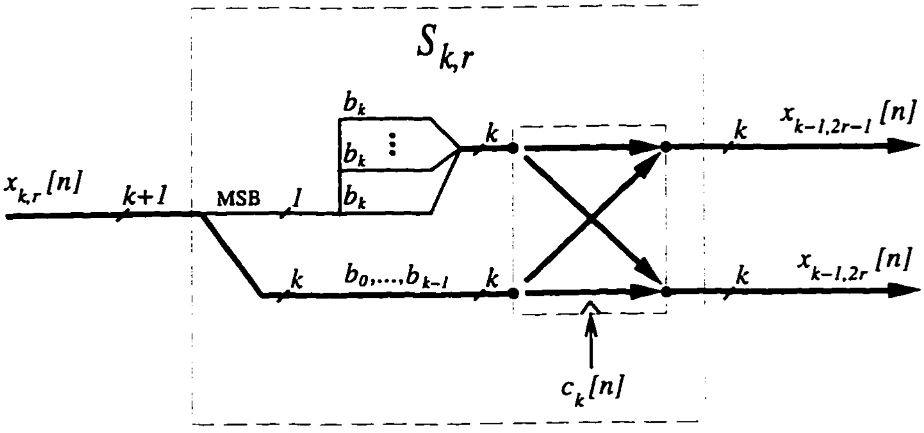


Figure 2.2: Details of the switching block of the k^{th} layer.

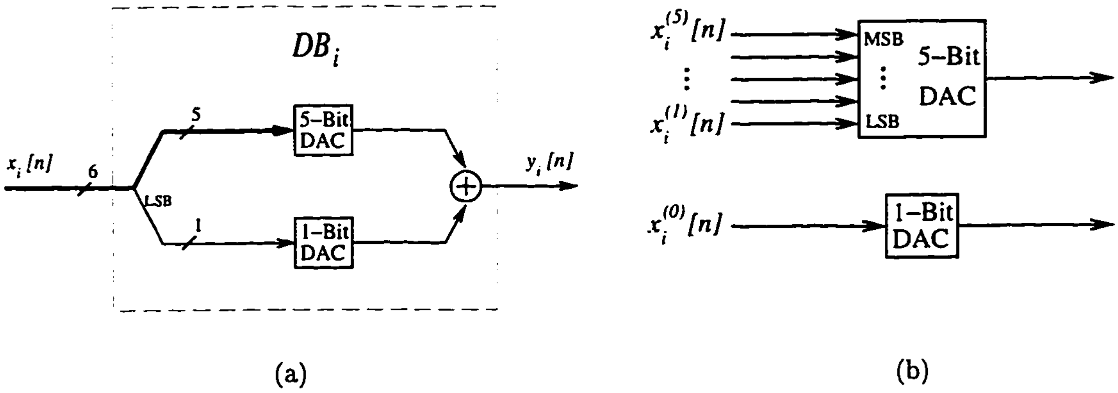


Figure 2.3: Details of the DAC-bank corresponding to the example 8-bit DAC of Figure 2.1. (a) The interconnection of the 5-bit and 1-bit conventional DACs, and (b) the assignment of the input bits.

the subscript "5" has been omitted for convenience. The DAC-bank input $x_i[n]$ is interpreted as a sequence of unsigned integers in the range $0, 1, \ldots, 32$, formed as

$$x_i^{(0)}[n] + \sum_{j=1}^{5} 2^{j-1} x_i^{(j)}[n], \tag{1}$$

where $x_i^{(j)}[n]$ denotes the j^{th} bit of $x_i[n]$. As shown in Figure 2.3b, the input to the 5-bit conventional DAC consists of the five MSBs of $x_i[n]$, and is interpreted as a sequence of unsigned integers in the range $0, 1, \ldots, 31$. The input to the 1-bit conventional DAC is the LSB of $x_i[n]$, and is interpreted as a sequence of numbers

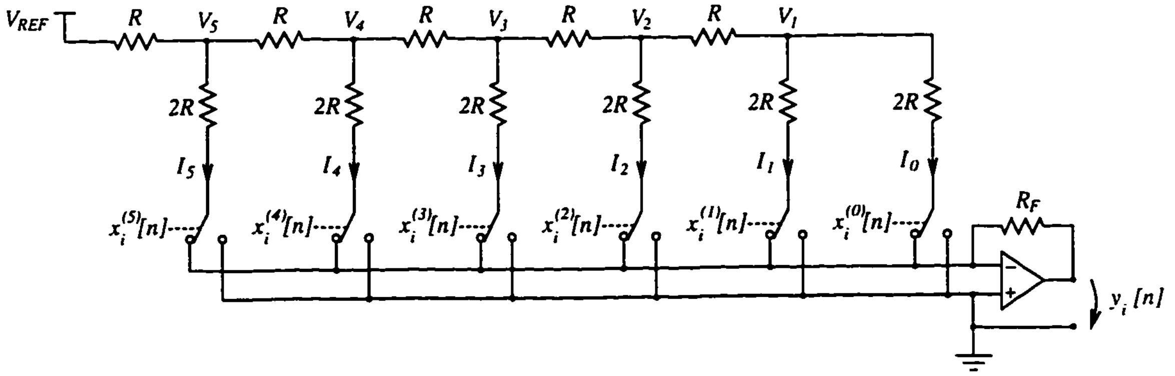


Figure 2.4: An implementation of the DAC-bank using an (ideal) $R - 2R$ ladder network.

that are either 0 or 1.

Figure 2.4 shows an implementation of the DAC-bank using an $R - 2R$ ladder network. Notice that the network has been modified slightly compared to the traditional architecture in that an extra switch has been introduced at the right-most $2R$ resistor to implement the one-bit DAC in Figure 2.3a. For the case of ideal circuit behavior, it is easy to verify that the node voltages, V_j, satisfy

$$V_j = V_j^{ideal} = \frac{V_{REF}}{2^{6-j}}. \tag{2}$$

It follows that the binary weighted currents, I_j, are given by

$$I_j = I_j^{ideal} = \frac{V_j}{2R}. \tag{3}$$

The output voltage, $y_i[n]$, is dependent upon the currents flowing through the feedback resistor, R_F, such that

$$\begin{aligned} y_i[n] = y_i^{ideal}[n] &= R_F \sum_{j=0}^{5} I_j \\ &= \frac{R_F V_{REF}}{R2^6}\left[x_i^{(0)}[n] + \sum_{j=1}^{5} 2^{j-1} x_i^{(j)}[n]\right]. \end{aligned} \tag{4}$$

With appropriate values of V_{REF}, R_F, and R, (4) can be made to equal (1) or a scaled version thereof, if desired. It will shortly be considered how non-ideal circuit behavior affects the performance of the DAC-bank.

The 8-bit version of the proposed DAC with randomization index 3 shown in Figure 2.1 can easily be modified to accommodate other bit-resolutions and/or randomization indices. For example, another layer of switching blocks, denoted layer 5, could be added. In this case, each of the switching blocks in this layer would have 5-bit outputs, and the number of DAC-banks would increase two-fold to 16. The DAC-banks would each consist of a 4-bit conventional DAC and a 1-bit conventional DAC. In general, more layers require more hardware, but—as will be shown—also provide greater suppression of harmonic distortion.

The switching blocks of the digital encoder can be constructed using binary switches [2]. The binary switch is a 2-input × 2-output device that simply either passes the inputs directly through to the outputs or connects the inputs to the outputs in reverse order, depending upon the value of $c_k[n]$. The hardware complexity of the digital encoder is a function of both the required number of binary switches and the required number of random control bits, and was discussed in detail in [2]. For convenience, the formulae dictating the hardware requirements of partial randomization DEM are repeated in the following. For a b-bit version of the DAC with randomization index I, the number of binary switches required is

$$N_{bs} = (b - I + 2)2^I - b - 2, \tag{5}$$

and the number of random control bits is simply

$$N_{cb} = I. \tag{6}$$

Thus, the 8-bit DAC of Figure 2.1 requires 46 binary switches and 3 random control bits.

III. PERFORMANCE DETAILS

During each sample interval, the DAC-bank output $y_i[n]$ ideally equals the analog representation of its digital input $x_i[n]$. With this assumption, it was shown in [2] that interconnecting the switching blocks and DAC-banks in the network of Figure 2.1 results in a DAC for which

$$y[n] = x[n]. \tag{7}$$

However, in practice, the DAC-banks operate such that

$$y_i[n] = x_i[n] + \Delta\{x_i[n]\}, \tag{8}$$

where the $\Delta\{\cdot\}$ are errors associated with the 33 analog output levels that arise from non-ideal circuit performance. As an example of how these errors might arise, consider again an implementation of the DAC-bank using an $R-2R$ ladder network, as shown in Figure 2.5. Non-ideal IC fabrication processing and non-zero ohmic switch resistance result in errors, ΔR_k, of the resistor values and thus perturb the node voltages V_j. Specifically, the node voltages deviate from their ideal values given by (2) as

$$V_j = V_j^{ideal} + \Delta V_j, \tag{9}$$

where the ΔV_j are functions of the resistor errors ΔR_k. As a consequence, the binary weighted currents into the op-amp summing node deviate by some quantity, ΔI_j, from their ideal values given by (3) according to

$$I_j = I_j^{ideal} + \Delta I_j. \tag{10}$$

It follows from (2) through (4), (9), and (10) that

$$\begin{aligned} y_i[n] &= (R_F + \Delta R_F)\left[(I_0^{ideal} + \Delta I_0)\, x_i^{(0)}[n] + \sum_{j=1}^{5}(I_j^{ideal} + \Delta I_j)\, x_i^{(j)}[n]\right] \\ &= y_i^{ideal}[n] + \sum_{j=0}^{5} \Delta_j\, x_i^{(j)}[n], \end{aligned}$$

where

$$\Delta_j = \Delta I_j R_F + \Delta R_F I_j. \tag{11}$$

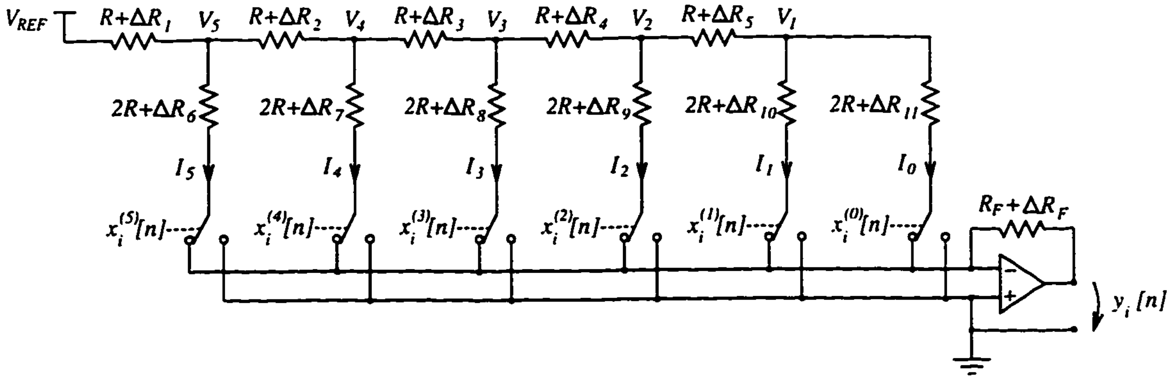


Figure 2.5: The DAC-bank implemented using an $R - 2R$ ladder network with resistor errors and non-zero ohmic switch resistance.

Thus, each analog output level error can be written as a linear combination of the individual bits of $x_i[n]$.

The analog output level error of the DAC-bank will subsequently be assumed to be of the form

$$\Delta\{x_i[n]\} = \sum_{j=0}^{5} e^{(j)}[n], \tag{12}$$

where

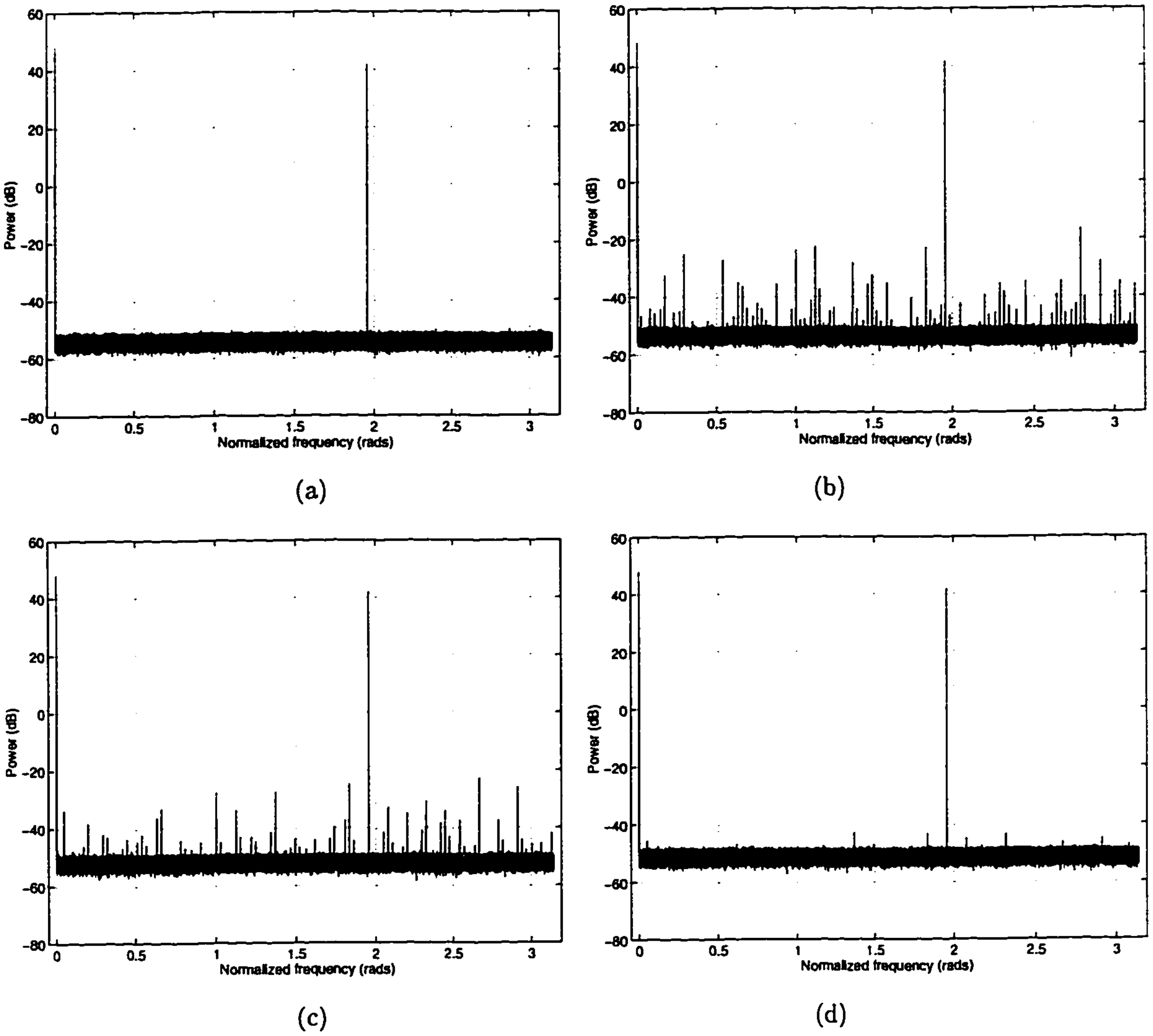
$$e^{(j)}[n] = \begin{cases} e_h^{(j)} & \text{if } x_i^{(j)}[n] = 1; \\ e_l^{(j)} & \text{if } x_i^{(j)}[n] = 0, \end{cases} \tag{13}$$

and where the $e_h^{(j)}$ and $e_l^{(j)}$ are time-invariant, but otherwise arbitrary, and are referred to as *static DAC-element errors.* Notice that (13) allows for asymmetric contributions of $x_i^{(j)}[n]$ in (12), depending upon whether $x_i^{(j)}[n]$ is 0 or 1. This form of the analog output level error accommodates for other popular architectures that can be used to realize the DAC-bank, including the weighted resistor network DAC, the binary weighted current-steering DAC, and the charge scaling DAC [3].

With non-zero static DAC-element errors, it was shown in [2] that the DAC output has the form

$$y[n] = \alpha x[n] + \beta + e[n],$$

where α is a constant gain, β is a DC offset, and $e[n]$ is a conversion error term



(a)

(b)

(c)

(d)

Figure 2.6: Simulated PSDs relative to x_{max}^2 of an example 8-bit DAC implemented using $R-2R$ ladder network DAC-banks. The resistor errors were normally distributed of standard deviation $\sigma = 1\%$. Plot (a) shows the DAC input $x[n]$ (and thus the ideal DAC output $y[n]$), and (b) through (d) show the DAC output $y[n]$ with randomization indices 1, 2, and 3, respectively.

referred to as *DAC noise.*

A. Simulation Results

The performance of partial randomization DEM is demonstrated in Figure 2.6 with simulation results of an example 8-bit version of the proposed DAC. The DAC-banks were each simulated as an $R-2R$ ladder network with $R_F = 2R$. The resistor errors ΔR_k were chosen as samples of independent, normally distributed random variables with standard deviation $\sigma = 1\%$ relative to the nominal resistor values. Thus, the static DAC-element errors satisfied

$$e_h^{(j)} = \Delta_j \quad \text{and} \quad e_l^{(j)} = 0,$$

with Δ_j given by (11). It should be emphasized that this particular choice of static DAC-element errors was made for simulation purposes only; the theoretical results developed in this paper do not depend upon any specific statistical distribution or correlation properties of the static DAC-element errors.

Each graph in Figure 2.6 shows the simulated power spectral density (PSD) of $y[n]$ in dB relative to x_{max}^2, where $x_{max} = 255$, when driving the DAC by a dithered and DC offset sinusoid. Specifically, $x[n]$ was formed by adding dither to the sequence $A\sin(\omega_0 n) + DC_x$, where $A = 126$, $\omega_0 = \frac{1275}{2048}\pi$, and $DC_x = 127.5$, and then quantizing the result to 8 bits. The dither added to the sinusoidal input was a white sequence with a triangular probability density function supported on $(-1, 1)$, so the quantization error was white noise [4].

