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

Henrik T. Jensen, *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 this technique over the prior art is a significantly reduced hardware complexity with no reduction in performance. It is particularly appropriate for applications such as direct digital synthesis (DDS) in wireless communications systems, where low hardware complexity and low harmonic distortion are essential.**

***Index Terms*— DAC, data-converter, digital-to-analog, direct digital synthesis, dynamic element matching, harmonic distortion, mismatch shaping, noise-shaping.**

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, nonideal circuit behavior causes the DAC to introduce *DAC noise*. At least a component of the DAC noise is a nonlinear function of the input sequence, so harmonic distortion 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 that the harmonic distortion introduced by the DAC is low. For example, an extremely low-complexity digital portion of an 8-bit DDS system has demonstrated that it is capable of achieving a minimum SFDR of 90 dB, provided that the minimum SFDR of the DAC is 90 dB or greater.

Thus, a remaining problem is to develop moderate-resolution DAC's 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 DAC's, the outputs of which together yield a single multibit DAC [2]–[4]. For most digital input values, there are many possible input codes to the bank of 1-bit DAC's that nominally yield the desired analog output value. Thus, the DAC noise arising from errors introduced by the 1-bit DAC's can be "scrambled" by randomly selecting one of the appropriate codes for each digital input value. Although DAC's based on this approach have been shown experimentally [2], [3] and through quantitative analysis [4] to achieve excellent SFDR's, the presented DAC's 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 DAC's 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 tradeoff 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 DAC's 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 DAC's presented in [2]–[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 Appendixes.

Manuscript received April 8, 1996; revised March 26, 1997. This paper was recommended by Associate Editor B. Lee.

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

Publisher Item Identifier S 1057-7130(98)00065-2.

1057–7130/98$10.00 © 1998 IEEE

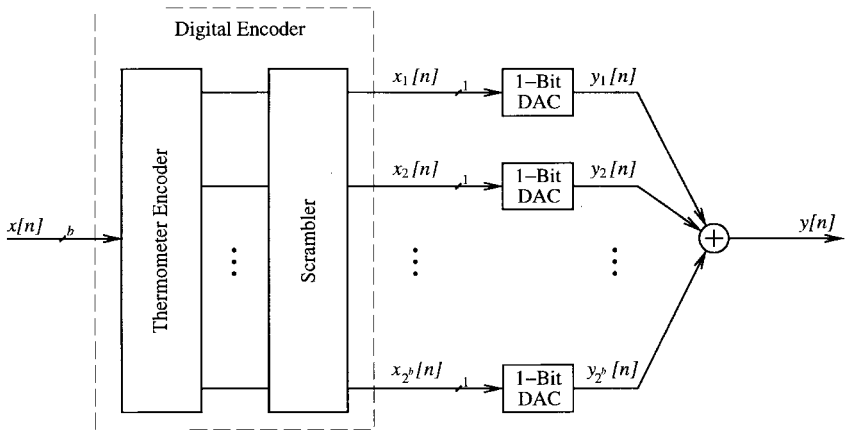

Fig. 1. The high-level topology of the low-harmonic distortion DAC's presented in [2] and [4].

II. Low-Harmonic Distortion DAC Approaches

A. Background and Prior Art

The high-level topology shared by the DAC's presented in [2] and [4] is shown in Fig. 1. The digital input, $x[n]$, is a sequence of unsigned b-bit binary numbers less than 2^b, i.e,

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

The DAC consists of a *digital encoder*, 2^b 1-bit DAC's referred to as *unit DAC-elements*, and an analog output summing node. At the high level of Fig. 1, the digital encoder maps each input sample to 2^b output bits, $x_1[n] \cdots x_{2^b}[n]$, 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 rth unit DAC-element, and e_{h_r} and e_{l_r} are errors in the analog output levels arising from inevitable nonidealities 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 rth 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)–(3) that $y[n] = x[n]$ in the absence of static DAC-element errors. However, as shown in [5], with nonzero 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 $\times$ 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, 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 Fig. 2. To simplify the figure, a 3-bit example is shown. The DAC is of the general 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.