Figure 2.6a corresponds to an ideal DAC (i.e., a DAC with no static DAC-element errors, so $y[n] = x[n]$), and Figures 2.6b through 2.6d correspond to a DAC with non-zero static DAC-element errors and randomization indices 1, 2, and 3, respectively. As is evident from the numerous spurs distributed across the spectrum in Figure 2.6b, quite severe harmonic distortion results from the static DAC-element errors with only one layer of random switching. With the particular choice of static DAC-element errors used for the plots of Figure 2.6, the maximum-amplitude spur occurs at a frequency of approximately 2.8π radians, and has power 58.1 dB below the power of the desired sinusoidal signal. Thus, the SFDR provided by the DAC of this example is 58.1 dB. The simulation results of Figures 2.6c and 2.6d indicate that the SFDR increases as the randomization index increases; the SFDR provided by the DAC in Figure 2.6c is 64.7 dB, whereas the SFDR provided by the DAC in Figure 2.6d is 84.6 dB. Additional simulations using other sinusoid frequencies yield similar findings.

Additional details of the simulation results are as follows. The PSDs were each estimated by averaging 16 length-2^{18} periodograms [5]. The frequency scales were normalized such that π corresponds to half the clock rate of the DAC.

B. Performance Equations

A complete theoretical analysis of partial randomization DEM is given in the appendices. For the purpose of comparing simulation results and theory, this section presents the main results of the analysis.

Consider a b-bit version of the proposed DAC architecture with randomization index I, i.e., with random switching in layers $b-I+1, b-I+2, \ldots b$. Let $x[n]$ be a deterministic input sequence and let $x^{(j)}[n]$ denote the j^{th} bit of $x[n]$, $1 \le j \le b$. In accordance with the usual definitions, let the *time-average means* of $x[n]$, $x^{(i)}[n]$, and $x^{(i)}[n]x^{(j)}[n]$ be defined as

$$\overline{M}_x = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n],$$

$$\overline{M}_{x^{(i)}} = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x^{(i)}[n],$$

and

$$\overline{M}_{x^{(i,j)}} = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x^{(i)}[n]x^{(j)}[n],$$

respectively, and let the *time-average autocorrelation* of $x[n]$ be defined as

$$\overline{R}_{xx}[k] = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n]x[n+k].$$

The time-average autocorrelation of $y[n]$ is defined analogously with x replaced by y in the above definition. The two main theoretical results of this paper can now be stated as follows:

Result 1. *The DAC output can be written in the form*

$$y[n] = \alpha x[n] + \beta + e[n] \tag{14}$$

where

$$e[n] = w[n] + s[n], \tag{15}$$

$w[n]$ *is a zero-mean, white random process of the form*

$$w[n] = \sum_{i=1}^{b} w_i[n]\, x^{(i)}[n], \tag{16}$$

$w_i[n]$ *is a zero-mean, white sequence uncorrelated with* $x[n]$, *and* $s[n]$ *is a deterministic sequence of the form*

$$s[n] = \sum_{i=1}^{b-I} s_i\, x^{(i)}[n]. \tag{17}$$

Appendix 2.A provides formulae for the constants α and β in (14), and develops expressions for $w_i[n]$ in (16) and s_i in (17). For now, it suffices to know that the constant coefficients s_i depend only upon the static DAC-element errors.

Result 2. *If* $\overline{M}_x$ *and* $\overline{R}_{xx}[k]$ *exist, then*

$$\overline{R}_{yy}[k] = \overline{R}_{x_s x_s}[k] + \overline{\eta} + \overline{\sigma}^2 \delta[k], \tag{18}$$

with probability 1, where

$$\overline{\eta} = 2\beta \overline{M}_{x_s} + \beta^2, \tag{19}$$

and

$$\overline{\sigma}^2 = \sum_{i=1}^{b} \gamma_i\, \overline{M}_{x^{(i)}} + \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \gamma_{i,j}\, \overline{M}_{x^{(i,j)}}. \tag{20}$$

In (18), $x_s[n]$ *is the sequence resulting from lumping* $\alpha x[n]$ *and* $s[n]$ *together, i.e.,*

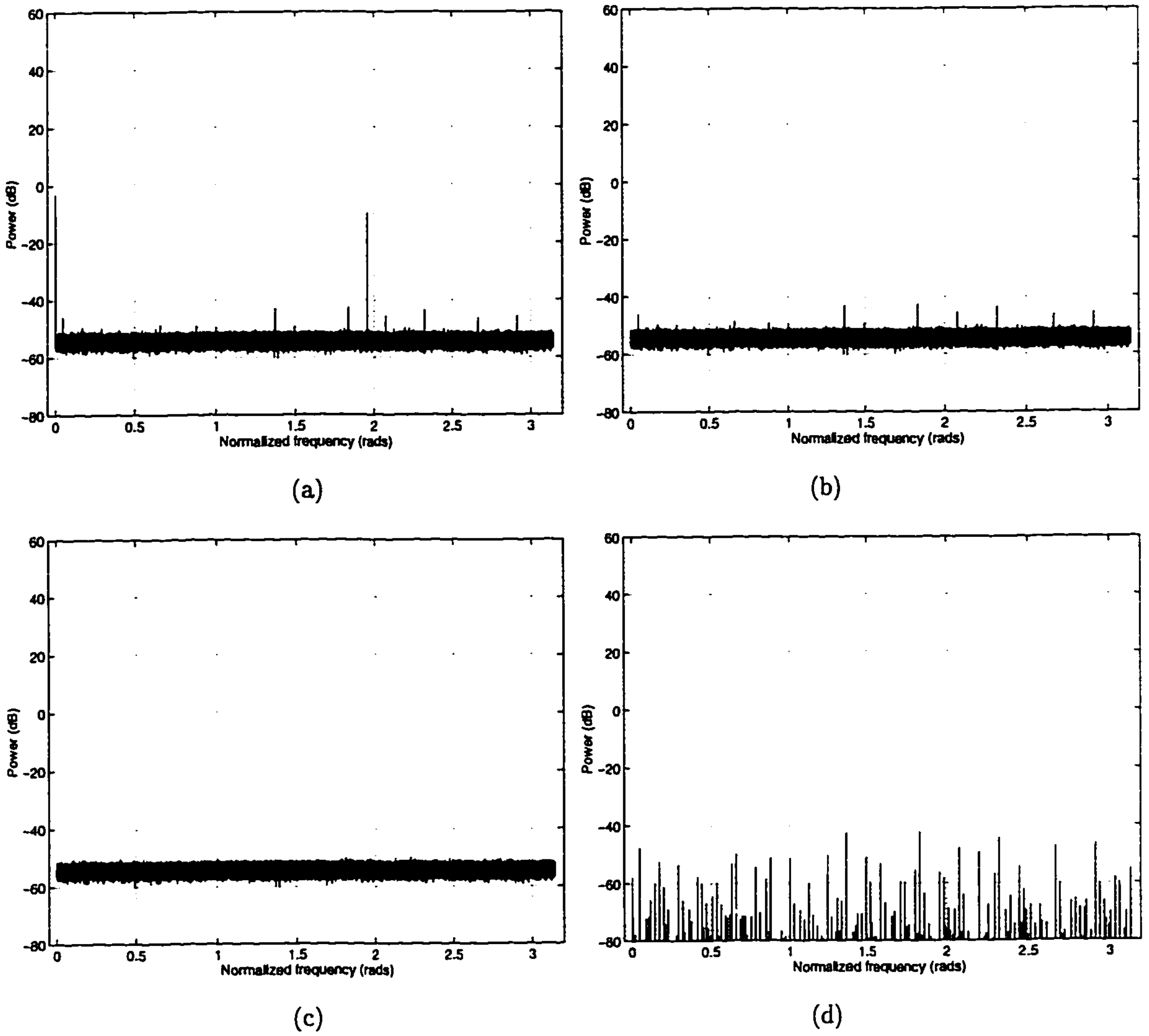
$$x_s[n] = \alpha x[n] + s[n]. \tag{21}$$

Appendix 2.B provides formulae for the constant coefficients γ_i and $\gamma_{i,j}$ in (20). For now, it suffices to know that these coefficients depend only upon the static DAC-element errors. It must be mentioned that Result 1 and Result 2 are independent of the underlying statistical distribution or any correlation properties of the static DAC-element errors.

C. Comparison of Simulation and Theory

To summarize, (14) through (17) imply that $y[n]$ consists of a scaled version of $x[n]$, a DC-offset, and DAC noise, $e[n]$, which consists of the sum of white noise, $w[n]$, and a signal dependent component, $s[n]$. Thus, $s[n]$ accounts for *all* spurious tones present in $y[n]$, and, as can be seen from (17), the tones result from a linear combination of the input-bits $x^{(1)}[n]$ through $x^{(b-I)}[n]$. Given knowledge of $x[n]$ and the static DAC-element errors, these equations fully quantify all harmonic distortion present in $y[n]$.

The validity of the detailed knowledge of the spurious content of $y[n]$ implied by Result 1 can be demonstrated by computer simulation. As an example, each plot in Figure 2.7 shows the simulated PSD relative to x_{max}^2 of selected signals of the example 8-bit DAC used for Figure 2.6d. The DAC was driven by the same dithered and DC offset sinusoid, and the randomization index was fixed at 3. Figure 2.7a shows the sequence $y[n]-x[n]$. As expected from (14), the resulting sequence appears to consist of a scaled version of $x[n]$, white noise, and spurious tones. Figure 2.7b shows $e[n]$ formed as the sequence $y[n] - \alpha x[n] - \beta$, and it appears to consist of white noise and spurious tones. Figure 2.7c shows $w[n]$ formed as the sequence $y[n] - \alpha\ x[n] - \beta - s[n]$, where $s[n]$ is given by (17). As expected, $w[n]$ appears to be a zero-mean, white noise random process. Figure 2.7d shows the corresponding $s[n]$ as computed using (17). Notice that superimposing the results of Figure 2.7c and Figure 2.7d yields the results of Figure 2.7b, as expected from (15). Thus, the



(a)

(b)

(c)

(d)

Figure 2.7: Simulated PSD relative to x^2_{max} of selected signals of the example 8-bit DAC used for Figure 2.6d. The DAC was driven by the same dithered and DC offset sinusoid, and the randomization index was fixed at 3. Figures 2.7a through 2.7d show the sequences $y[n] - x[n]$, $e[n]$, $w[n]$, and $s[n]$, respectively.

formulae for $y[n]$, $e[n]$, $w[n]$, and $s[n]$ are supported by the simulation results of Figure 2.7.

The power of $w[n]$ is given by (20), and the validity of this expression can also be demonstrated using the simulation results of Figure 2.7. Specifically, evaluating (20) for the simulated values of static DAC-element errors and the input sequence applied to the DAC yields a power of $w[n]$ of $\overline{\sigma}^2 = -54.14$ dB relative to x_{max}^2. Numerically integrating the results of Figure 2.7c yields -54.15 dB relative to x_{max}^2, in agreement with the theoretical result.

D. An Interpretation of the Performance Equations

The theoretical results presented above are used in the following to develop a simple expression for the guaranteed minimum SFDR resulting from partial randomization DEM. A few definitions helpful in this development are first presented. Let $\mathcal{I}^+$ denote the set of indices i in (17) such that $s_i > 0$, i.e.,

$$\mathcal{I}^+ = \{\, i : s_i > 0 \,\} \tag{22}$$

and let $\mathcal{I}^-$ be the set of indices i in (17) such that $s_i < 0$, i.e.,

$$\mathcal{I}^- = \{\, i : s_i < 0 \,\} \tag{23}$$

Let s^+ denote the sum of all positive s_i, i.e.,

$$s^+ = \sum_{i \in \mathcal{I}^+} s_i, \tag{24}$$

let s^- denote the sum of all negative s_i, i.e.,

$$s^- = \sum_{i \in \mathcal{I}^-} s_i, \tag{25}$$

and let $A_{s_{max}}$ denote half the difference of s^+ and s^-, i.e.,

$$A_{s_{max}} = \frac{(s^+ - s^-)}{2}. \tag{26}$$

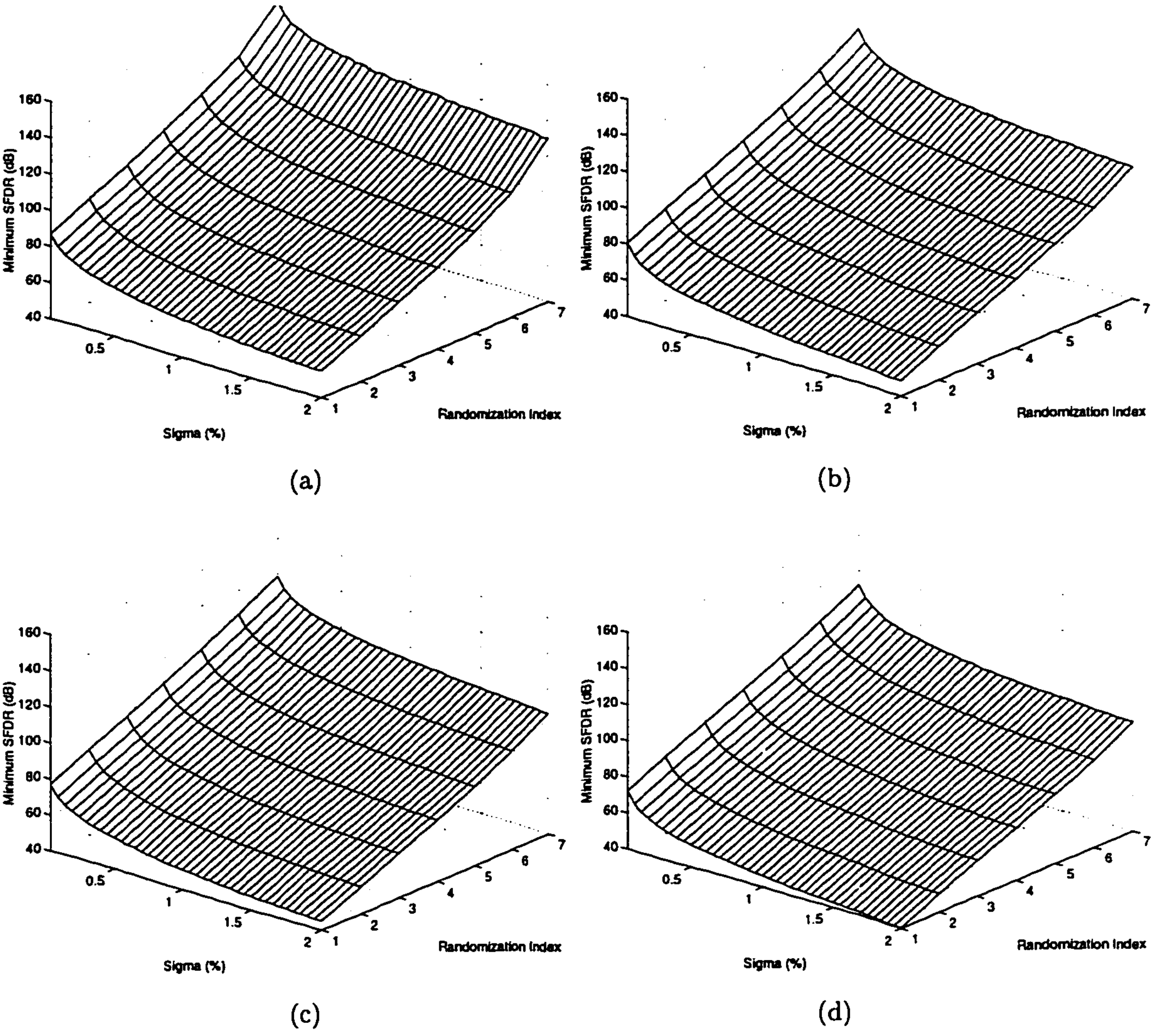
Parseval's relation implies that the power of $s[n]$ equals the sum of the powers of the distinct spurs occurring in the PSD of $s[n]$. Thus, worst-case SFDR performance occurs when the power of $s[n]$ is at maximum and it consists of as few distinct spurs as possible. Since $s[n]$ is real, its PSD is symmetric, and the minimum number of distinct spurs is *two*, corresponding to non-zero frequencies ω_s and $-\omega_s$. Since $x^{(i)}[n] \in \{0, 1\}$, it follows from (17) that $s^- \leq s[n] \leq s^+$. Thus, worst-case total spur power is $A^2_{s_{max}}$, and the power of each spur is bounded by $A^2_{s_{max}}/2$. The amplitude of the signal component of $y[n]$ for a maximum-amplitude sinusoidal input signal is $\alpha x_{max}/2$. It follows that the DAC provides an SFDR of at least $(\alpha x_{max}/2A_{s_{max}})^2$. Stating this result in dB yields

$$\text{Minimum SFDR} : 20 \log_{10}\left(\frac{\alpha x_{max}}{2A_{s_{max}}}\right) \text{ dB.} \tag{27}$$

IV. IC FABRICATION YIELD ESTIMATION

With knowledge—or an assumption—of the statistical distribution of the static DAC-element errors, (20) and (27) allow for IC fabrication yield estimations of $\overline{\sigma}^2$ and the minimum SFDR, respectively. The IC fabrication yield estimation procedure presented in the following was first introduced in [6], and is based upon the idea of computing a large number of samples of the parameter of interest for a given level of static DAC-element errors, thereby employing the law of large numbers to generate data that closely resemble the corresponding statistical distribution of the parameter.

An example IC fabrication yield estimation of the minimum SFDR computed using (27) and its supporting equations is shown in Figure 2.8. Specifically, the figure shows the minimum SFDR for the 8-bit version of the DAC implemented using $R - 2R$ ladder network DAC-banks with normally distributed resistor errors of standard deviation σ ranging from 0.05% to 2%, and randomization index ranging from 1 to 7. Figures 2.8a through 2.8d show the smallest of the largest 5%, 35%, 65%, and 95% values, respectively. Each plot was based upon 10^4 calculated values. For example, with $\sigma = 1\%$ and randomization index 3, Figure 2.8a predicts that merely 5% of all 8-bit DACs provide at least 81.0 dB SFDR, while Figure 2.8d



(a)

(b)

(c)

(d)

Figure 2.8: An IC fabrication yield estimation of the minimum SFDR of the 8-bit version of the DAC implemented using $R-2R$ ladder network DAC-banks. The resistor errors were normally distributed of standard deviation σ ranging from 0.05% to 2%, and the randomization index ranged from 1 to 7. Plots (a) through (d) show the smallest of the largest 5%, 35%, 65%, and 95% values, respectively.

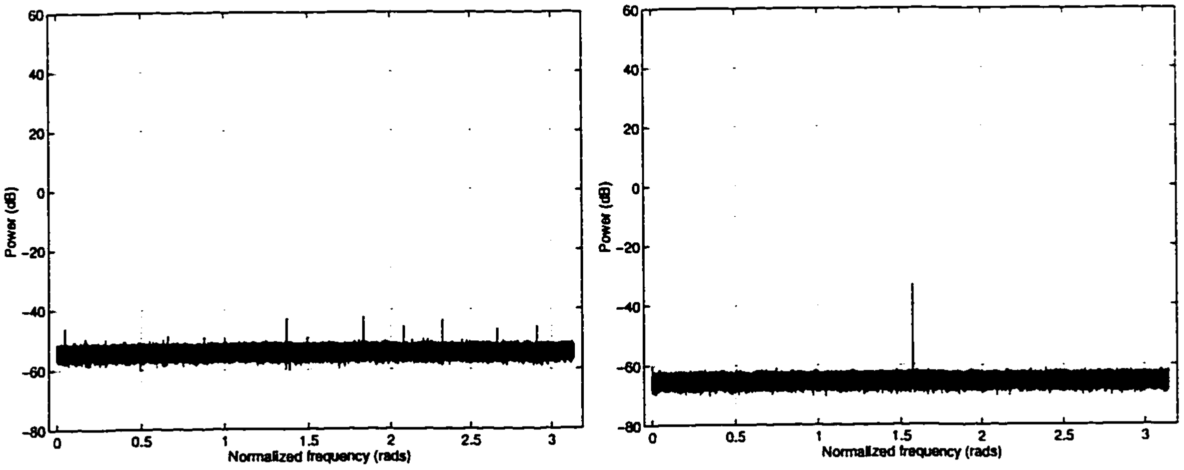
predicts that 95% of all 8-bit DACs provide at least 65.7 dB SFDR. Thus, 90% of all 8-bit DACs with randomization index 3 fabricated in a VLSI process resulting in normally distributed resistor errors with standard deviation $\sigma = 1\%$ satisfy

$$65.7 \text{ dB} \leq \text{Minimum SFDR} \leq 81.0 \text{ dB}. \tag{28}$$

An interesting conclusion to be drawn from Figure 2.8 is that the estimated minimum SFDR increases by approximately 10 dB per unit increment of the randomization index, independent of the standard deviation of the resistor errors. Additional computations suggest that this conclusion generalizes to DACs of any bit-resolution of interest. It should be mentioned that in the special case of full randomization DEM [2], i.e., with random switching in all 8 layers, the DAC-noise is fully uncorrelated with $x[n]$. Harmonic distortion is therefore completely eliminated and the DAC—in principle—provides an infinite SFDR. Thus, the 10 dB increase of the minimum SFDR per increment of the randomization index does not apply in the case of incrementing the number of random layers from 7 to 8.

At first glance, the above estimates for the minimum SFDR achievable with partial randomization DEM may seem overly conservative given that the example 8-bit DAC used for the plots of Figure 2.6 achieved an SFDR of 84.6 dB with randomization index 3. Recall, however, that the IC fabrication yield estimation represents worst-case performance, where it is assumed that all power in $s[n]$ is lumped into *two* spurs occurring at frequencies $+\omega_s$ and $-\omega_s$. When driving the DAC by a (dithered) sinusoid, typically *many* spurs occur in the PSD of $e[n]$, as observed in Figure 2.9a, which is Figure 2.6b repeated here for convenience. The occurrence of numerous spurs results in decreased power of the maximum-amplitude spur—and thus an increased SFDR—relative to worst-case performance. Figure 2.9b shows the PSD of $e[n]$ of the same 8-bit DAC used for the plot of Figure 2.9a, but driven by a sequence $x[n]$ chosen such that the resulting PSD of $s[n]$ consists of only two spurs. In this case, the maximum-amplitude spur has power -75.1 dB relative to $(\alpha x_{max})^2/8$, in support of (28).

An example IC fabrication yield estimation of $\overline{\sigma}^2$ computed using (20) is shown in Figure 2.10. Specifically, the figure shows $\overline{\sigma}^2$ in dB relative to x_{max}^2 for the 8-bit version of the DAC implemented using $R-2R$ ladder network DAC-banks with



(a) (b)

Figure 2.9: Simulated PSDs of $e[n]$ relative to x^2_{max} of the example 8-bit DAC. The randomization index was fixed a 3, and the plots show (a) $e[n]$ resulting from the dithered sinusoidal input, and (b) $e[n]$ resulting from an input sequence $x[n]$ chosen such that the PSD of $s[n]$ consists of only *two* spurs.

normally distributed resistor errors of standard deviation σ ranging from 0.05% to 2%, and randomization index ranging from 1 to 7. Figures 2.10a through 2.10d show the smallest of the largest 95%, 65%, 35%, and 5% values of $\overline{\sigma}^2$, respectively, when driving the DAC by a maximum-amplitude, DC offset sinusoidal input of frequency $\omega_0 = \frac{625}{2048}\pi$. Each plot was based upon 3000 calculated values. For example, with $\sigma = 1\%$ and randomization index 3, Figure 2.10a predicts that 95% of all 8-bit DACs satisfy $\overline{\sigma}^2 \leq -53.2$ dB relative to x^2_{max}, while of Figure 2.10d predicts that 5% of all 8-bit DACs satisfy $\overline{\sigma}^2 \leq -63.6$ dB relative to x^2_{max}. Thus, 90% of all fabricated 8-bit DACs satisfy

$$-63.6 \text{ dB} \leq \overline{\sigma}^2 \leq -53.2 \text{ dB} \tag{29}$$

relative to x^2_{max}. This conclusion is supported by the results of the simulated example 8-bit DAC of Figure 2.6, for which $\overline{\sigma}^2 = -54.2$ dB relative to x^2_{max}.

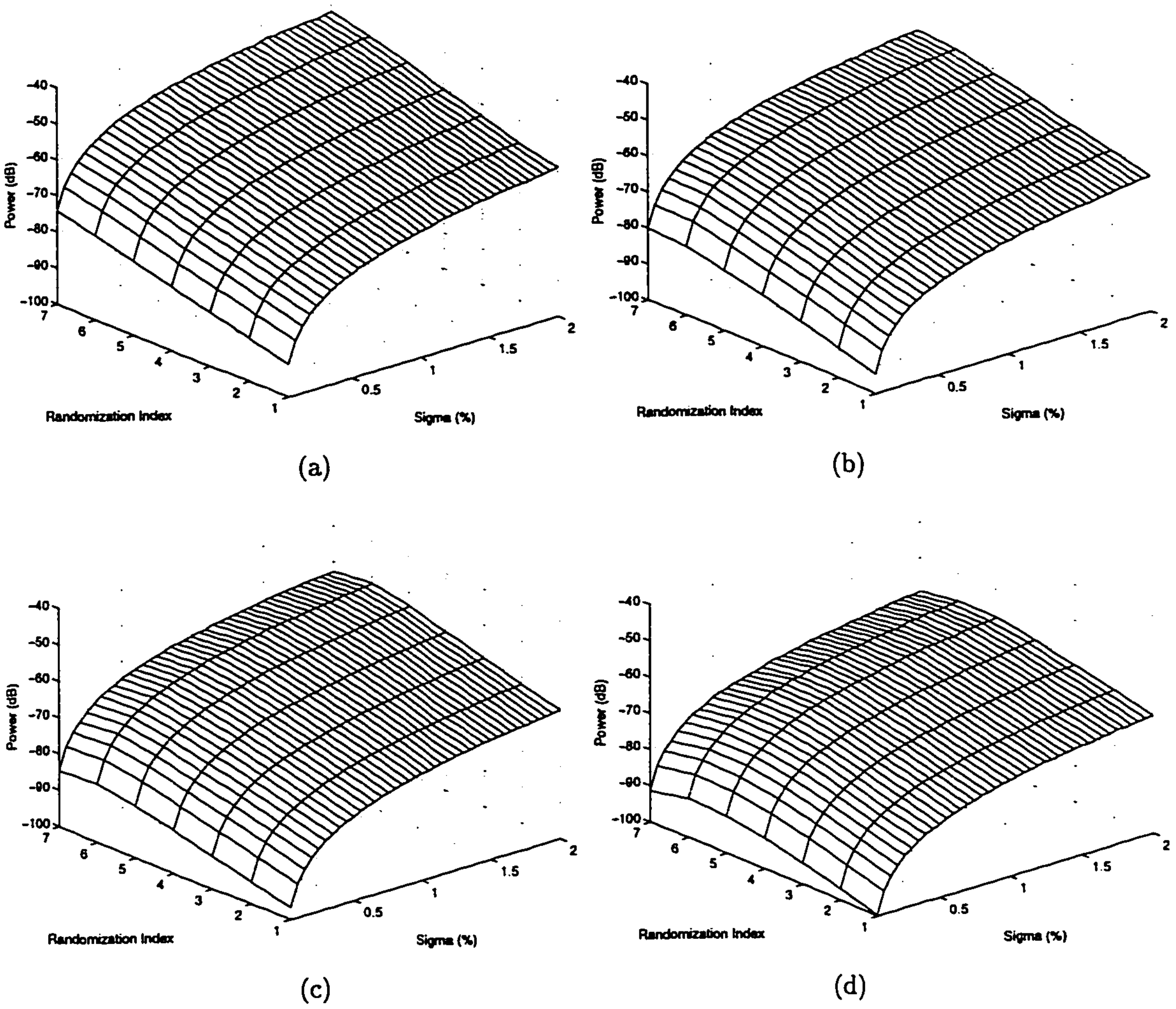


Figure 2.10: An IC fabrication yield estimation of $\overline{\sigma}^2$ in dB relative to x_{max}^2 of the 8-bit version of the DAC implemented using $R-2R$ ladder network DAC-banks. The resistor errors were normally distributed of standard deviation σ ranging from 0.05% to 2%, and the randomization index ranged from 1 to 7. Plots (a) through (d) show the largest of the smallest 95%, 65%, 35%, and 5% values, respectively.

V. CONCLUSION

A detailed analysis of partial randomization DEM originally proposed in [2] has been presented. It was demonstrated using simulation results and shown with theory that harmonic distortion resulting from the static DAC-element errors is increasingly suppressed as the randomization index is increased. Specifically, it was observed that when implementing the DAC-banks using $R-2R$ ladder networks with normally distributed resistor errors, the estimated minimum SFDR increases by approximately 10 dB per increment of the randomization index, independent of the particular DAC bit-resolution.

An IC fabrication yield estimation of the minimum SFDR has been presented that, given knowledge of the statistics of the static DAC-element errors, can be used in the design of a DAC with minimum hardware complexity while still providing the SFDR required for a given application. An IC fabrication yield estimation of the noise-floor as a function of the resistor errors and the randomization index has also been presented.

Harmonic distortion resulting from inevitable non-ideal circuit behavior such as clock-skew, non-linear settling, and finite slew-rates has not been considered in this paper. Such non-ideal circuit behavior is typically quite implementation dependent, and research to quantify and mitigate its effects must likely be performed on a case-by-case basis. However, as has been shown, using partial randomization DEM, the harmonic distortion resulting from static DAC-element errors can be attenuated to the level of inevitable harmonic distortion, independent of the particular circuit technology.

APPENDIX 2.A

The purpose of this appendix is to verify the form of the DAC output as stated by Result 1 of Section III.

Claim A. *Consider a b-bit version of the proposed DAC architecture with randomization index I, $1 \leq I < b$, i.e., with random switching in layers $b - I + 1, b - I + 2, \ldots, b$. In this case, the DAC-banks consists of a $b - I$-bit conventional DAC and a 1-bit conventional DAC. Suppose that the analog output level error of the i^{th} DAC-bank is of the form given by the generalized versions of* (12) *and* (13). *Let $x[n]$ be a deterministic input sequence and let $x^{(j)}[n]$ denote the j^{th} bit of $x[n]$, $1 \leq j \leq b$. The output of the DAC can then be written in the form*

$$y[n] = \alpha x[n] + \beta + e[n] \tag{30}$$

where

$$e[n] = w[n] + s[n], \tag{31}$$

and $w[n]$ is a zero-mean, white random process of the form

$$w[n] = \sum_{i=1}^{b} w_i[n]\, x^{(i)}[n], \tag{32}$$

and $s[n]$ is a deterministic sequence of the form

$$s[n] = \sum_{i=1}^{b-I} s_i\, x^{(i)}[n]. \tag{33}$$

In (30),

$$\alpha = 1 + \frac{1}{2^b} \sum_{i=1}^{2^I} \sum_{j=0}^{b-I} (e_{h_i}^{(j)} - e_{l_i}^{(j)}) \tag{34}$$

and

$$\beta = \sum_{i=1}^{2^I} \sum_{j=0}^{b-I} e_{l_i}^{(j)}. \tag{35}$$

In (32), each $w_i[n]$ is a zero-mean, white random process of the form

$$w_i[n] = \begin{cases} w_{i,h_i[n]} & \text{if } i > b - I; \\ w_{i,h_i[n]} - s_i & \text{if } i \leq b - I; \end{cases} \tag{36}$$

where

$$w_{i,j} = \begin{cases} \displaystyle\sum_{m=(j-1)2^{i-b+I-1}+1}^{j2^{i-b+I-1}} \sum_{k=0}^{b-I} (e_{h_m}^{(k)} - e_{l_m}^{(k)}) - 2^{i-1}(\alpha - 1) & \text{if } i > b - I; \\ e_{h_j}^{(i)} - e_{l_j}^{(i)} - 2^{i-1}(\alpha - 1) & \text{if } i \leq b - I; \end{cases} \tag{37}$$

and

$$h_i[n] = \begin{cases} \displaystyle\sum_{j=1}^{b-i} 2^j c_{i+j}[n] + (1 - c_i[n]) + 1 & \text{if } i > b - I; \\ \displaystyle\sum_{j=1}^{I} 2^{j-1} c_{b-I+j}[n] + 1 & \text{if } i \leq b - I. \end{cases} \tag{38}$$

In (33) and (36),

$$s_i = -\frac{1}{2^I} \sum_{k=1}^{2^I} \left(\sum_{\substack{j=1 \\ j \neq i}}^{b-I} w_{j,k} + e_{h_k}^{(0)} - e_{l_k}^{(0)} - (\alpha - 1) \right). \tag{39}$$

The above results do not depend upon any particular form or statistical properties of the static DAC-element errors.

Proof. To prove Claim A, several of the theoretical results developed in [2] will be used. For convenience, these are repeated in the following.

Results for full randomization DEM. *For a b-bit version of the DAC architecture with full randomization DEM, the output of the DAC can be written in the form*

$$y[n] = \alpha x[n] + \beta + e[n], \tag{40}$$

where

$$\alpha = 1 + \frac{1}{2^b} \sum_{i=1}^{2^b} (e_{h_i} - e_{l_i}), \tag{41}$$

$$\beta = \sum_{i=1}^{2^b} e_{l_i}, \tag{42}$$

and $e[n]$ *is a zero-mean, white random process of the form*

$$e[n] = \sum_{i=1}^{b} w_i[n]\, x^{(i)}[n]. \tag{43}$$

In (43)*, each* $w_i[n]$ *is a zero-mean, white random process of the form*

$$w_i[n] = w_{i,h_i[n]}, \tag{44}$$

where

$$w_{i,j} = \sum_{m=(j-1)2^{i-1}+1}^{j2^{i-1}} (e_{h_m} - e_{l_m}) - 2^{i-1}(\alpha - 1) \tag{45}$$

and

$$h_i[n] = \sum_{j=1}^{b-i} 2^j c_{i+j}[n] + (1 - c_i[n]) + 1. \tag{46}$$

The above results do not depend upon any particular form or statistical properties of the static DAC-element errors.

As explained in [2], partial randomization DEM with randomization index I is equivalent to full randomization DEM with the restriction that

$$c_k[n] = 1 \quad \text{for} \quad k = 1, 2, \ldots, b - I, \tag{47}$$

so that the layers 1 through $b-I$ can be eliminated and substituted with an array of DAC-banks, each consisting of a $b-I$-bit conventional DAC and a 1-bit conventional DAC. Specifically, since the static DAC-element errors of full randomization DEM are denoted e_{h_i} and e_{l_i}, $1 \leq i \leq 2^b$, the restriction (47) implies that the i^{th} DAC-bank operates according to (12) and (13), where the static DAC-element errors associated with the $b - I$-bit conventional DAC satisfy

$$e_{h_i}^{(j)} = \sum_{m=i2^{b-I}-2^j+1}^{i2^{b-I}-2^{j-1}} e_{h_m} \quad \text{and} \quad e_{l_i}^{(j)} = \sum_{m=i2^{b-I}-2^j+1}^{i2^{b-I}-2^{j-1}} e_{l_m}, \tag{48}$$

and the static DAC-element errors associated with the 1-bit conventional DAC satisfy

$$e_{h_i}^{(0)} = e_{h_{i2^{b-I}}} \qquad \text{and} \qquad e_{l_i}^{(0)} = e_{l_{i2^{b-I}}}. \tag{49}$$

In the following, the results for partial randomization DEM stated in Claim A will be shown to be equivalent to the results for full randomization DEM with the restriction (47).

First, the formulae for α and β given by (34) and (35) are developed. As noted above, the results for full randomization DEM hold for *arbitrary* values of the static DAC-element errors e_{h_i} and e_{l_i}. In particular, if in (48) the following choice is made

$$e_{h_m} = e_{l_m} = 0 \quad \text{for} \quad i2^{b-I} - 2^j + 1 \le m < i2^{b-I} - 2^{j-1}, \tag{50}$$

then

$$e_{h_i}^{(j)} = e_{h_{i2^{b-I}-2^{j-1}}} \quad \text{and} \quad e_{l_i}^{(j)} = e_{l_{i2^{b-I}-2^{j-1}}} \quad \text{for} \quad j = 1, 2, \ldots, b-I. \tag{51}$$

It follows using (48) through (51) that for fixed value of i,

$$\sum_{j=(i-1)2^{b-I}+1}^{i2^{b-I}} (e_{h_j} - e_{l_j}) = \sum_{j=0}^{b-I} (e_{h_i}^{(j)} - e_{l_i}^{(j)}), \tag{52}$$

and thus

$$\sum_{i=1}^{2^b} (e_{h_i} - e_{l_i}) = \sum_{i=1}^{2^I} \sum_{j=0}^{b-I} (e_{h_i}^{(j)} - e_{l_i}^{(j)}),$$

which shows that (34) is equivalent to (41). Similarly,

$$\sum_{i=1}^{2^b} e_{l_i} = \sum_{i=1}^{2^I} \sum_{j=0}^{b-I} e_{l_i}^{(j)},$$

which shows that (35) is equivalent to (42).