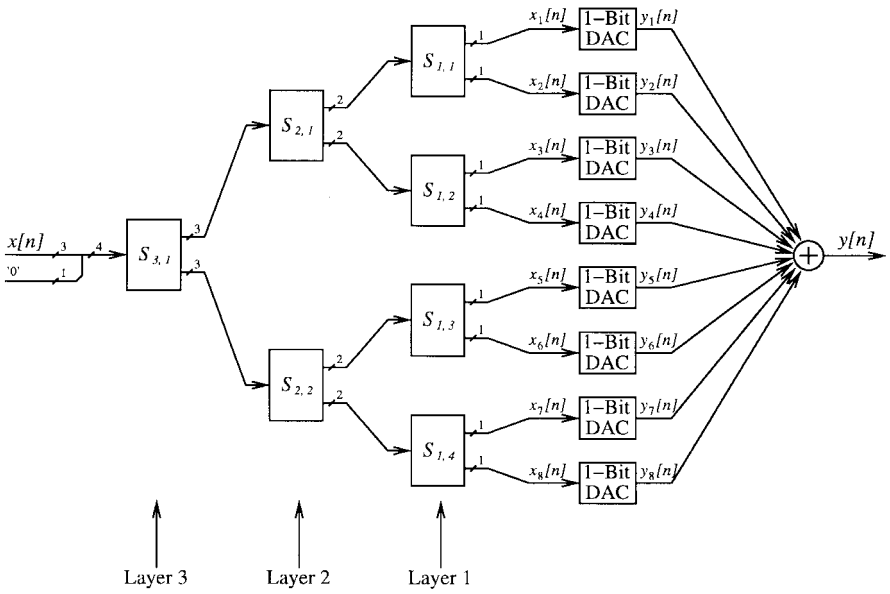

Fig. 2. A 3-bit version of the proposed DAC architecture.

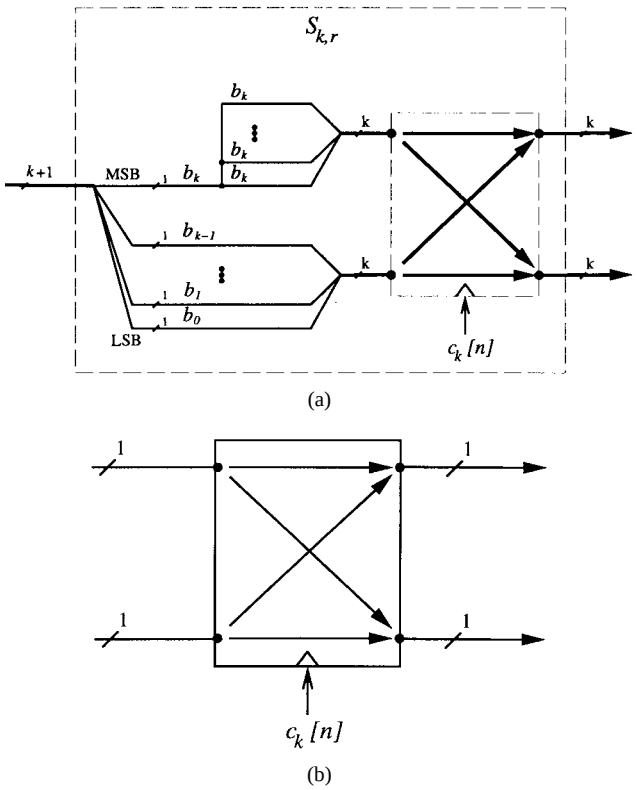

Fig. 3. Details of (a) the switching block $S_{k,r}$, and (b) the binary switch.

Fig. 3(a) 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 kth layer (for clarity, the random control bits are not shown in Fig. 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 $c_k[n]$. Fig. 3(b) 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–b_b, and a zero is assigned to the input bit b_0, as indicated in Fig. 2. It is shown in Appendix A that the digital encoder obtained by interconnecting the switching blocks of Fig. 3(a) as shown in Fig. 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. *full randomization* DEM refers to a DAC with random switching in *all* layers, i.e., layers 1 through b. *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 Fig. 4, where random switching is performed in layers 6–8. Layers 1–5 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 Fig. 5. The LSB of the input controls a unit DAC-element, whereas the remaining five 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 Fig. 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

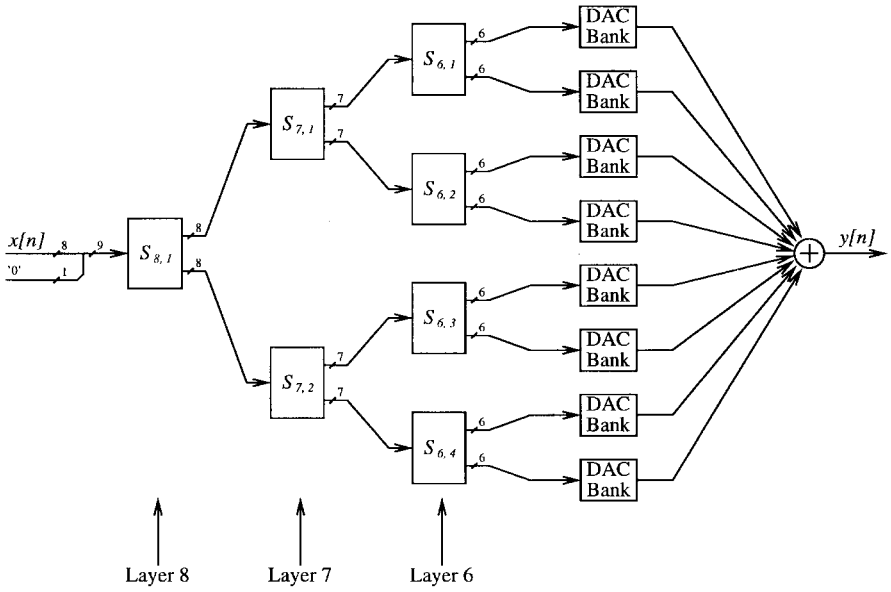

Fig. 4. An 8-bit DAC with partial randomization DEM.