Next, the formulae (36) through (39) detailing the form of the DAC noise will be verified. For the case $i > b - I$, it is straight-forward to show that (36) through (38) follow from (43) through (46). Specifically, it suffices to show that (37) is equivalent to (45). For this purpose, the use of (52) and some algebraic manipulations result in

$$\sum_{k=(j-1)2^{i-1}+1}^{j2^{i-1}} (e_{h_k} - e_{l_k}) = \sum_{m=(j-1)2^{i-b+I-1}+1}^{j2^{i-b+I-1}} \sum_{k=0}^{b-I} (e_{h_m}^{(k)} - e_{l_m}^{(k)}),$$

which suffices to show the asserted result. Notice that it follows directly from [2] that $w_i[n]$ is zero-mean, since the possible values of $w_i[n]$ are unaffected by (47).

For the case $i \leq b - I$, it will first be shown that $w_{i,j}$ in (37) with subscript defined by (38) is equivalent to $w_{i,j}$ in (45) with subscript defined by (46) and the restriction (47). For this purpose, notice that with the restriction (47), it follows after some algebraic manipulations that (46) can be written as

$$h_i[n] = \left(\sum_{j=1}^{I} 2^{j-1} c_{b-I+j}[n] + 1 \right) 2^{b-I+1-i} - 1. \tag{53}$$

Thus, if in (38) $h_i[n] = j$, it follows that (53) may be written as

$$h_i[n] = j2^{b-I+1-i} - 1.$$

But, as follows from (45) and some algebraic manipulations,

$$w_{i,j2^{b-I+1-i}-1} = \sum_{m=j2^{b-I}-2^i+1}^{j2^{b-I}-2^{i-1}} (e_{h_m} - e_{l_m}) - 2^{i-1}(\alpha - 1),$$

which—referring to (48)—is equivalent to (37), as asserted. It will next be shown that $w_i[n]$ is zero-mean. It follows from (44) through (46) that the effect of the restriction (47) is to reduce the set of possible values of $w_i[n]$ to a subset of the set of values otherwise possible. Therefore, with the restriction (47), $w_i[n]$ in (44) is a white random process, generally with a non-zero mean value. Denoting this mean value s_i, (43) may be written as

$$e[n] = \sum_{i=b-I+1}^{b} w_i[n]\, x^{(i)}[n] + \sum_{i=1}^{b-I} (w_i[n] - s_i)\, x^{(i)}[n] + \sum_{i=1}^{b-I} s_i\, x^{(i)}[n].$$

Defining the white, *zero-mean* random processes $w_i[n]$ as in (36) results in

$$e[n] = \sum_{i=1}^{b} w_i[n]\, x^{(i)}[n] + \sum_{i=1}^{b-I} s_i\, x^{(i)}[n],$$

or, referring to (32) and (33),

$$e[n] = w[n] + s[n],$$

as asserted in (31). It remains to verify (39) by establishing that $w_i[n]$ in (36) is indeed zero-mean. From (38) it follows that there are 2^I possible equally probable values of $w_i[n]$. Specifically, during the n^{th} sample period,

$$w_i[n] = w_{i,j} - s_i, \quad \text{where} \quad j \in \{1, 2, \ldots, 2^I\}.$$

Then, by definition,

$$\mathrm{E}\left\{w_i[n]\right\} = \frac{1}{2^I} \sum_{j=1}^{2^I} (w_{i,j} - s_i),$$

and defining s_i as in (39) yields

$$\begin{aligned}
\mathrm{E}\left\{w_i[n]\right\} &= \frac{1}{2^I} \sum_{j=1}^{2^I} \left(\sum_{i=1}^{b-I} w_{i,j} + e_{h_j}^{(0)} - e_{l_j}^{(0)} - (\alpha - 1) \right) \\
&= \frac{1}{2^I} \sum_{j=1}^{2^I} \left(\sum_{i=0}^{b-I} (e_{h_j}^{(i)} - e_{l_j}^{(i)}) - 2^{b-I}(\alpha - 1) \right) \\
&= \frac{1}{2^I} \left(2^b(\alpha - 1) - 2^b(\alpha - 1) \right) \\
&= 0,
\end{aligned}$$

as asserted. ■

APPENDIX 2.B

This appendix states and proves an expression for the variance of the zero-mean, white random process $w[n]$.

Claim B. *The variance of $w[n]$ is given by*

$$\mathrm{Var}\{w[n]\} = \sum_{i=1}^{b} \gamma_i \, x^{(i)}[n] + \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \gamma_{i,j} \, x^{(i)}[n] x^{(j)}[n],$$

where

$$\gamma_i = \begin{cases} \dfrac{1}{2^{b-i+1}} \displaystyle\sum_{j=1}^{2^{b-i+1}} (w_{i,j})^2 & \text{if } i > b - I; \\ \dfrac{1}{2^I} \displaystyle\sum_{j=1}^{2^I} (w_{i,j} - s_i)^2 & \text{if } i \leq b - I; \end{cases} \tag{54}$$

and

$$\gamma_{i,j} =$$

$$\begin{cases} \dfrac{1}{2^{b-j-1}} \displaystyle\sum_{k=1}^{2^{b-i}} w_{i,2k-1} w_{i,2k} & \text{if } i,j > b - I; \\ \dfrac{1}{2^{I-1}} \displaystyle\sum_{l=0}^{1} \sum_{k=0}^{2^{b-i}-1} w_{i,2(k+1)-l} \sum_{m=0}^{2^{i-b+I-1}-1} (w_{j,2^{i-b+I-1}(2k+l)+m} - s_j) & \text{if } i > b - I,\ j \leq b - I; \\ \dfrac{1}{2^{I-1}} \displaystyle\sum_{k=1}^{2^I} (w_{i,k} - s_i)(w_{j,k} - s_j) & \text{if } i,j \leq b - I. \end{cases} \tag{55}$$

Proof. Since $w[n]$ is zero-mean, $\mathrm{Var}\{w[n]\} = \mathrm{E}\{(w[n])^2\}$, where $\mathrm{E}\{\cdot\}$ denotes the statistical expectation operator. Using (16) and rearranging terms results in

$$\mathrm{Var}\{w[n]\} = \sum_{i=1}^{b} \mathrm{E}\left\{(w_i[n])^2\right\} x^{(i)}[n] + 2 \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \mathrm{E}\left\{w_i[n] w_j[n]\right\} x^{(i)}[n] x^{(j)}[n]. \tag{56}$$

Consider first $\mathrm{E}\left\{(w_i[n])^2\right\}$. As indicated by (36), two cases are of interest: For $i > b - I$, the corresponding result in Appendix B of [2] is directly applicable to obtain the result asserted in (54), since the possible values of $w_i[n]$ are unaffected by (47).

For $i \leq b-I$, it follows from (38) that there are 2^I possible equally probable values of $w_i[n]$, dictated by $c_k[n], k = b-I+1, b-I+2, \ldots, b$. Using (36) and applying the definition of statistical expectation gives the result asserted in (54).

Consider next $\mathrm{E}\left\{w_i[n]w_j[n]\right\}$, where $1 \leq j < i \leq b$. Three cases are of interest: For $i, j > b-I$, the corresponding result in Appendix B of [2] is directly applicable to obtain the result asserted in (54).

For $i, j \leq b-I$, the probability of a specific value of $w_i[n]$ is $\frac{1}{2^I}$, as discussed above. It follows from (38) that the value of $w_j[n]$ fully depends upon $w_i[n]$; suppose that the value of $w_i[n]$ is $w_{i,k} - s_i$ for some $k \in \left\{1, 2, \ldots, 2^I\right\}$. Then there is only *one* possible value of $w_j[n]$, namely $w_{j,k} - s_j$. Applying the definition of statistical expectation—and accounting for the factor of 2 occurring in (56)—gives the result asserted in (55).

For $i > b-I$ and $j \leq b-I$, each of the 2^{b-i+1} possible values of $w_i[n]$ occurs with probability $\frac{1}{2^{b-i+1}}$. From (38) it follows that for integer parameters k and l,

$$w_i[n] = w_{i,2(k+1)-l}, \quad \text{where} \quad k \in \left\{0, 1, \ldots, 2^{b-i}-1\right\} \quad \text{and} \quad l \in \{0, 1\}. \tag{57}$$

For fixed value of $w_i[n]$, i.e., for fixed values of k and l in (57), there are $2^{i-b+I-1}$ possible equally probable values of $w_j[n]$. Using (38) and a number of algebraic manipulations, these values can be expressed in terms of k, l, and an integer parameter m as

$$w_j[n] = w_{j,2^{i-b+I-1}(2k+l)+m} - s_j, \quad \text{where} \quad m \in \left\{0, 1, \ldots, 2^{i-b+I-1}-1\right\}.$$

Again, applying the definition of statistical expectation—and accounting for the factor of 2 occurring in (56)—verifies the result asserted in (55). ∎

APPENDIX 2.C

The purpose of this appendix is to verify the time-average properties of the DAC output as stated in Result 2 of Section III.

Claim C. *If* $\overline{M}_x$ *and* $\overline{R}_{xx}[k]$ *exist, then* $\overline{R}_{yy}[k]$ *is given by* (19), (20), *and* (21) *with probability* 1. *In* (20), γ_i *and* $\gamma_{i,j}$ *are given by* (54) *and* (55), *respectively.*

Proof. From Claim A and (21) it follows that

$$y[n] = x_s[n] + \beta + w[n].$$

The statistical autocorrelation of $y[n]$ can then be written in the form

$$R_{yy}[n,k] = \mathrm{E}\left\{(x_s[n] + \beta + w[n])\,(x_s[n+k] + \beta + w[n+k])\right\}.$$

It is easy to verify that if $\overline{M}_x$ and $\overline{R}_{xx}[k]$ exist, then $\overline{M}_{x_s}$ and $\overline{R}_{x_s x_s}[k]$ exist, since α and the s_i are finite. Arguments fully identical to those presented for the corresponding result in [2] can then be used to establish Claim C. ■

REFERENCES

1. M. J. Flanagan, G. A. Zimmerman, "Spur-reduced digital sinusoid synthesis," *IEEE Trans. on Communications*, vol. 43, no. 7, pp. 2254-2262, July 1995.
2. H. T. Jensen, I. Galton, "A low-complexity dynamic element matching DAC for direct digital synthesis," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, accepted for publication.
3. D. F. Hoeschele, *Analog-to-Digital and Digital-to-Analog Conversion Techniques.* New York, NY: John Wiley & Sons, Inc., 1994.
4. R. M. Gray, T. G. Stockham Jr., "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 805-812, May 1993.
5. A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1989.
6. I. Galton, P. Carbone, "A rigorous error analysis of D/A conversion with dynamic element matching," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 763-772, Dec. 1995.
7. L. R. Carley, "A noise shaping coder topology for 15+ bits converters," *IEEE J. Solid-State Circuits*, vol. SC-24, pp. 267-273, April 1989.
8. I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, to appear.
9. L. R. Carley and J. Kenney, "A 16-bit 4'th order noise-shaping D/A converter," *IEEE Proc. CICC*, pp. 21.7.1-21.7.4, 1988.
10. R. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters.* Boston, MA: Kluwer Academic Publishers, 1994.

Chapter 3

Yield Estimation of a Second-Order $\Delta\Sigma$ADC Employing a Noise-Shaping DAC[†]

Henrik T. Jensen, *Student Member, IEEE,* and Ian Galton, *Member, IEEE* [‡]

Abstract—**Various dynamic element matching techniques for noise-shaping DACs have recently been proposed and their efficiency in $\Delta\Sigma$ data converter applications demonstrated with simulation results and in actual circuit implementations. However, no theoretical quantification of the effects of the noise-shaping DAC on the overall $\Delta\Sigma$ data converter converter performance has been published to date. Such work is presented in this paper for a second-order $\Delta\Sigma$ADC employing a first-order noise-shaping DAC, and the results are applied in an IC fabrication yield estimation analysis of the $\Delta\Sigma$ADC conversion precision. As an example, to the extend that performance is limited by static DAC element errors, 90% of all $\Delta\Sigma$ADCs employing a 2-bit noise-shaping DAC fabricated with 1% element matching provide a conversion precision of between 13.0 and 14.8 bits when operating at an oversampling ratio of 128. The second-order $\Delta\Sigma$ modulator considered here is of particular interest because of its wide-spread use in commercial products.**

I. INTRODUCTION

As is well-known, errors in the analog output levels of conventional multi-bit DACs employed in $\Delta\Sigma$ data converters introduce *DAC noise* that resides in the signal-band and thus deteriorates the performance of the $\Delta\Sigma$ data converter

[†] In preparation for: IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING

[‡] The authors are with the Department of Electrical and Computer Engineering, UNIVERSITY OF CALIFORNIA, SAN DIEGO, La Jolla, CA 92093-0407, USA; email: htjensen@ece.ucsd.edu and galton@ece.ucsd.edu

[1], [2]. Specifically, the effects of the DAC noise are a limited spurious-free dynamic range (SFDR) and a drastic reduction of the conversion precision compared to ideal performance. A number of investigators have recently proposed dynamic element matching techniques for multi-bit *noise-shaping* DACs in an on-going effort to develop practical multi-bit DACs for use in $\Delta\Sigma$ data converters [3], [4], [5]. By moving most of the DAC noise out of the signal-band, noise-shaping DACs do not significantly deteriorate the performance of $\Delta\Sigma$ data converters. While all publications of proposed noise-shaping DAC architectures have presented simulation and experimental results that indicate promising noise-shaping properties, no work that theoretically quantifies the performance has been published previously.

This paper presents such a theoretical analysis for the first-order noise-shaping DAC proposed in [5] employed in a second-order $\Delta\Sigma$ADC. While the theory developed is applicable to any bit-resolution of interest, an example $\Delta\Sigma$ADC with a 2-bit quantizer and a 2-bit version of the noise-shaping DAC is considered in detail. The second-order $\Delta\Sigma$ modulator is considered specifically because it is widely used in both single and multi-stage commercial ADCs. To date, these ADCs mainly use one-bit quantizers because of the element matching errors in conventional multi-bit DACs. However, multi-bit quantization offers the benefit of a many-fold increase of the no-overload range compared to one-bit quantization [6], so it stands to reason that a practical multi-bit noise-shaping DAC is highly desirable for such $\Delta\Sigma$ modulators. Thus, while the introduction of hardware-efficient noise-shaping DACs—such as the first-order version in [5]—has made it feasible to employ a noise-shaping DAC in the second-order $\Delta\Sigma$ modulator, the lack of theoretical performance analyses has rendered IC fabrication yield estimation of the overall $\Delta\Sigma$ data converter performance difficult.

The results of this paper do allow for an IC fabrication yield estimation analysis

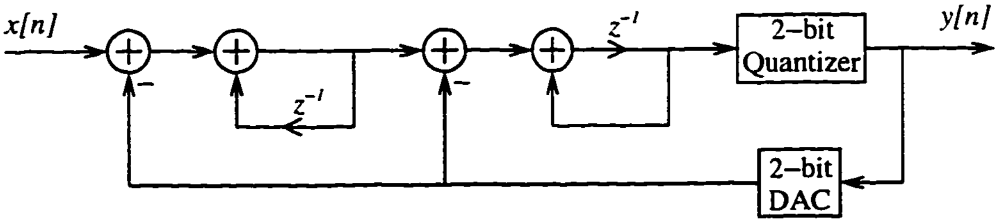


Figure 3.1: The architecture of the second-order ΔΣADC.

of a second-order ΔΣ data converter employing a noise-shaping DAC. Specifically, for the first-order noise-shaping DAC in [5], results are presented that allow for a theoretical prediction of the power spectral density (PSD) of the DAC noise given knowledge of the statistics of the static analog output level errors. The performance of an example 2-bit noise-shaping DAC employed in a ΔΣADC is demonstrated with simulation results, and a comparison between simulated and predicted PSDs is provided. An IC fabrication yield estimation of the power of the DAC noise residing in the signal-band of the ΔΣADC is presented. This data is subsequently used to generate an IC fabrication yield estimation of the overall conversion precision of the second-order ΔΣADC. For example, with an oversampling ratio of 128, the IC fabrication yield estimation shows that 90% of all ΔΣADCs fabricated with normally distributed static analog output level errors of standard deviation $\sigma = 1\%$ achieve a conversion precision of between 13.0 and 14.8 bits.

II. THE ARCHITECTURES OF THE ΔΣADC AND THE DAC

Many references exist on the analysis and implementation of the second-order ΔΣADC [1], [7]. Figure 3.1 shows the architecture of the particular second-order ΔΣADC to be considered in detail in this paper. The analog input and the digital output are denoted $x[n]$ and $y[n]$, respectively. The coarsely quantized ΔΣADC output is generated by a 2-bit uniform quantizer with quantization levels $0, \Delta, 2\Delta$, and 3Δ, where the *quantizer step-size*, Δ, is the spacing between the quantization

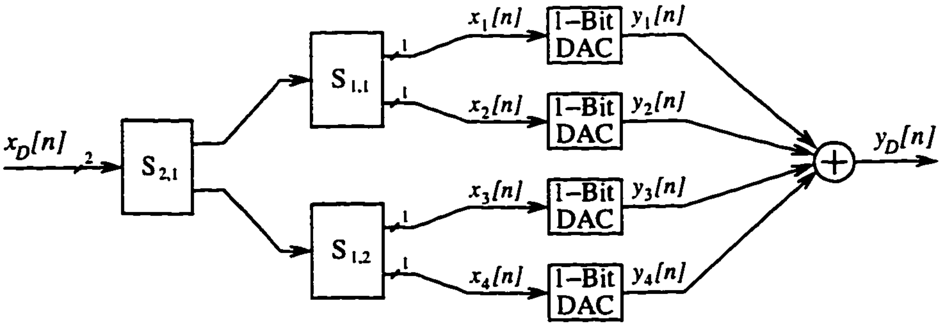


Figure 3.2: The architecture of the 2-bit noise-shaping DAC.

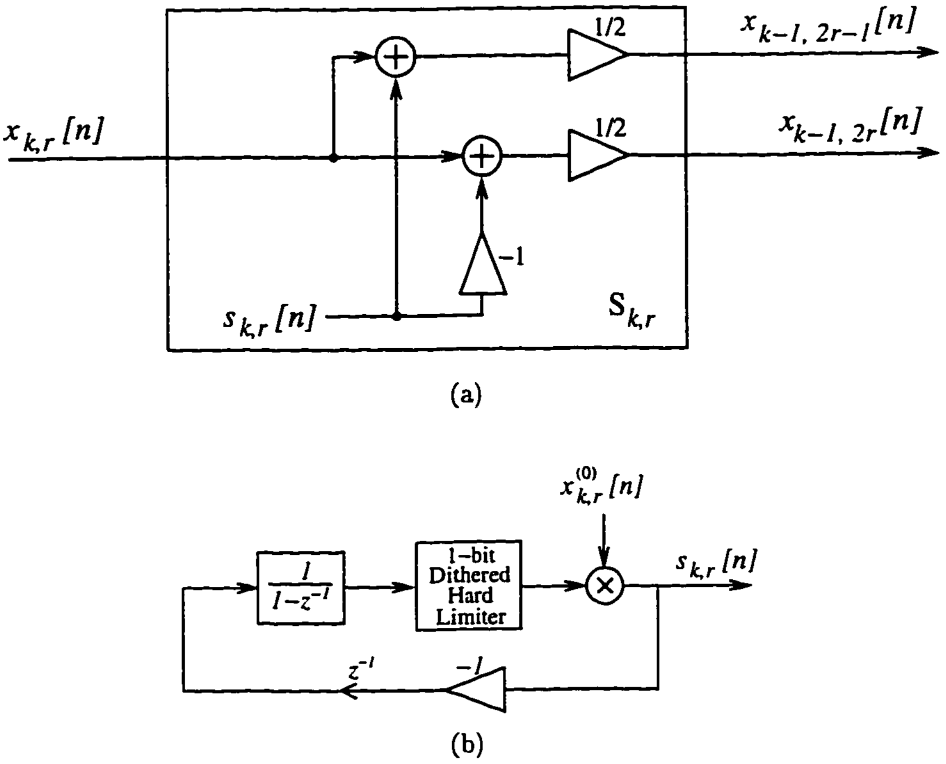
levels. It follows that the *quantizer no-overload range* is given by

$$(\delta_l, \delta_h) = (-\frac{1}{2}\Delta, \frac{7}{2}\Delta). \qquad (1)$$

A 2-bit version of the first-order noise-shaping DAC proposed in [5] is employed in the feedback path of the $\Delta\Sigma$ modulator loop. Figure 3.2 shows the architecture of the first-order noise-shaping DAC. For simplicity, the quantizer step-size Δ is taken to be unity in the following. The DAC input, $x_D[n]$, is identical to the $\Delta\Sigma$ADC output $y[n]$, and is thus a sequence of unsigned 2-bit binary numbers in the range 0 through 3. The DAC consists of 2 layers of digital devices, each referred to as a *switching block*, followed by 4 one-bit DACs, each referred to as a *unit DAC-element*, and an analog output summing node which generates the DAC output, $y_D[n]$. Each switching block is labeled $S_{k,r}$, where k denotes the layer number and r denotes the position of the switching block in the layer. The signal-processing details of the switching blocks are shown in Figure 3.3a. They have one digital input, $x_{k,r}[n]$, and two digital outputs, $x_{k-1,2r-1}[n]$ and $x_{k-1,2r}[n]$, respectively. The outputs are formed according to

$$x_{k-1,2r-1}[n] = \frac{1}{2}(x_{k,r}[n] + s_{k,r}[n]) \quad \text{and} \quad x_{k-1,2r}[n] = \frac{1}{2}(x_{k,r}[n] - s_{k,r}[n]), \qquad (2)$$

where $s_{k,r}[n]$ is a *switching sequence* generated according to the signal-processing algorithm of Figure 3.3b. The structure of Figure 3.3b closely resembles a first-order



(a)

(b)

Figure 3.3: (a) The general form of the switching blocks. (b) Details of the generation of the switching sequence $s_{k,r}[n]$.

$\Delta\Sigma$ modulator with no input signal, except for the LSB multiplier and the *one-bit dithered hard limiter.* The one-bit dithered hard limiter operates according to

$$v_{k,r}[n] = \begin{cases} 1, & \text{if } u_{k,r}[n] = 1; \\ -1, & \text{if } u_{k,r}[n] = -1; \\ d_{k,r}[n], & \text{if } u_{k,r}[n] = 0; \end{cases} \tag{3}$$

where $v_{k,r}[n]$ is the output of the device, $u_{k,r}[n]$ is the input of the device, and $d_{k,r}[n]$ is a random ± 1 sequence that is white, independent of $x_{k,r}[n]$, and uncorrelated with the $d_{k,r}[n]$ sequences in the other switching blocks.

A hardware-efficient gate-level implementation of the 2-bit DAC is suggested in Figure 3.4. A pair of flip-flops, FF_1 and FF_2, two tri-stateable buffers, a random binary sequence, $r_{k,r}[n]$, and two simple binary adders perform the switching block calculations defined in Figure 3.3. The random binary sequences $r_{k,r}[n]$ are white, mutually uncorrelated, and independent of the switching block inputs. Binary se-

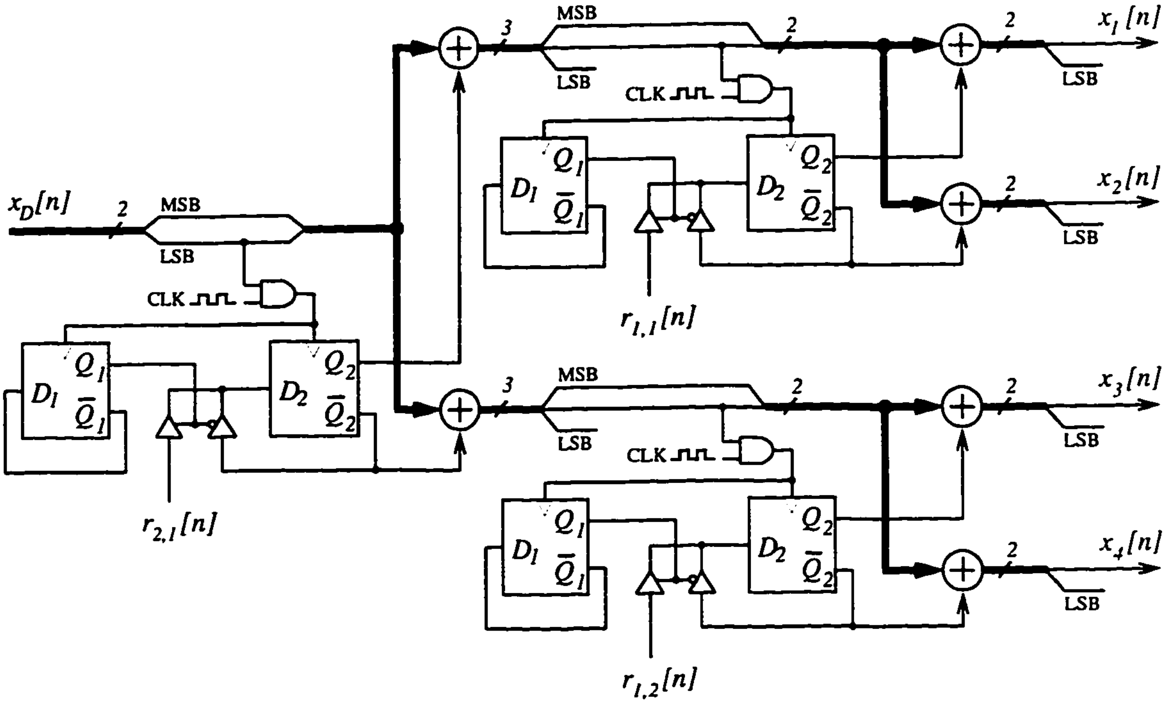


Figure 3.4: A hardware-efficient gate-level implementation of the 2-bit DAC.

quences that well approximate the desired statistics can be generated using simple feedback shift registers [8]. The least-significant bit (LSB) of the switching block input is used to gate the clock signal applied to the flip-flops. Initially, the buffer associated with $r_{k,r}[n]$ is enabled by setting the FF$_1$ output Q_1 to "1" so that Q_2 assumes a random value when FF$_2$ is clocked by the LSB. At the subsequent clocking of FF$_2$, the output Q_2 assumes the complementary value because of the feedback of $\overline{Q}_2$. This two-state process is repeated with additional clockings of FF$_1$ and FF$_2$. The binary adders form the sum of the switching block input and the outputs of FF$_2$. It can be verified that calculating these sums and subsequently right-shifting the results as indicated in Figure 3.4 has the effect of computing (2).

III. PERFORMANCE DETAILS

The analog section of the noise-shaping DAC of Figure 3.2 can be implemented with four switched-capacitor unit DAC-elements connected to the summing node of an integrator. During each sample period, the r^{th} unit DAC-element ideally transfers a unit amount of charge into the summing node if $x_r[n] = 1$, and no charge is transferred into the summing node if $x_r[n] = 0$. However, in practice, the unit DAC-elements operate according to

$$y_r[n] = \begin{cases} 1 + e_{h_r}, & \text{if } x_r[n] = 1; \\ e_{l_r}, & \text{if } x_r[n] = 0; \end{cases}$$

where $y_r[n]$ denotes the analog output of the r^{th} unit DAC-element, and e_{h_r} and e_{l_r} are errors in the analog output levels. The errors are assumed to be time-invariant, but otherwise arbitrary, and are referred to as *static DAC-element errors.*

As shown in [5], interconnecting the switching blocks and unit DAC-elements in the network of Figure 3.1 results in a DAC for which $y_D[n] = x_D[n]$ in the absence of static DAC-element errors. However, with non-zero static DAC-element errors, the DAC output has the form

$$y_D[n] = \alpha x_D[n] + \beta + e_D[n], \tag{4}$$

where α is a constant gain, β is a DC offset, and $e_D[n]$ is the DAC noise. The constants α and β depend only upon the static DAC-element errors; formulae for these parameters applicable to arbitrary DAC bit-resolution b have been derived in [5], and are repeated here for convenience as

$$\alpha = 1 + \frac{1}{2^b} \sum_{i=1}^{2^b} (e_{h_i} - e_{l_i}) \tag{5}$$

and

$$\beta = \sum_{i=1}^{2^b} e_{l_i}. \tag{6}$$

The DAC of Figure 3.2 achieves *first-order* noise-shaping by decorrelating $e_D[n]$ from $x_D[n]$ and spectrally shaping $e_D[n]$ such that its PSD ideally resembles that of white noise filtered by the highpass filter $H(z) = 1 - z^{-1}$.

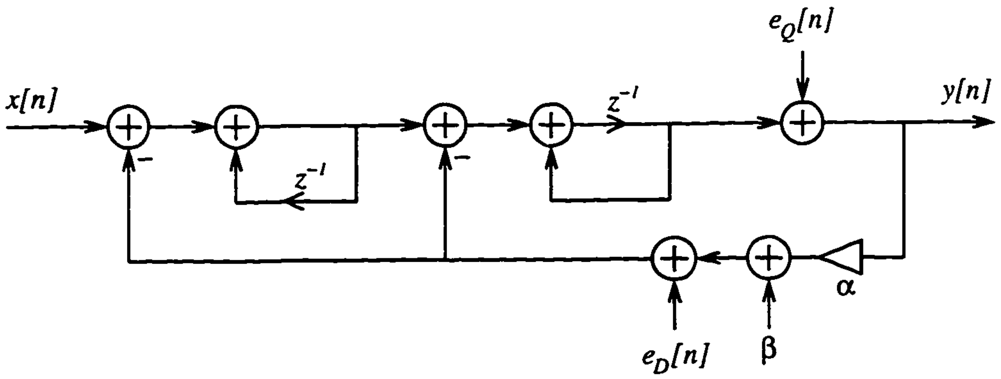


Figure 3.5: The signal-processing equivalent of the second-order ΔΣADC of Figure 3.1.

For the purposes of this paper, the ΔΣADC input $x[n]$ is assumed to be amplitude-limited to within the *ΔΣADC no-overload range*, defined as the range of input values for which the quantizer operates within its no-overload range (1). A conservative derivation of the ΔΣADC no-overload range is given in Appendix 3.A, the result of which is repeated here for convenience as

$$(\delta_l, \delta_h) = (\Delta, 2\Delta). \tag{7}$$

With this condition satisfied, the quantization noise, $e_Q[n]$, introduced by the quantizer is commonly modeled as an additive white-noise process, uncorrelated with the ΔΣADC input [1]. As shown in Appendix 3.C, $e_D[n]$ is uncorrelated with $x_D[n]$, and it follows that the signal-processing equivalent of the ΔΣADC of Figure 3.1 is given by Figure 3.5, where the signals $x[n]$, $e_Q[n]$, and $e_D[n]$ are uncorrelated. With a linear systems analysis it can be shown that the transfer functions between the ΔΣADC output $y[n]$ and the signals $x[n]$, $e_Q[n]$, and $e_D[n]$, are given by

$$\frac{Y(z)}{X(z)} = N_X(z)D(z) = z^{-1}D(z), \tag{8}$$

$$\frac{Y(z)}{E_Q(z)} = N_Q(z)D(z) = (1 - z^{-1})^2 D(z), \tag{9}$$

and

$$\frac{Y(z)}{E_D(z)} = N_D(z)D(z) = (z^{-1} - 2)z^{-1}D(z), \tag{10}$$

respectively, where the term

$$D(z) = [1 + 2(\alpha - 1)z^{-1} + (1 - \alpha)z^{-2}]^{-1} \tag{11}$$

can be viewed as a distortion of the ideal transfer functions $N_X(z), N_Q(z)$, and $N_D(z)$, resulting from a non-unity value of α. As will be demonstrated in the next section, the significance of $D(z)$ for practical values of α is vanishingly small. Notice that (10) reflects the importance of highpass shaping of $e_D[n]$; the magnitude of $N_D(z)$ increases monotonically from unity at low frequencies to three at high frequencies, and it follows that any DAC noise introduced in the frequencies corresponding to the signal-band appears unattenuated in the ΔΣADC output.

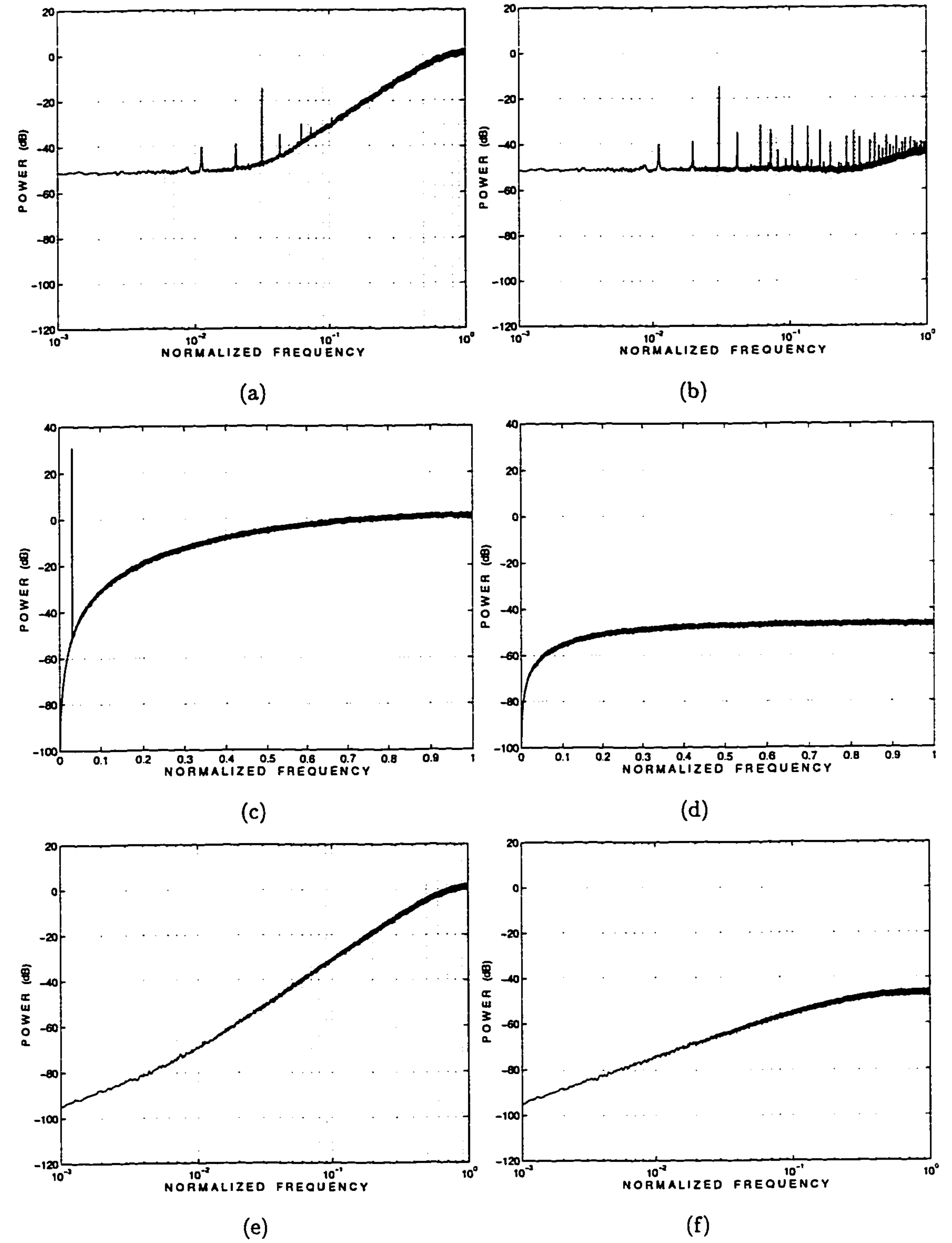
From Figure 3.5 it is seen that both of the noise-sources $e_Q[n]$ and $e_D[n]$ contribute to the *overall* conversion error at the output of the ΔΣADC. This error will subsequently be referred to as the *ΔΣADC noise*, $e_{ADC}[n]$. Using (8) and (11), the ΔΣADC noise can be written in the form

$$e_{ADC}[n] = y[n] + 2(\alpha - 1)y[n-1] + (1 - \alpha)y[n-2] - x[n-1]. \tag{12}$$

While this particular form is not very useful for a statistical analysis, it does provide a simple formula by which to compute $e_{ADC}[n]$ for simulation purposes.