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].

Fig. 6(a) corresponds to $y[n]$ of an ideal DAC (i.e., a DAC with no static DAC-element errors, so $y[n] = x[n]$), Fig. 6(b) corresponds to $y[n]$ with no random switching (the digital encoder thus being equivalent to a thermometer-encoder), Fig. 6(c) corresponds to $y[n]$ with full randomization DEM, and Fig. 6(d) corresponds to the signal $y[n] - x[n]$ with full randomization DEM. The static DAC-element 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 of the proposed DEM technique [2], [7].

As is evident from the numerous spurs distributed across the spectrum in Fig. 6(b), 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 rad, 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 Fig. 6(c) 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 Section I.

Additional details of the simulation results are as follows. The PSD's 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.

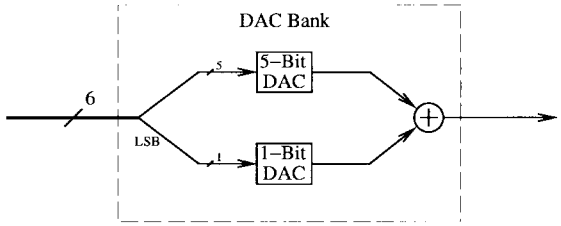

Fig. 5. Details of the DAC bank.

B. Performance Equations

A detailed theoretical performance analysis of full randomization DEM is given in Appendix B and Appendix 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 ith 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

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

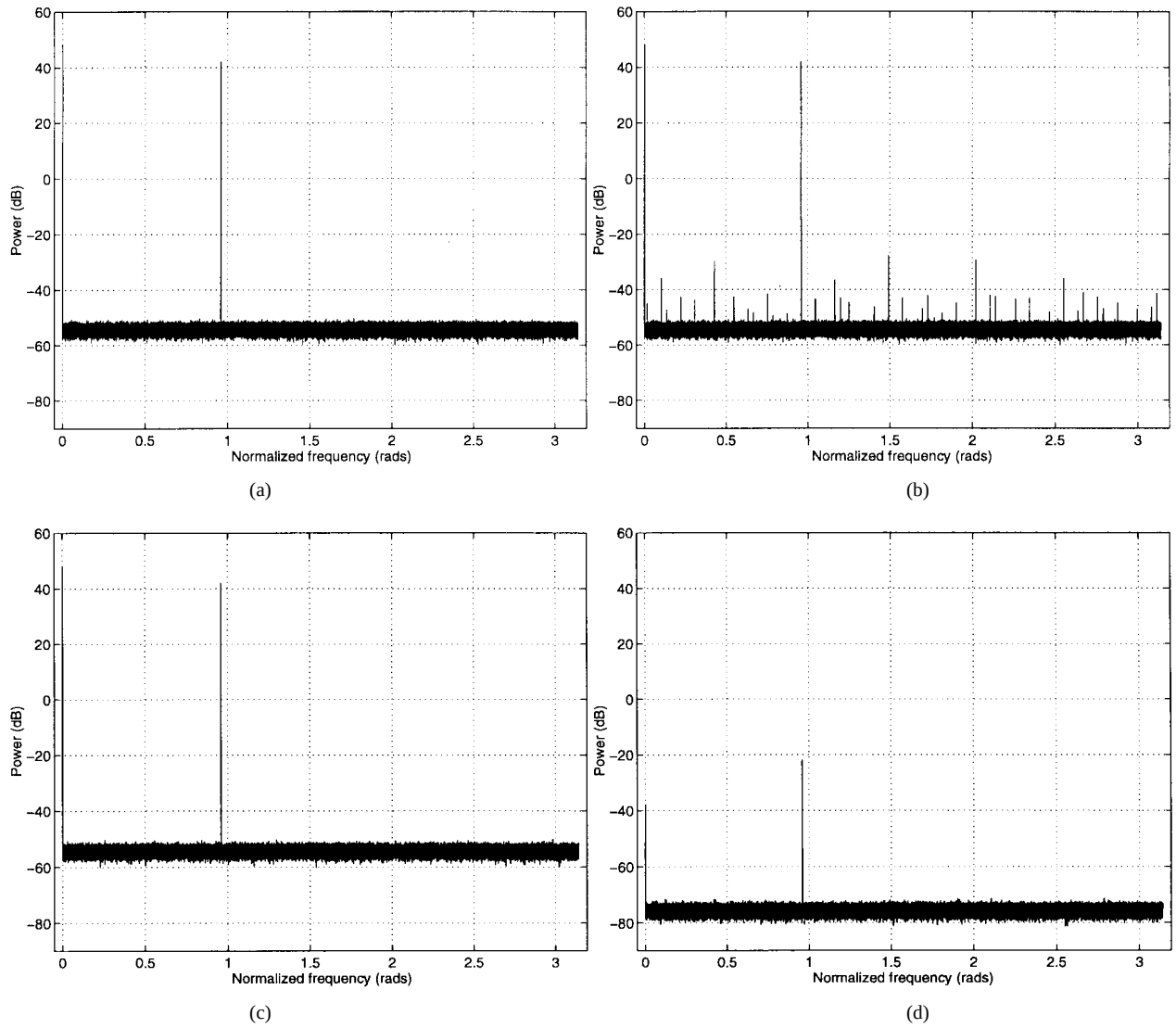

Fig. 6. Simulated PSD's relative to $x_{\max}^2$ of an example 8-bit DAC of (a) the ideal DAC output $y[n]$. (b) the DAC output $y[n]$ with no random switching, (c) the output $y[n]$ with full randomization DEM, and (d) the signal $y[n] - x[n]$ with full randomization DEM.

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 as

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

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 A provides exact formulas for the constants α and β in (5), and expressions for $w^{(i)}[n]$ in (6) are developed in Appendix 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 B provides formulas 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 Fig. 6.

It follows from (5) that a nonunity 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

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

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 formulas for α and β given in Appendix A, and resulted in $\alpha-1=-6.4361\times10^{-4}$ and $\beta=7.5634\times10^{-2}$, 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 Fig. 6(a) and (d) yields -63.82 and -85.96 dB, respectively, in agreement with the 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 Fig. 6(d) results in -75.44 dB, in agreement with the theory. As an additional comment pertaining to the details of Fig. 6, it is evident from a comparison of Fig. 6(c) and (d) 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 $\overline{\sigma}^2$, Fig. 7(a)–(d) shows plots of $\overline{\sigma}^2$ in decibels relative to $x_{\max}^2$ for four nominally identical 8-bit versions of the proposed DAC. The DAC's 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.

Fig. 8(a)–(d) shows plots of $\overline{\sigma}^2$ in dB relative to $x_{\max}^2$ for the same four example 8-bit DAC's as in Fig. 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 $x_1[n]=A_1\sin(\omega_1 n)$ 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, -77.84, -73.02, and -74.33 dB, respectively. Thus, the random variation of the static DAC-element errors of the example DAC's causes a spread in the average value of $\overline{\sigma}^2$ of 4.82 dB.

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 DAC's that will result in a value of $\overline{\sigma}^2$ less than any value of interest. The IC-fabrication 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, Fig. 9(a)–(d) shows IC-fabrication yield estimation data corresponding to 4-, 6-, 8-, and 10-bit DAC's, 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 decibels 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 Fig. 9(c) predicts that 95% of all 8-bit DAC's will satisfy $\overline{\sigma}^2 \le -72$ dB relative to $x_{\max}^2$ and that 5% will satisfy