A. Simulation Results

Simulated performance of the second-order ΔΣADC of Figure 3.1 is presented in Figure 3.6. Each plot in the figure shows the PSD in dB relative to Δ^2 of a particular signal of the ΔΣADC with a sinusoidal excitation. The PSDs were each calculated by averaging 100 length-2^{15} periodograms [9]. The frequency scales were normalized such that unity corresponds to half the sample rate of the ΔΣADC output. The static DAC-element errors of the 2-bit DAC were chosen randomly from a normal distribution with standard deviation $\sigma = 1\%$. It should be mentioned



(a) (b)

(c) (d)

(e) (f)

Figure 3.6: Simulated PSDs of the second-order ΔΣADC. (a) and (b), the ΔΣADC noise and the DAC noise, respectively, when employing a conventional 2-bit DAC, (c) and (d), the ΔΣADC output and the DAC noise, respectively, when employing the 2-bit noise-shaping DAC (plotted against a linear frequency scale), and (e) and (f), the ΔΣADC noise and the DAC noise, respectively, when employing the 2-bit noise-shaping DAC (plotted against a logarithmic frequency scale).

that this particular choice of static DAC-element errors was made for demonstration purposes only; the theoretical results presented in the next section do not depend upon any underlying statistical distribution or correlation properties of the static DAC-element errors.

Figures 3.6a and 3.6b show the $\Delta\Sigma$ADC noise $e_{ADC}[n]$ and the DAC noise $e_D[n]$, respectively, when employing a conventional DAC in the $\Delta\Sigma$ADC. The results were formed using (12) and (4), and are plotted against a logarithmic frequency scale normalized such that unity corresponds to half the sampling rate. Each of Figures 3.6c through 3.6f shows a particular signal of the $\Delta\Sigma$ADC when employing the first-order noise-shaping DAC. Specifically, Figure 3.6c shows the $\Delta\Sigma$ADC output $y[n]$, Figure 3.6d shows the DAC noise $e_D[n]$ formed using (4), Figure 3.6e shows the $\Delta\Sigma$ADC noise $e_{ADC}[n]$ formed using (12) and plotted against a logarithmic frequency scale, and Figure 3.6f shows $e_D[n]$ plotted against a logarithmic frequency scale.

Figures 3.6a and 3.6b clearly demonstrate the detrimental effects of the DAC noise when employing a conventional DAC in the $\Delta\Sigma$ADC. A considerable amount of the DAC noise resides in the signal-band, thereby severely decreasing the overall conversion performance. Furthermore, the occurrence of spurious tones limits the SFDR achievable by the $\Delta\Sigma$ADC. The results of Figures 3.6c through 3.6f demonstrate the spectral shaping of $e_{ADC}[n]$ and $e_D[n]$, respectively, when employing the first-order noise-shaping DAC in the $\Delta\Sigma$ADC. From Figure 3.6f it is seen that the DAC noise exhibits first-order behavior, i.e., a decrease by 6 dB of the noise-floor per octave decrease in frequency. Notice also the absence of spurious tones in the DAC noise; $e_D[n]$ and $x[n]$ are uncorrelated and $e_D[n]$ resembles highpass filtered white noise. Consequently, the SFDR of the $\Delta\Sigma$ADC is not limited by the DAC noise. Because of a considerable difference in the power-levels at high frequencies between

$e_Q[n]$ and $e_D[n]$, the $\Delta\Sigma$ADC noise of Figure 3.6e exhibits both first-order and second-order behavior. Specifically, while $e_Q[n]$ and $e_D[n]$ contribute with comparable powers at frequencies around $\omega_c = 10^{-2}$, the $\Delta\Sigma$ADC noise exhibits dominant first-order behavior for frequencies less than ω_c and dominant second-order behavior for frequencies higher than ω_c.

B. Performance Equations

For a statistical analysis of $e_{ADC}[n]$, notice that (9) and (10) imply that the PSD of $e_{ADC}[n]$ is given by

$$S_{e_{ADC}e_{ADC}}(e^{j\omega}) = S_{e_Qe_Q}(e^{j\omega})|N_Q(e^{j\omega})D(e^{j\omega})|^2 + S_{e_De_D}(e^{j\omega})|N_D(e^{j\omega})D(e^{j\omega})|^2 \tag{13}$$

where $S_{e_Qe_Q}(e^{j\omega})$ and $S_{e_De_D}(e^{j\omega})$ denote the PSDs of $e_Q[n]$ and $e_D[n]$, respectively. For the model of $e_Q[n]$ used in Figure 3.5,

$$S_{e_Qe_Q}(e^{j\omega}) = \frac{\Delta^2}{12}. \tag{14}$$

To evaluate (13), it remains to develop an expression for $S_{e_De_D}(e^{j\omega})$. This is done in detail in the appendices; the main result for a b-bit version of the DAC will merely be stated here as

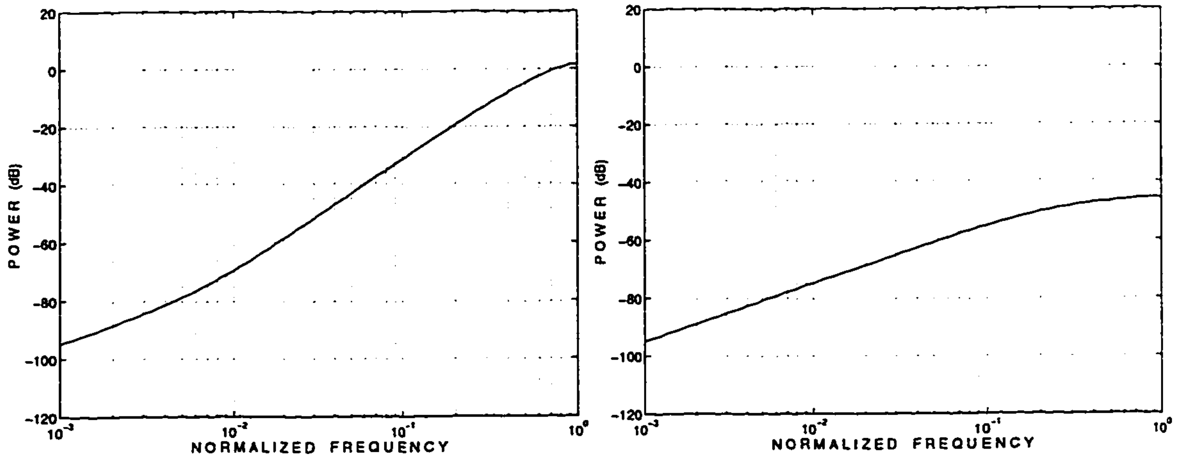
$$S_{e_De_D}(e^{j\omega}) = \sum_{k=1}^{b} \gamma_k S_k(e^{j\omega}), \tag{15}$$

where the constants γ_k are given by

$$\gamma_k = \sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2, \tag{16}$$

where

$$\Delta_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} \left[(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})\right]. \tag{17}$$



(a) (b)

Figure 3.7: Theoretically computed PSDs of the second-order ΔΣADC plotted against logarithmic scales. (a) The ΔΣADC noise, and (b) the DAC noise.

As shown in the appendices, the terms $S_k(e^{j\omega})$ occurring in (15) depend only upon the DAC input $x_D[n]$, and an algorithm to compute the $S_k(e^{j\omega})$ is provided.

Figure 3.7 presents the theoretically computed PSDs relative to Δ^2 of the ΔΣADC noise and the DAC noise. Specifically, Figure 3.7a shows $e_{ADC}[n]$ as computed using (13) and plotted against a logarithmic scale, and Figure 3.7b shows $e_D[n]$ as computed using (15) and plotted against a logarithmic scale. The plots are in agreement with the simulation results of Figure 3.6e and Figure 3.6f, respectively.

C. The ΔΣADC Conversion Precision

A standard measure of the ΔΣADC conversion precision is the power of the ΔΣADC noise with the restriction that the signal component of the ΔΣADC output $y[n]$ be an undistorted version of the ΔΣADC input $x[n]$. As asserted previously, $x[n]$ is subjected to the linear distortion $D(z)$ given by (11), whose deviation from unity depends only on α. Although—as will be demonstrated shortly—the effects of $D(z)$ on the ΔΣADC conversion precision can be neglected for practical values of α, it will be taken into account here for completeness.

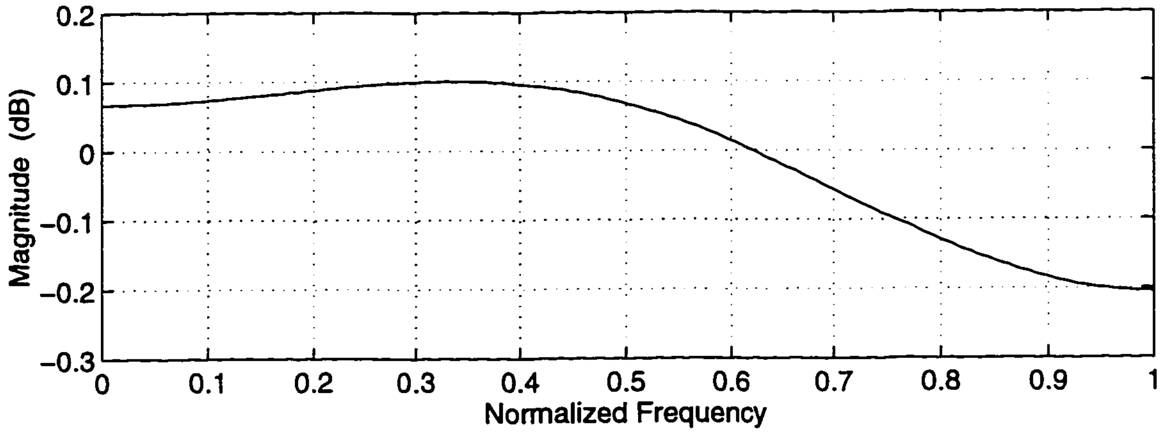


Figure 3.8: Typical magnitude response of the equalization filter $E(z)$.

It follows from (11) that the signal distortion can be compensated for using an *equalization filter* of the form

$$E(z) = [D(z)]^{-1} = 1 + 2(\alpha - 1)z^{-1} + (1 - \alpha)z^{-2}. \tag{18}$$

As can easily be verified, $E(1) = \alpha$ and $E(-1) = 4 - 3\alpha$. Writing α in the form $\alpha = 1 + \Delta_\alpha$, where Δ_α represents the deviation of α from unity, it follows that $1 - E(1) = -\Delta_\alpha$ and $1 - E(-1) = 3\Delta_\alpha$. Thus, the deviation of $|E(z)|$ from unity is three times greater at high frequencies than at low frequencies. Because of oversampling, the range of frequencies of interest is limited to low frequencies, and it follows that relatively little distortion is introduced inside the signal-band. Figure 3.8 illustrates typical behavior of the equalization filter over all frequencies. Specifically, the figure shows the magnitude response of $E(z)$ corresponding to $\alpha = 1.008$, which was the value of α of the simulated $\Delta\Sigma$ADC of Figure 3.6. The peak deviation of $|E(z)|$ from unity over all frequencies is 0.20 dB, whereas the peak deviation of $|E(z)|$ from unity for frequencies corresponding to an oversampling ratio of 2 or greater is merely 0.10 dB. Thus, it is likely that the signal distortion $D(z)$ need not be compensated for in a practical design, since it will have minimal effect on the $\Delta\Sigma$ADC conversion precision.

As found previously, the $\Delta\Sigma$ADC noise consists of additive components resulting from the DAC noise and the quantization noise. The power of the $\Delta\Sigma$ADC noise residing in the signal-band *after* signal equalization can therefore be written in the form

$$P_{e_{ADC}} = P_{e_D} + P_{e_Q}, \tag{19}$$

where P_{e_D} and P_{e_Q} denote the power of the DAC noise and the power of the quantization noise, respectively, residing in the signal-band after signal equalization. From (10), (15), and (18), it follows that

$$P_{e_D} = \sum_{k=1}^{b} \gamma_k P_{e_{D,k}}, \tag{20}$$

where

$$P_{e_{D,k}} = \frac{1}{2\pi} \int_{-\frac{\pi}{N}}^{\frac{\pi}{N}} S_k(e^{j\omega}) |N_D(e^{j\omega})|^2 \, d\omega, \tag{21}$$

and where N denotes the oversampling ratio corresponding to the signal-band of interest. Similarly, it follows from (9), (13), and (18) that

$$P_{e_Q} = \frac{\Delta^2}{24\pi} \int_{-\frac{\pi}{N}}^{\frac{\pi}{N}} |N_Q(e^{j\omega})|^2 \, d\omega. \tag{22}$$

It is customary to determine the conversion bit-precision of the $\Delta\Sigma$ADC by the number of bits that a uniform quantizer would require to generate quantization noise of power equal to $P_{e_{ADC}}$. A commonly used formula for this purpose is

$$R = \frac{1}{2}\log_2 \left[\frac{\sigma^2}{3P_{e_{ADC}}} \right], \tag{23}$$

where R is the number of bits, σ is determined from the no-overload range (δ_l, δ_h) of the ADC as

$$\sigma = \frac{(\delta_h - \delta_l)}{2}, \tag{24}$$

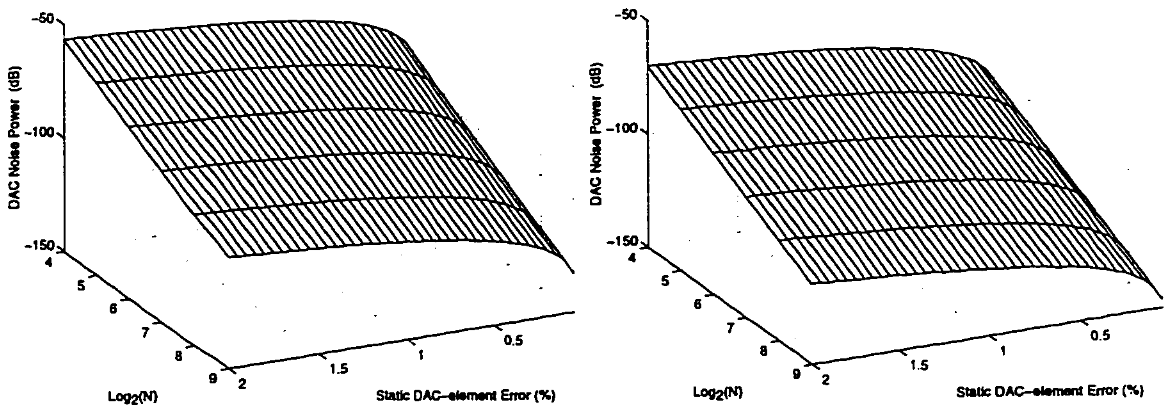
and $P_{e_{ADC}}$ is the power of the $\Delta\Sigma$ADC noise residing in the signal band [10]. The significance of employing a multi-bit quantizer rather than a 1-bit quantizer in the $\Delta\Sigma$ADC becomes apparent from (23) and (24). For example, the $\Delta\Sigma$ADC no-overload range (7) resulting from the use of a 2-bit quantizer is at least five-to-six times larger than the no-overload range of the $\Delta\Sigma$ADC when employing a 1-bit quantizer [6]. Ignoring the effects of the DAC noise, it follows from (23) that the improvement of the $\Delta\Sigma$ADC conversion precision is at least 2.5 bits.

IV. IC FABRICATION YIELD ESTIMATION

With knowledge of the statistical distribution of the static DAC-element errors, the results of the previous section allow for an IC fabrication yield estimation of the power of DAC noise residing in the signal-band and the overall $\Delta\Sigma$ADC conversion precision. The IC fabrication yield estimation procedure used in the following is based upon the idea of computing a large number of samples of the parameter of interest for a given statistical distribution of the static DAC-element errors, thereby generating data that closely resemble the corresponding statistical distribution of the parameter.

An algorithm to compute an IC fabrication yield estimation of the power of the DAC noise residing in the signal-band, P_{e_D}, is as follows:

1. Calculate (21) for the oversampling ratio of interest. This quantity is independent of the static DAC-element errors, and need thus only be calculated once.
2. Choose the values of the static DAC-element errors given knowledge of the underlying statistical distribution.
3. Calculate the γ_k using (16).
4. Calculate P_{e_D} using (20) and the data from step 1.
5. Repeat steps 2 through 4 for a number of samples large enough to employ the law of large numbers to generate data that accurately resemble the corresponding statistical distribution of P_{e_D}.

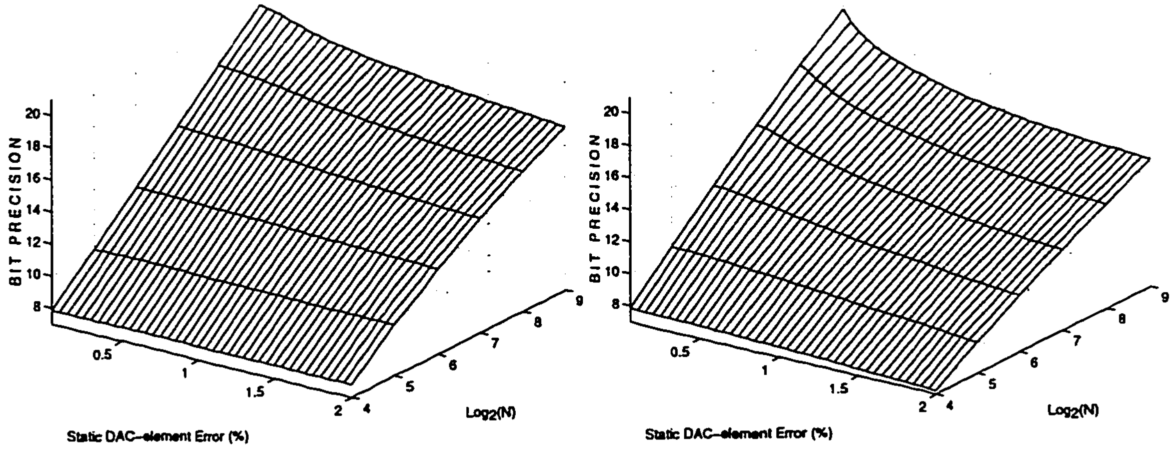


(a) (b)

Figure 3.9: An IC fabrication yield estimation of the power of the DAC noise residing in the signal-band, P_{e_D}, of the 2-bit noise-shaping DAC. The static DAC-element errors were chosen from a normal distribution with standard deviation σ ranging from 0.05% to 2%, and the oversampling ratio ranged from 16 to 512. Plots (a) and (b) show the largest of the smallest 95% and 5% values of P_{e_D}, respectively.