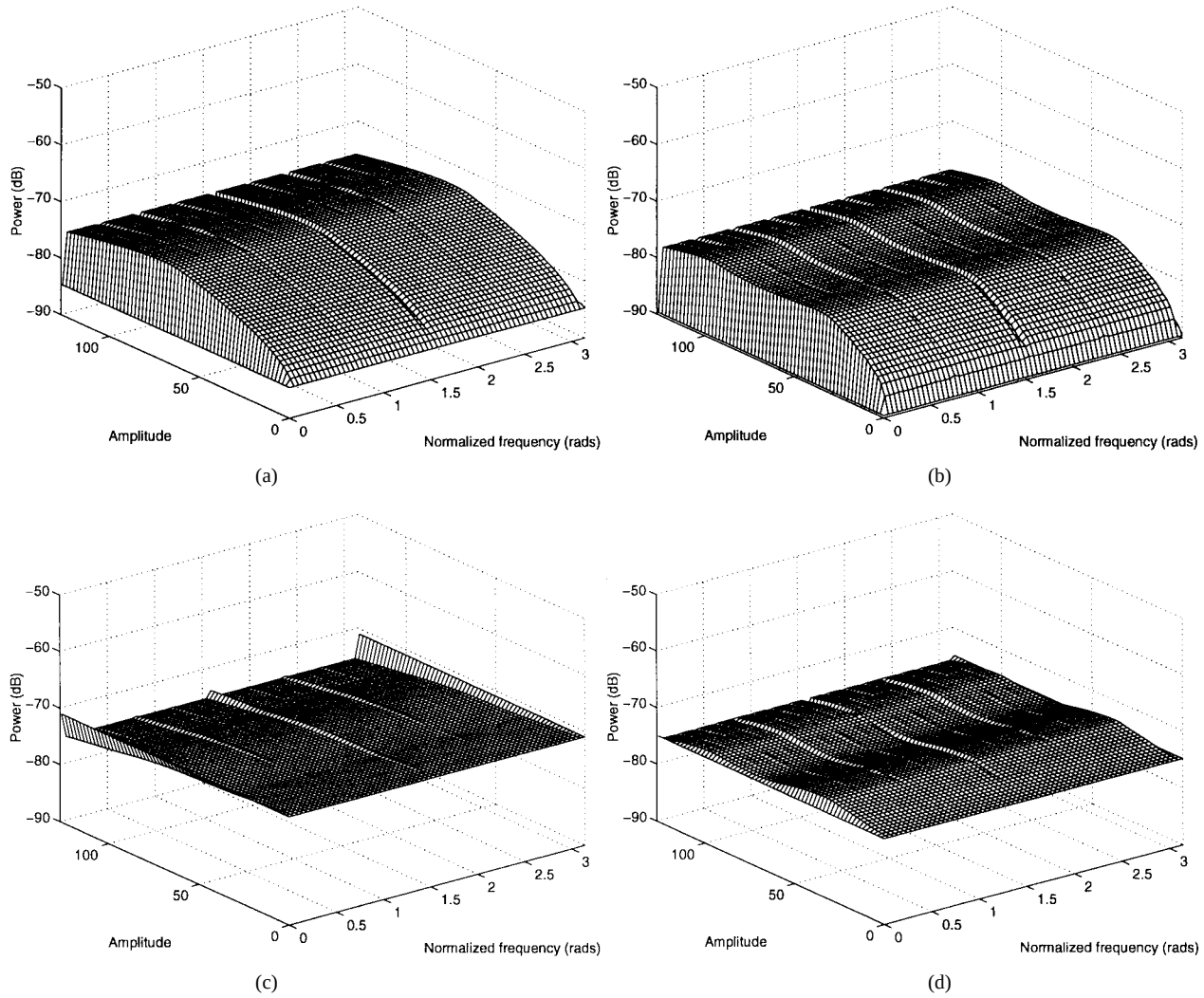

Fig. 7. Plots of $\overline{\sigma}^2$ in decibels relative to $x_{\max}^2$ for four nominally identical 8-bit versions of the proposed DAC. The DAC's in (a)–(d) differed only in the static DAC-element errors, which were randomly chosen from a normal distribution with a standard deviation of 0.5%. In 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.

$\overline{\sigma}^2 \leq -81$ dB relative to $x_{\max}^2$. Thus, 90% of all 8-bit DAC's 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 Fig. 6, for which $\overline{\sigma}^2 = -75.44$ dB relative to $x_{\max}^2$, and by the four example 8-bit DAC's of Fig. 7, for which $\overline{\sigma}^2$ equals $-75.53, -78.35, -73.91$, 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 DAC's driven by a particular sinusoid. However, as was demonstrated in Fig. 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 Fig. 9. As was also demonstrated with the data in Fig. 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 Fig. 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 static DAC-element errors. It follows that Fig. 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 nonideal

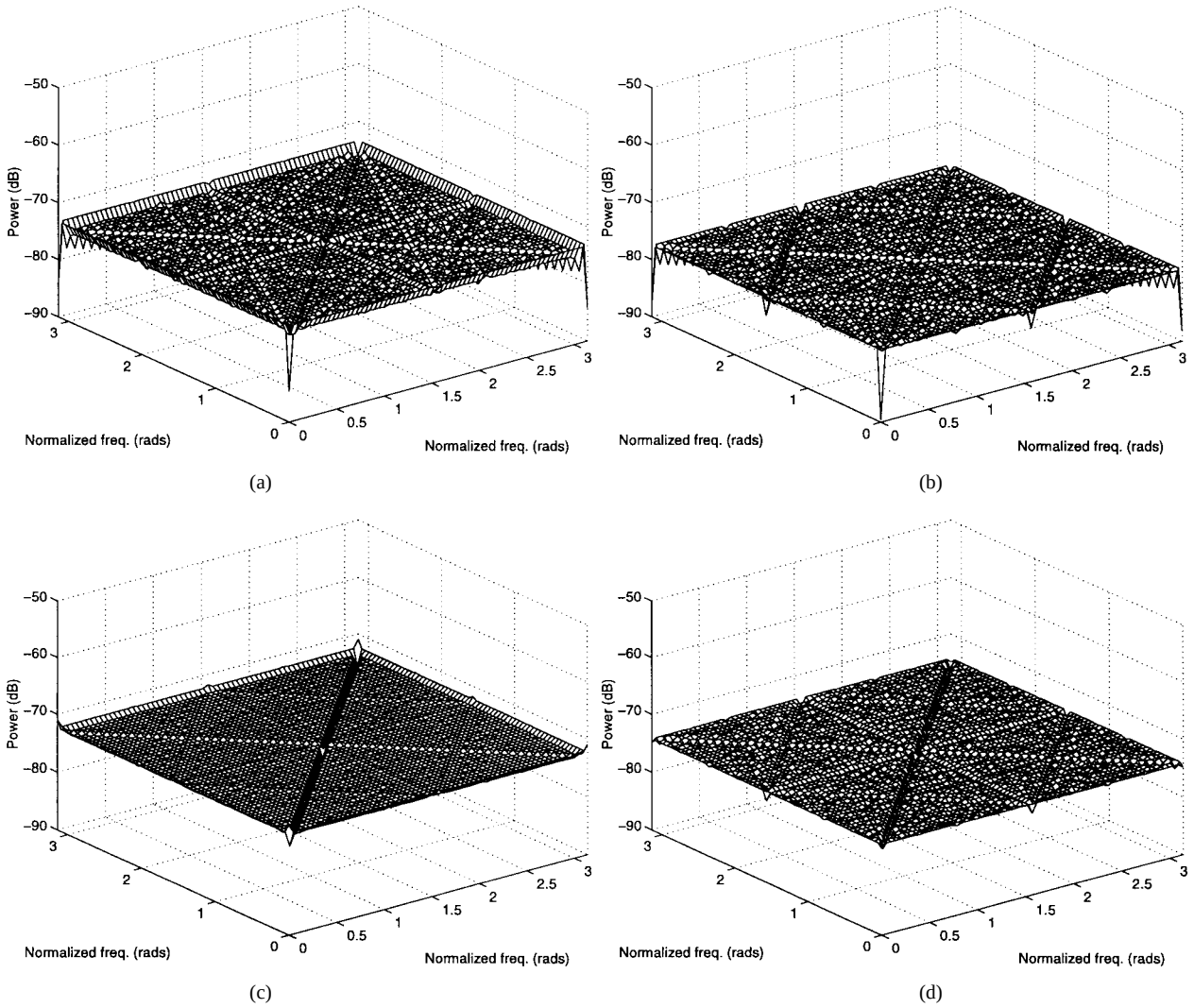

Fig. 8. Plots of $\overline{\sigma}^2$ in decibels relative to $x_{\max}^2$ for the same four example 8-bit DAC's as in Fig. 7 when driven by the sum of two sinusoids. In each plot, the amplitude of each sinusoidal component was fixed, and $\overline{\sigma}^2$ computed using (10) is shown as a function of the frequencies of the sinusoidal components.

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 are shown in Fig. 10. In particular, Fig. 10(a)–(c) correspond to the signal $y[n] - x[n]$ with random switching in layer 8, layers 7 and 8, and layers 6–8, respectively. Fig. 10(d) corresponds to $y[n]$ with random switching in layers 6–8. In all cases, the DAC input and static DAC-element errors were identical to those used for the full randomization DEM example of Fig. 6. The simulation results indicate that the harmonic distortion is gradually attenuated as the number of layers with random switching is increased. The maximum-amplitude spurs of Fig. 10(a)–(c) have powers -73.55, -89.47, 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 SFDR's provided by the DAC are 73.55, 89.47, and 97.49 dB, respectively. The parameters used to compute the PSD's were identical to the parameters used to compute the PSD's of Fig. 6.

To summarize the simulation results of Fig. 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 Section I. 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