An example IC fabrication yield estimation of P_{e_D} is shown in Figure 3.9. Specifically, the figure shows P_{e_D} in dB relative to Δ^2 of the 2-bit DAC as a function of N and the static DAC-element errors. The oversampling ratio ranged from 16 to 512, and the static DAC-element errors were chosen randomly from a normal distribution with standard deviation σ ranging from 0.05% to 2%. Each IC fabrication yield estimation was based upon 10^4 calculated samples. Figures 3.9a and 3.9b show the largest of the smallest 95% and 5% values of P_{e_D}, respectively. For example, with $N = 128 = 2^7$ and static DAC-element errors with standard deviation $\sigma = 1\%$, Figure 3.9a predicts that 95% of all 2-bit DACs have signal-band noise powers less than -89.2 dB relative to Δ^2, while Figure 3.9b predicts that merely 5% of all 2-bit DACs have signal-band noise powers less than -102.6 dB relative to Δ^2. Thus, 90% of all 2-bit DACs fabricated with normally distributed static DAC-element errors of standard deviation $\sigma = 1\%$ satisfy

$$-89.2 \text{ dB} \leq P_{e_D} \leq -102.6 \text{ dB}. \tag{25}$$



(a) (b)

Figure 3.10: An IC fabrication yield estimation of the $\Delta\Sigma$ADC conversion bit-precision, R. The static DAC-element errors were chosen from a normal distribution with standard deviation σ ranging from 0.05% to 2%, and the oversampling ratio ranged from 16 to 512. (a) and (b) show the largest of the smallest 95% and 5% values of R, respectively.

relative to Δ^2. For the purpose of comparing theoretical results with simulated results, numerically integrating the simulated PSD of $e_D[n]$ of Figure 3.6d over the range of frequencies corresponding to an oversampling ratio of 128 yields $P_{e_D} = -93.7$ dB relative to Δ^2, in support of (25).

The IC fabrication yield estimation of P_{e_D} can be used to generate an IC fabrication yield estimation of the overall $\Delta\Sigma$ADC conversion precision using the following algorithm:

1. Calculate (22) for the oversampling ratio of interest. P_{e_Q} is independent of the static DAC-element errors, and need thus only be calculated once.
2. To each IC fabrication yield estimation datum of P_{e_D} obtained in the above, add the quantity from step 1 to obtain the power of the $\Delta\Sigma$ADC noise, $P_{e_{ADC}}$ relative to Δ^2.
3. The $\Delta\Sigma$ADC conversion bit-precision corresponding to $P_{e_{ADC}}$ is determined from (23), (24), and (7).

An IC fabrication yield estimation of the $\Delta\Sigma$ADC conversion bit-precision is shown in Figure 3.10. For the same example as in the above, Figure 3.10a predicts that 95% of all $\Delta\Sigma$ADCs have conversion precisions less than 14.79 bits, while Figure 3.10b predicts that merely 5% of all $\Delta\Sigma$ADCs have conversion precisions less than 13.00 bits. It follows that 90% of all $\Delta\Sigma$ADCs fabricated with normally distributed static DAC-element errors of standard deviation $\sigma = 1\%$ satisfy

$$13.00 \text{ bits} \leq R \leq 14.79 \text{ bits}. \tag{26}$$

Again, for the purpose of comparing theoretical results with simulated results, numerically integrating the simulated PSD of $e_{ADC}[n]$ of Figure 3.6d over the range of frequencies corresponding to an oversampling ratio of 128 yields $P_{e_{ADC}} = -93.3$ dB relative to Δ^2. This value of $P_{e_{ADC}}$ is equivalent to a conversion precision of $R =$ 13.70 bits, in support of (26). To conclude the comparison with the performance of the conventional DAC, it should also be mentioned that numerically integrating the noise-floor of Figure 3.6b over the range of frequencies corresponding to an oversampling ratio of 128 yields $P_{e_{ADC}} = -71.6$ dB relative to Δ^2, equivalent to a conversion precision of $R = 10.10$ bits. Thus, for this example, the use of the first-order noise-shaping DAC increases the $\Delta\Sigma$ADC conversion precision by more than 3 bits, corresponding to about 30%.

The result in (26) can be used to quantify the detrimental effect of the DAC noise of the 2-bit noise-shaping DAC on the $\Delta\Sigma$ADC conversion precision. Specifically, evaluating (22) for $N = 128$ yields $P_{e_Q} = -103.3$ dB relative to Δ^2. In the absence of DAC noise, $P_{e_{ADC}} = P_{e_Q}$, and the corresponding conversion precision found using (23) is 15.36 bits. Thus, for 90% of all $\Delta\Sigma$ADCs, the detrimental effect, ΔR, of the DAC noise on the conversion bit-precision satisfies

$$-2.36 \text{ bits} \leq \Delta R \leq -0.57 \text{ bits}. \tag{27}$$

As demonstrated by Figure 3.6 and Figure 3.7, the $\Delta\Sigma$ADC noise exhibits both first-order and second-order behavior, resulting from the effects of $e_D[n]$ and $e_Q[n]$,

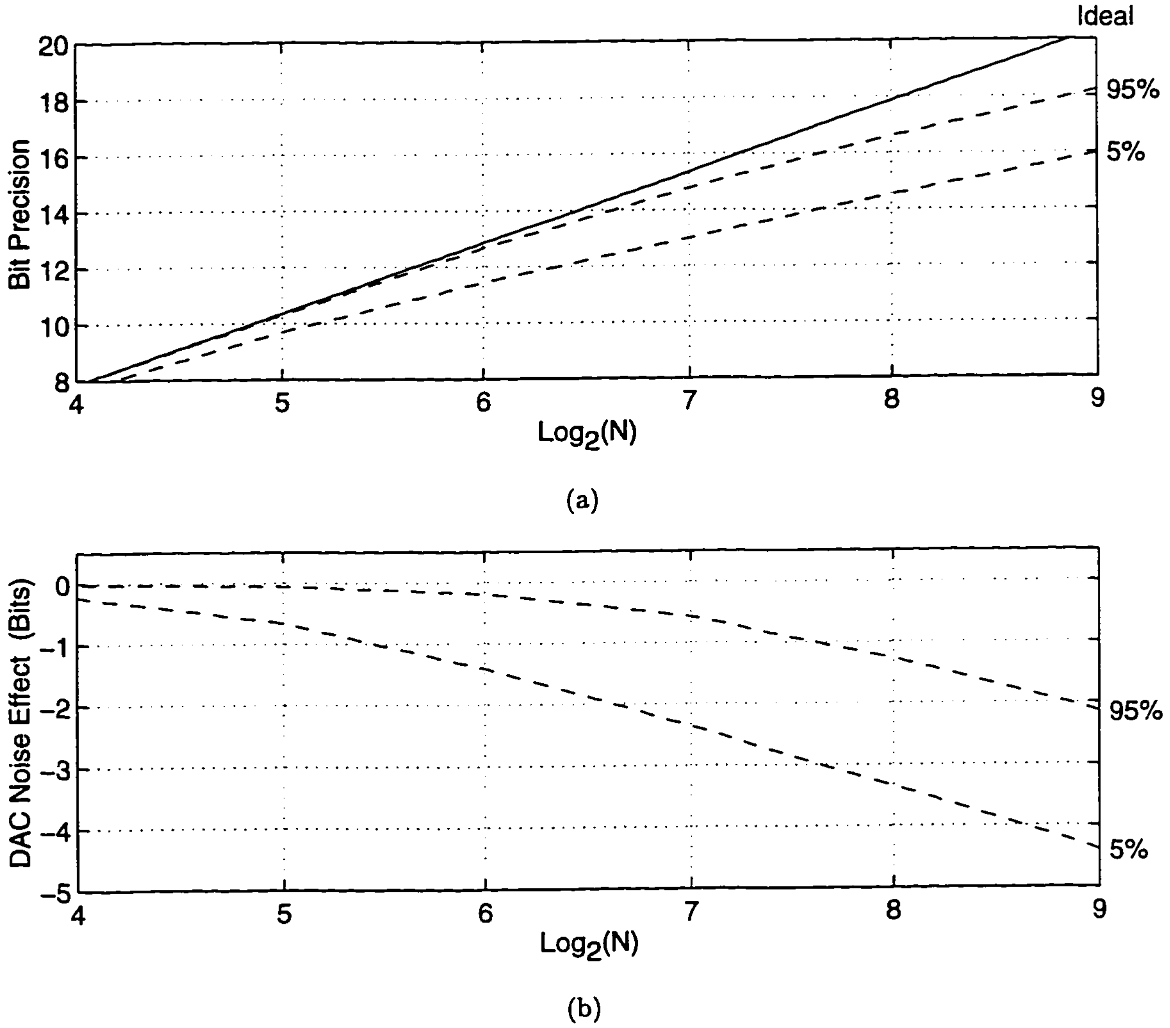


Figure 3.11: A quantification of the detrimental effects of the DAC noise. (a) The ideal ΔΣADC conversion bit-precision (solid line), and the IC fabrication yield estimation of Figure 3.10 corresponding to static DAC-element errors with standard deviation $\sigma = 1\%$ (dashed lines). (b) An IC fabrication yield estimation of the detrimental effect (measured in bits) of the DAC noise on the ΔΣADC conversion precision corresponding to static DAC-element errors with standard deviation $\sigma = 1\%$.

respectively. Exactly which term dominates $P_{e_{ADC}}$ depends upon the signal-band under consideration but, in general, the effects of $e_D[n]$ dominates at high oversampling ratios, and the effects of $e_Q[n]$ dominates at low oversampling ratios. It follows that the detrimental effect of the DAC noise on the ΔΣADC conversion precision varies with the oversampling ratio. As an example, the dashed lines in Figure 3.11a show the IC fabrication yield estimation of Figure 3.10 for the case of static DAC-element errors with standard deviation $\sigma = 1\%$ and oversampling ratio ranging from 16 to 512. Also shown (solid line) is the ideal ΔΣADC performance, i.e., the conversion bit-precision achieved in the absence of DAC noise. Figure 3.11b quantifies the effect of the DAC noise on the ΔΣADC conversion bit-precision corresponding to the data of Figure 3.11a. Specifically, the dashed lines correspond to the bounding limits of ΔR calculated by forming the difference between each of the dashed lines in Figure 3.11a and the solid line in Figure 3.11a, respectively. For example, the figure shows that for 90% of all ΔΣADCs operating at an oversampling ratio of 512, the detrimental effect of the DAC noise on the conversion precision is bounded by

$$-4.39 \text{ bits} \;\leq\; \Delta R \;\leq\; -2.14 \text{ bits},$$

whereas for $N = 16$, the detrimental effect of the DAC noise is merely

$$-0.23 \text{ bits} \;\leq\; \Delta R \;\leq\; -0.01 \text{ bits}.$$

VI. CONCLUSION

The performance of a second-order ΔΣADC employing a multi-bit first-order noise-shaping DAC proposed in [5] has been analyzed in detail. Results that allow for a theoretical quantification of the effects of the DAC noise on the ΔΣADC noise have been developed. It was demonstrated how knowledge of the statistics of the static DAC-element errors allowed for an IC fabrication yield estimation analysis of the ΔΣADC conversion precision.

Focus was brought upon an example ΔΣADC employing a 2-bit noise-shaping DAC fabricated with normally distributed static DAC-element errors of standard deviation $\sigma = 1\%$. To the extend that performance is limited by static DAC element errors, it was found that with an oversampling ratio of 128, the conversion precision provided by 90% of all fabricated ΔΣADCs is greater than 13.0 bits but less than 14.8 bits. The detrimental effect of the DAC noise on the $\Delta\Sigma$ADC conversion precision was quantified to be greater than -2.4 bits, but less than -0.6 bits.

Other non-ideal circuit behavior such as clock-skew, non-linear settling, and finite slew-rates may impose additional limitations on the achievable conversion precision of the second-order $\Delta\Sigma$ADC. Although the effects of these error sources have not been considered in this paper, the results presented here still remain valid as upper bounds on the performance of the data converter, independent of the particular circuit technology used in an implementation.

APPENDIX 3.A

Claim A. *The quantizer of the $\Delta\Sigma ADC$ of Figure 3.1 operates within the no-overload range at all times provided* (7) *is satisfied.*

Proof. Recall that a uniform quantizer with input $x_Q[n]$, no-overload range (δ_l, δ_h), and step-size Δ operates such that the quantization noise $e_Q[n]$ satisfies

$$|e_Q[n]| \leq \frac{\Delta}{2}, \text{ if } x_Q[n] \in (\delta_l, \delta_h), \qquad \text{and} \qquad |e_Q[n]| > \frac{\Delta}{2}, \text{ if } x_Q[n] \notin (\delta_l, \delta_h). \tag{28}$$

Assuming that $\alpha = 1$ in Figure 3.5, it can be verified that the transfer function between $x[n]$ and $x_Q[n]$ is

$$\frac{X_Q(z)}{X(z)} = S(z) = z^{-1}, \tag{29}$$

and that the transfer function between $e_Q[n]$ and $x_Q[n]$ is

$$\frac{X_Q(z)}{E_Q(z)} = Q(z) = -z^{-1}(2 - z^{-1}). \tag{30}$$

It follows from (29) and (30) that $x_Q[n]$ depends on the previous sample of $x[n]$ and the previous two samples of $e_Q[n]$. Claim A can now be shown by induction as follows.

To establish the induction basis, consider sample-time $n = 1$. Suppose that the initial condition of both integrators in Figure 3.5 is zero. Then $x_Q[0] = 0$, and consequently $e_Q[0] = 0$. With the assumption of (7) being satisfied, $x_Q[1] \in (\Delta,\ 2\Delta)$, and $|e_Q[1]| \leq \frac{\Delta}{2}$. It follows that no overload occurs at sample-time $n = 1$.

Next, consider the sample-time $n + 1$. By the induction hypothesis, no overload has occurred previously, i.e., $|e_Q[k]| \leq \frac{\Delta}{2}$ for $k = 0, 1, \ldots, n-1, n$. It follows from (30) that

$$|q[n+1]| \leq |2e_Q[n]| + |e_Q[n-1]| \leq 3\frac{\Delta}{2},$$

where $q[n+1]$ denotes the output of the filter $Q(z)$ at sample-time $n + 1$. Thus,

$$x[n] - \frac{3\Delta}{2} \leq x_Q[n+1] \leq x[n] + \frac{3\Delta}{2}.$$

It follows from (1) that quantizer overload is avoided if

$$x[n] - \frac{3\Delta}{2} > -\frac{\Delta}{2} \qquad \text{and} \qquad x[n] + \frac{3\Delta}{2} < \frac{7\Delta}{2},$$

which can be re-arranged as (7). ■

APPENDIX 3.B

The purpose of this appendix is to present a derivation of the bulk of the results regarding the PSD of the output of the noise-shaping DAC as stated in Section III. A few helpful definitions are first given.

In accordance with the usual definitions, let the *statistical mean* and the *time average* of a sequence $x[n]$ be defined as

$$m_x[n] = \mathrm{E}\{x[n]\} \tag{31}$$

and

$$\overline{M}_x = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n], \tag{32}$$

respectively, where $\mathrm{E}\{\cdot\}$ denotes the statistical expectation operator. Furthermore, let the *statistical autocorrelation* and the *time average autocorrelation* of $x[n]$ be given by

$$R_{xx}[n, m] = \mathrm{E}\{x[n]x[n+m]\} \tag{33}$$

and

$$\overline{R}_{xx}[m] = \lim_{P\to\infty} \frac{1}{P} \sum_{n=1}^{P} x[n]x[n+m], \tag{34}$$

respectively. The *time average power spectral density*—referred to here as the *PSD*—is defined as the Fourier transform of (34), i.e.,

$$S_{xx}(e^{j\omega}) = \sum_{m=-\infty}^{\infty} \overline{R}_{xx}[m]e^{-j\omega m}.$$

The main theoretical result of this paper can now be stated as follows:

Claim B. *Consider the b-bit version of the first-order noise-shaping DAC of Figure 3.2 driven by a deterministic input $x_D[n]$, for which $\overline{M}_{x_D}$ and $\overline{R}_{x_Dx_D}[m]$ exists. The PSD of the DAC output $y_D[n]$ can be written in the form*

$$S_{y_Dy_D}(e^{j\omega}) = \alpha^2 S_{x_Dx_D}(e^{j\omega}) + S_{e_De_D}(e^{j\omega}) + 2\pi\overline{\eta}\delta(\omega), \tag{35}$$

where α is given by (5), and

$$\overline{\eta} = 2\alpha\beta\overline{M}_{x_D} + \beta^2, \tag{36}$$

where β is given by (6). The sequence $e_D[n]$ is a zero-mean random process such that $S_{e_De_D}(e^{j\omega})$ has the form stated in (15).