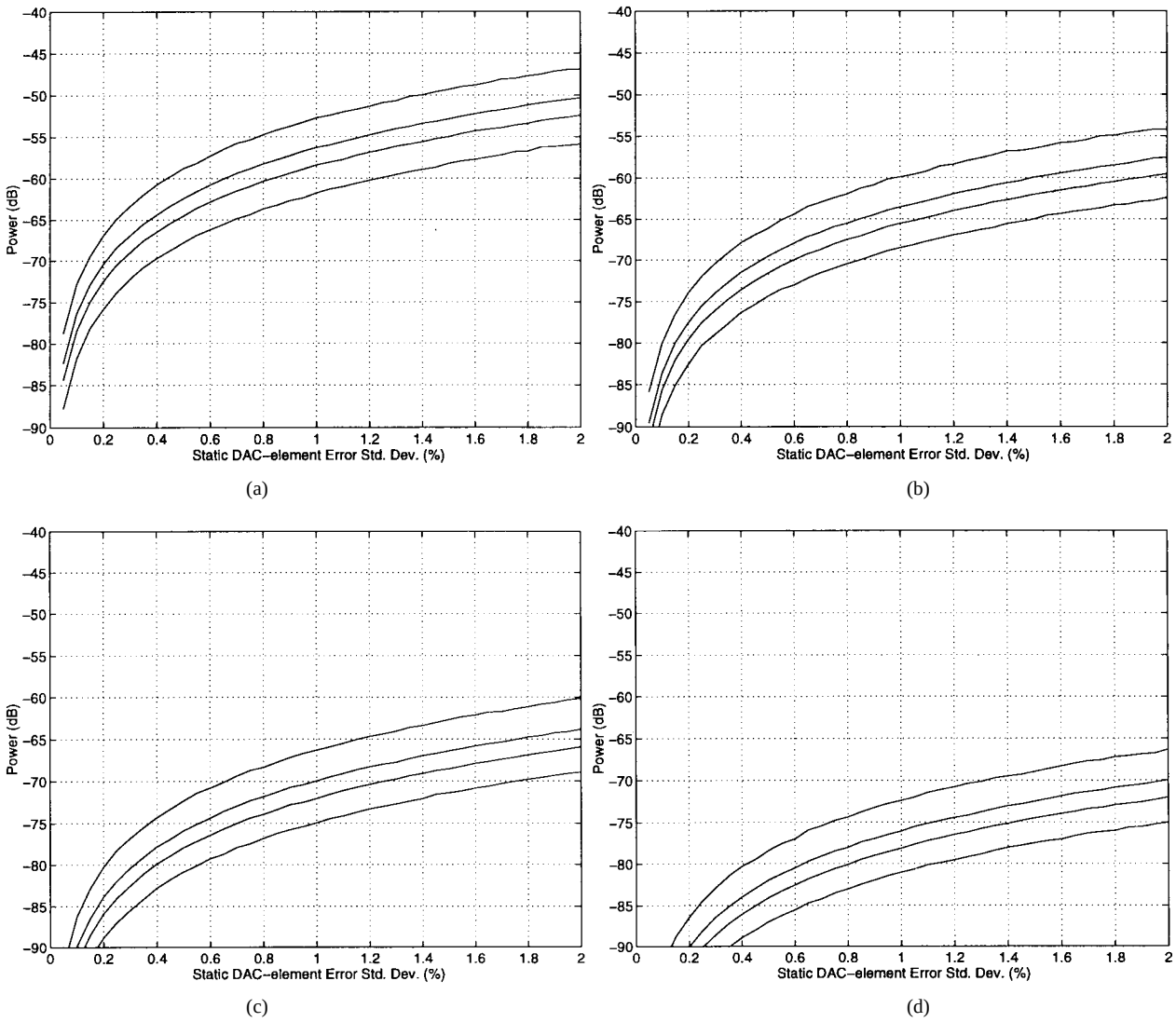

Fig. 9. IC-fabrication yield estimation data for (a) 4-, (b) 6-, (c) 8-, and (d) 10-bit versions of the proposed DAC. The static DAC-element errors were chosen from a normal distribution with a standard deviation ranging from 0.05 to 2%. In each case, from top to bottom, the curves show the largest of the smallest 95, 65, 35, and 5% of $\overline{\sigma}^2$ values in decibels relative to $x_{\max}^2$, respectively, when driving the DAC by a sinusoidal input.

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.

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 number of required binary switches of the digital encoders presented in [2] and [4] is $b2^{b-1}$ 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 DAC's is shown in Fig. 11. It shows the hardware complexity of the digital encoder presented in [2] and the proposed architecture for bit-resolutions 6–12. 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.

B. Partial Randomization DEM

As discussed previously, very low hardware complexity is achievable with partial randomization DEM. To obtain a