Proof. To verify (35), consider first the statistical autocorrelation of $y_D[n]$. From (4) it follows that

$$R_{y_Dy_D}[n,m] = \mathrm{E}\left\{(\alpha x_D[n] + \beta + e_D[n])\,(\alpha x_D[n+m] + \beta + e_D[n+m])\right\}. \tag{37}$$

As shown in Appendix 3.C, $e_D[n]$ is a zero-mean random process, i.e., $m_{e_D}[n] = 0$. Making use of this and the fact that $x_D[n]$ is deterministic, (37) can be rewritten in the form

$$R_{y_Dy_D}[n,m] = \alpha^2 x_D[n]x_D[n+m] + R_{e_De_D}[n,m] + \alpha\beta(x_D[n] + x_D[n+m]) + \beta^2. \tag{38}$$

Using (32), (34), and (36),

$$\lim_{P\to\infty}\frac{1}{P}\sum_{n=1}^{P} R_{y_Dy_D}[n,m] = \alpha^2\overline{R}_{x_Dx_D}[m] + \lim_{P\to\infty}\frac{1}{P}\sum_{n=1}^{P} R_{e_De_D}[n,m] + \overline{\eta}.$$

Arguments similar to the ones presented in the proof of the time average results in [11] establishes

$$\overline{R}_{y_Dy_D}[m] = \alpha^2\overline{R}_{x_Dx_D}[m] + \overline{R}_{e_De_D}[m] + \overline{\eta} \tag{39}$$

with probability 1. Fourier transforming (39) then yields (35). To complete the proof of Claim B, it remains to develop an expression for $R_{e_De_D}[n,m]$.

As shown in [5], the DAC noise has the form

$$e_D[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n], \tag{40}$$

where the $\Delta_{k,r}$ are constants given by (17). It follows that

$$\begin{aligned} R_{e_D e_D}[n,m] &= \mathrm{E}\{e_D[n] e_D[n+m]\} \\ &= \mathrm{E}\left\{ \left(\sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n] \right) \left(\sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n+m] \right) \right\}. \end{aligned}$$

As shown in Appendix 3.C, the switching sequences $s_{k,r}[n]$ are mutually uncorrelated, and therefore

$$R_{e_D e_D}[n,m] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r}^2 R_{s_{k,r} s_{k,r}}[n,m].$$

As also shown in Appendix 3.C, the autocorrelations of the switching sequences within each layer are equal. Thus, with the compact notation

$$R_{s_{k,r} s_{k,r}}[n,m] \equiv R_k[n,m],$$

it follows that

$$R_{e_D e_D}[n,m] = \sum_{k=1}^{b} \gamma_k R_k[n,m],$$

where the γ_k are defined in (16). Thus,

$$\overline{R}_{e_D e_D}[m] = \sum_{k=1}^{b} \gamma_k \overline{R}_k[m].$$

The time-average autocorrelation functions $\overline{R}_k[m]$ depend only upon $x_D[n]$, and an algorithm for their calculation is given in Appendix 3.D. ■

APPENDIX 3.C

This appendix presents derivations of the statistical properties of the DAC noise asserted in Appendix 3.B.

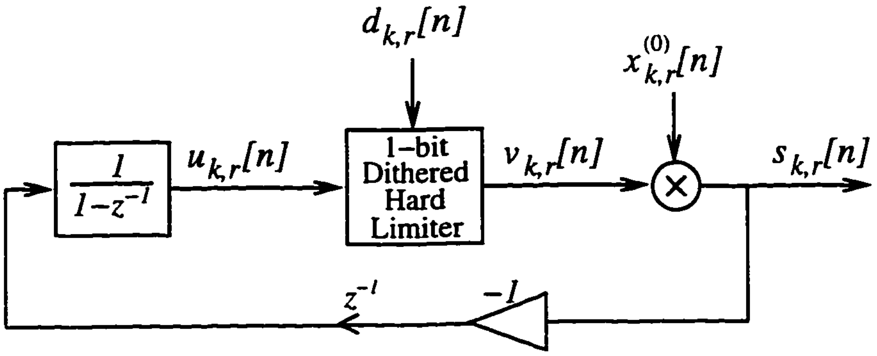


Figure 3.12: The details of the generation of the switching sequences $s_{k,r}[n]$.

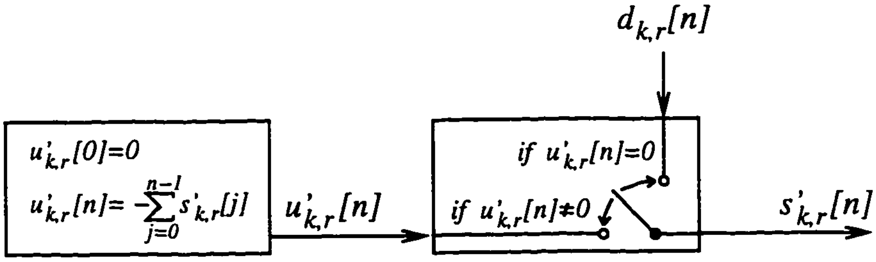


Figure 3.13: A signal-processing equivalent of Figure 3.12 when $x^{(0)}_{k,r}[n] = 1$.

Claim C. *The DAC noise $e_D[n]$ given by (40) is a zero-mean random process. The switching sequences $s_{k,r}[n]$ are mutually uncorrelated, and the statistical autocorrelations of the $s_{k,r}[n]$ within each layer are identical, i.e., $R_{s_{k,r}s_{k,r}}[n,m] = R_k[n,m]$.*

Proof. To show that $e_D[n]$ is zero-mean, notice that it follows from (40) that it suffices to show that the switching sequences $s_{k,r}[n]$ are zero-mean. For that purpose, consider again the details of the generation of the $s_{k,r}[n]$, as depicted in Figure 3.12. The operation of the 1-bit dithered hard limiter is defined in terms of the sequences $d_{k,r}[n]$, $v_{k,r}[n]$, and $u_{k,r}[n]$ as given by (3). It follows from the figure that $s_{k,r}[n] = 0$ if $x^{(0)}_{k,r}[n] = 0$. Thus, the claim is trivial for $x^{(0)}_{k,r}[n] = 0$. As is easy to verify, if $x^{(0)}_{k,r}[n] = 0$, then $u_{k,r}[n+1] = u_{k,r}[n]$. It follows that for non-zero values of $x^{(0)}_{k,r}[n]$, the system of Figure 3.12 is equivalent to the system of Figure 3.13, where $s'_{k,r}[n]$ consists of the non-zero samples of $s_{k,r}[n]$ in Figure 3.12, and where $u'_{k,r}[0] = 0$ indicates an initial condition of zero of the discrete-time integrator. From

Figure 3.13 it is easy to verify that for $n \geq 0$,

$$s'_{k,r}[n] = \begin{cases} d_{k,r}[n], & \text{if } n \text{ is even;} \\ -d_{k,r}[n-1], & \text{else;} \end{cases} \tag{41}$$

Since $d_{k,r}[n]$ is zero-mean, it follows that $s'_{k,r}[n]$ is zero-mean. Thus, $s_{k,r}[n]$ is zero-mean, as asserted.

To show that the $s_{k,r}[n]$ are mutually uncorrelated, notice that it follows from the previous that it suffices to show $\mathrm{E}\left\{s_{k,r}[n]s_{j,q}[n]\right\} = 0$, for $k \neq j$ and $r \neq q$. This holds trivially when $s_{k,r}[n] = 0$ and/or $s_{j,q}[n] = 0$. It follows from (41) that the sequence of samples of $s_{k,r}[n]$ where both $s_{k,r}[n] \neq 0$ and $s_{j,q}[n] \neq 0$ can be written in the form

$$s''_{k,r}[n] = z_{k,r}[n],$$

where $z_{k,r}[n]$ is zero-mean since $d_{k,r}[n]$ is zero-mean. Similarly,

$$s''_{j,q}[n] = z_{j,q}[n],$$

where $z_{j,q}[n]$ is zero-mean. Since the sequences $d_{k,r}[n]$ and $d_{j,q}[n]$ are mutually uncorrelated, $z_{k,r}[n]$ and $z_{j,q}[n]$ are mutually uncorrelated, and thus

$$\mathrm{E}\left\{s''_{k,r}[n]s''_{j,q}[n]\right\} = 0,$$

and it follows that $\mathrm{E}\left\{s_{k,r}[n]s_{j,q}[n]\right\} = 0$, as asserted.

To show that the statistical autocorrelations of the $s_{k,r}[n]$ within each layer are equal, notice that it follows from the above that the *probability density function* (PDF) of $s_{k,r}[n]$ fully depends upon the PDF of $x^{(0)}_{k,r}[n]$. Thus, the PDF of $s_{k,r}[n]$ fully depends upon the PDF of $x_{k,r}[n]$. Since the discrete-time integrators all have equal (zero) initial conditions, it suffices to show that the PDFs of the $x_{k,r}[n]$ within each layer are equal. This will be shown by induction.

To establish the induction basis, notice that the claim obviously holds true for $k = b$. Next, suppose the claim holds true for $k = b, b-1, \ldots, j+1, j$, where $j > 1$.

It must be shown that the claim holds true for $k = j-1$. It follows from the defining equations

$$x_{j-1,2r-1}[n] = \frac{1}{2}(x_{j,r}[n] + s_{j,r}[n]) \quad \text{and} \quad x_{j-1,2r}[n] = \frac{1}{2}(x_{j,r}[n] - s_{j,r}[n]), \tag{42}$$

that $x_{j-1,2r-1}[n]$ and $x_{j-1,2r}[n]$ have equal PDFs by invoking the induction hypothesis and by noticing that it follows from (41) that $s_{j,r}[n]$ is either 1 or -1 with equal probability, or identically zero. This is sufficient to show the claim, since a recursive argument can be used to establish that the remaining $x_{j-1,i}[n]$ sequences of the $j-1^{\text{st}}$ layer have PDFs that are equal to the PDFs of $x_{j-1,2r-1}[n]$ and $x_{j-1,2r}[n]$. ■

APPENDIX 3.D

This section describes an algorithm by which to compute the time-average autocorrelation functions $\overline{R}_k[m]$.

It follows from the discussion in Appendix 3.C that a state transition diagram for $s_{k,r}[n]$ can be depicted as in Figure 3.14. The states correspond to the possible values of the discrete-time integrator output, and $p_{k,r}[n]$ is the probability that $x_{k,r}[n]$ equals one, i.e., $p_{k,r}[n] = P\{x_{k,r}^{(0)}[n] = 1\}$. For example, transition from state "0" to state "-1" can only occur when $x_{k,r}^{(0)}[n] = 1$, in which case the state transition probability is $\frac{p_{k,r}[n]}{2}$. Transition to the state "1" is equally likely. By the recursive nature of the DAC architecture, $p_{k,r}[n]$ depends upon the switching sequences in layers $k, k+1, \ldots, b$, which, in turn, depend upon input bits $x^{(0)}[n], x^{(1)}[n], \ldots, x^{(k-1)}[n]$. Thus, $p_{k,r}[n]$ is time-varying and the state transition probabilities are therefore time-varying.

When considering the limiting properties of the resulting state transition chain, it is useful to divide the chain into *non-recurrent* state transition paths and *recurrent* state transition paths. Notice that it follows from the symmetry of the state transition diagram that state "0" is a recurrent state of all recurrent state transition

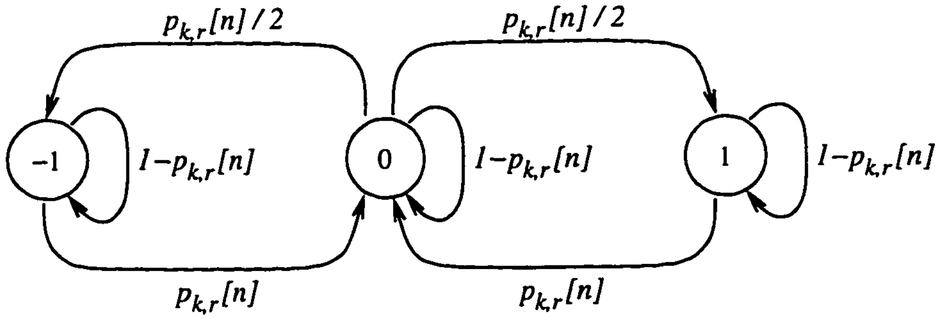


Figure 3.14: A state transition diagram for $s_{k,r}[n]$. The states correspond to the possible values of the discrete-time integrator output, and $p_{k,r}[n]$ is the probability that $x_{k,r}^{(0)}[n]$ equals one, i.e., $p_{k,r}[n] = P\{x_{k,r}^{(0)}[n] = 1\}$.

paths. A useful result to be used in the following is stated next.

Claim D. *Suppose the state transition chain is started in state "0" at time $n = 0$. Let $t_1, t_2, \ldots$ be the times when all recurrent state transition paths return to state "0". Then $\mathrm{E}\{s_{k,r}[0]s_{k,r}[m]\} = 0$ for all $m \geq t_1$.*

Proof. From the state transition diagram of Figure 3.14 it follows that $s_{k,r}[0]$ is either -1 or 1 with probability $\frac{p_{k,r}[0]}{2}$, or identically zero. Similarly, $s_{k,r}[t_1]$ is either -1 or 1 with probability $\frac{p_{k,r}[t_1]}{2}$, or identically zero. Since the possible transitions to non-zero states at time t_1 are controlled by $d_{k,r}[n]$, they occur *independently* of the possible transitions to non-zero states at time 0. Thus,

$$\begin{aligned}\mathrm{E}\{s_{k,r}[0]s_{k,r}[t_1]\} &= \Big[(-1)(-1) + (-1)(1) + (1)(-1) + (1)(1)\Big]\frac{p_{k,r}[0]}{2}\frac{p_{k,r}[t_1]}{2} \\ &= 0\end{aligned} \tag{43}$$

Similar reasoning verifies $\mathrm{E}\{s_{k,r}[0]s_{k,r}[m]\} = 0$ for $m > t_1$. ■

To allow for the computation of a state transition chain, the $p_{k,r}[n]$ must be computed. As found in the above, it suffices to compute one $p_{k,r}[n]$ within each layer, say $p_{k,1}[n]$, from which the state transition chain for $s_{k,1}[n]$ can be determined. A recursive algorithm to compute $p_{k,1}[n]$—derived from Figure 3.12 and (42)—is as follows:

```
function $p_k$( integer j, integer $x[n]$)
if j = k then
    return LSB($x[n]$)
else
    if $u_{j,1}[n] = 0$ then
        return $\frac{1}{2}\Big[p_k(j-1, (x[n]+\mathrm{LSB}(x[n]))/2)+p_k(j-1, (x[n]-\mathrm{LSB}(x[n]))/2)\Big]$
    else
        return $p_k(j-1, (x[n]-(u_{j,1}[n] \times \mathrm{LSB}(x[n])))/2)$
```

Thus, $p_{k,1}[n]$ is found by computing $p_k(b, x_D[n])$.

An algorithm to compute $\overline{R}_k[m]$ can now be stated as follows:

1. Let $R_k[m]$ denote an initially empty list.
2. Starting in state "0" at time 0, traverse the recurrent state transition paths until they all arrive at state "0" at time t_1. Label each state transition with the corresponding transition probability $p_{k,1}[n]$.
3. For $0 \leq n < t_1$, compute all possible values of

   $$A[m] = (i)(j)(P\{s_{k,r}[n] = i\})(P\{s_{k,r}[n+m] = j\}), \quad i = \pm 1, j = \pm 1,$$

   for all m such that $0 \leq n + m < t_1$, and where $P\{s_{k,r}[n] = i\}$ is the probability that $s_{k,r}[n] = i$, given by the product of the transition probabilities along the path leading to the state under consideration.
4. Add the samples of $A[m]$ to the samples of $R_k[m]$, i.e., $R_k[m] = R_k[m] + A[m]$.
5. Repeat steps 2 through 4 with starting time t_i and ending time t_{i+1} for a number of iterations, K, large enough to employ the law of large numbers to ensure that

   $$\frac{1}{t_K} R_k[m] \tag{44}$$

equals $\overline{R}_k[m]$ with the desired accuracy. In the limiting case, i.e., for $K \to \infty$, (44) equals $\overline{R}_k[m]$ with probability 1.

REFERENCES

1. *Oversampling Delta-Sigma Data Converters Theory, Design and Simulation,* Edited by J. C. Candy and G. C. Temes, New York, IEEE Press, 1992.
2. R. van de Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters.* Boston, MA: Kluwer Academic Publishers, 1994.
3. R. T. Baird, T. S. Fiez, "Linearity enhancement of multi-bit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 42, no. 12, pp. 753-762, Dec. 1995.
4. R. Schreier, B. Zhang, "Noise-shaped multi-bit D/A converter employing unit elements," *Electronics Letters*, vol. 31, no. 20, pp. 1712-1713, Sept. 28, 1995.
5. I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing,* accepted for publication.
6. J. C. Candy, "A use of double integration is sigma delta modulation," *IEEE Trans. on Communications*, vol. com-33, no. 3, pp. 249-258, March 1985.
7. S. R. Norsworthy, R. Schreier, G. C. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation.* Piscataway, NJ: IEEE Press, 1997.
8. S. W. Golomb, *Shift Register Sequences.* San Francisco, CA: Holden-Day, 1967.
9. A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1989.
10. N. S. Jayant, P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

11. I. Galton, P. Carbone, "A rigorous error analysis of D/A conversion with dynamic element matching," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 763-772, Dec. 1995.

12. B. H. Leung, S. Sutarja, "Multi-bit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 1, pp. 35-51, Jan. 1992.

13. R. W. Adams, T. W. Kwan, "Data-directed Scrambler for Multi-bit Noise Shaping D/A Converters", U.S. Patent No. 5,404,142, Apr. 4, 1995.

14. T. W. Kwan, R. W. Adams, R. Libert, "A stereo multi-bit $\Sigma\Delta$ D/A with asynchronous master-clock interface", *IEEE ISSCC Dig. of Tech. Papers*, vol. 39, pp. 226-227, Feb. 1996.

15. R. K. Henderson, O. J. A. P. Nys, "Dynamic element matching techniques with arbitrary noise shaping functions," *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1996.

16. F. Chen, B. H. Leung, "A high resolution multibit sigma-delta modulator with individual level averaging," *IEEE J. Solid-State Circuits*, vol. SC-30, no. 4, pp. 453-460, April 1995.

17. L. R. Carley, "A noise shaping coder topology for 15+ bits converters," *IEEE J. Solid-State Circuits*, vol. SC-24, pp. 267-273, April 1989.

18. L. R. Carley, J. Kenney, "A 16-bit 4'th order noise-shaping D/A converter," *IEEE Proc. CICC*, pp. 21.7.1-21.7.4, 1988.

19. K. R. Laker, W. M. C. Sansen, *Design of Analog Integrated Circuits and Systems,* New York: McGraw-Hill, 1994.

20. M. Ismail, T. Fiez, *Analog VLSI,* New York: McGraw-Hill, 1994.