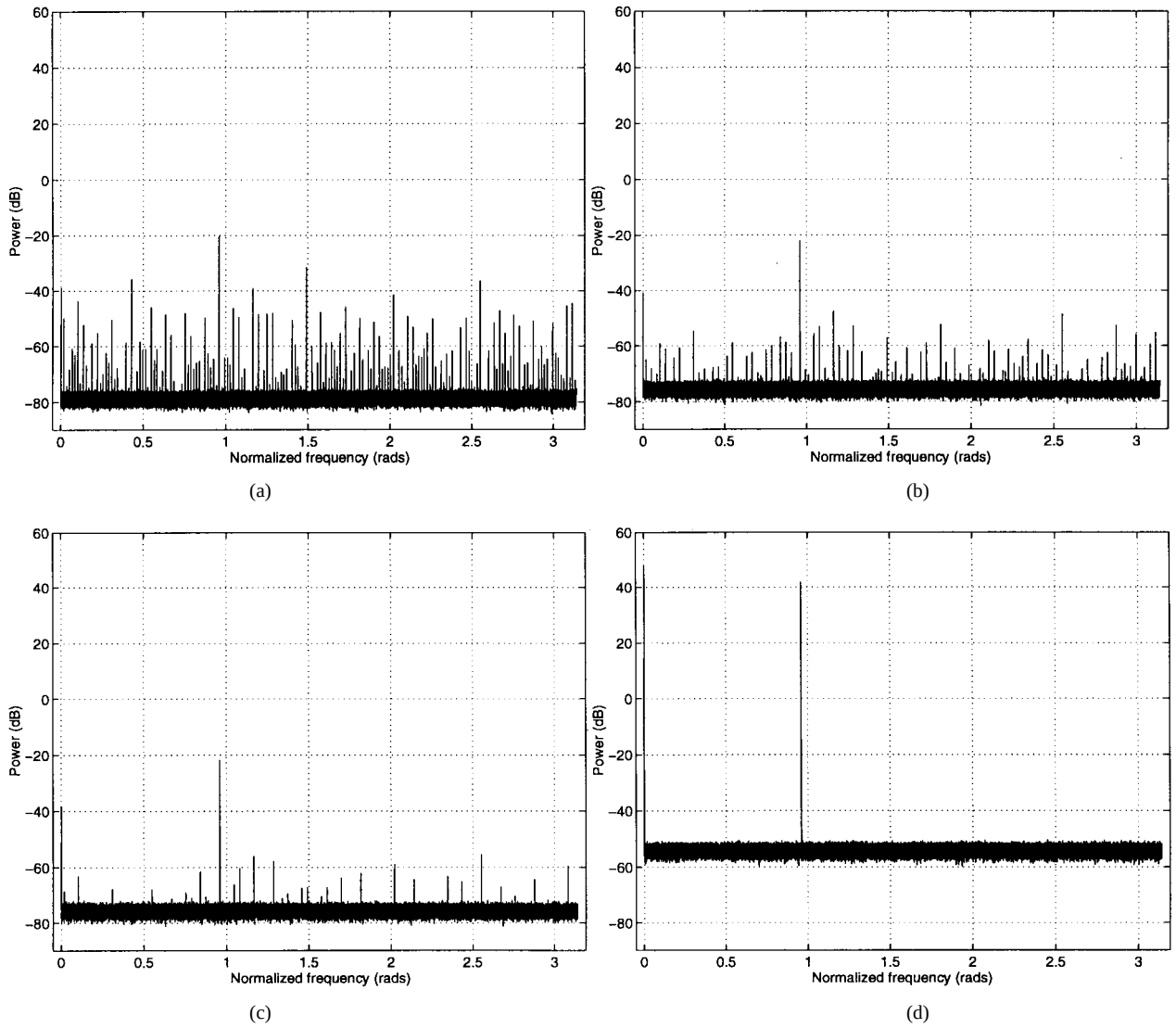

Fig. 10. Simulated PSD's relative to $x_{\max}^2$ of an example 8-bit DAC with partial randomization DEM of (a) $y[n]-x[n]$ with random switching in layer 8, (b) $y[n]-x[n]$ with random switching in layers 7 and 8, (c) $y[n]-x[n]$ with random switching in layers 6–8, and (d) $y[n]$ with random switching in layers 6–8.

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

Fig. 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.

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 6–8 requires $7\times 8-8-2=46$ binary switches and merely three random control bits. This should be compared to the requirement of 502 binary switches and eight 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 Fig. 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 very large

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

Fig. 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.

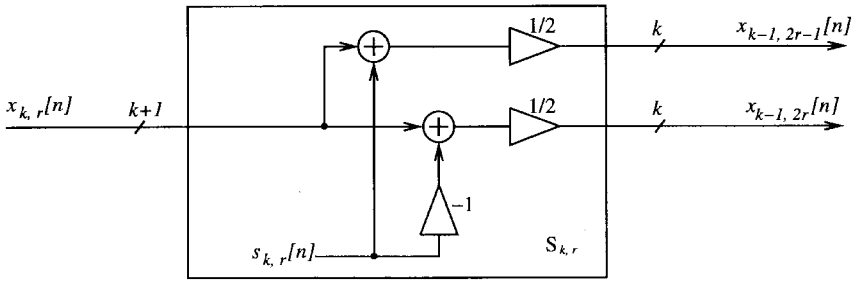

Fig. 13. The signal processing equivalent of the switching block $S_{k,r}$.

scale integration (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 considerable additional reduction in hardware complexity. It has been shown that for an 8-bit DAC with partial randomization DEM, merely three 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.

Nonideal circuit behavior such as clock-skew, clock-coupling, and finite slew-rates inevitably contributes to harmonic distortion of the DAC output. Such nonideal circuit behavior is typically quite implementation dependent, and research to quantify and mitigate its effects must be performed on a case-by-case basis. Nevertheless, the results presented in this paper are still applicable to 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.

Appendix 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 B, the general form of the DAC noise $e[n]$ given here is rewritten 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 DAC's considered in [5] have switching blocks that perform the signal processing operations depicted in Fig. 13, and, as is shown below, the switching blocks of the DAC architecture proposed in this paper can also be viewed as shown in Fig. 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 ith 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)

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

and $s_{k,r}[n]$ is defined by (13). These results do not depend upon any particular form or statistical property 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

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

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

and

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

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

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

Inserting this in (22) yields (19).

Next, it will be shown that the switching blocks presented in this paper perform signal processing according to Fig. 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 kth layer. Then, (23) and (24) will be used to verify (20) and (21).

Suppose $c_k[n] = 0$. It follows from Fig. 3, (13), and (22) that

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

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)

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

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

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

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 $c_k[n] = 1$, $k = 1, 2, \cdots, R-1$. Thus, it follows that Claim A also holds for partial randomization DEM. ■

Appendix 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

$$w^{(i)}[n] = w_{h_i[n]}^{(i)}. \tag{27}$$

In (27),

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

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 B, 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

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

which, combined with (18), yields

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

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

$$e[n]=\Delta_{1,1}s_{1,1}[n] \quad (31)$$

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

$$e[n]=w^{(1)}[n]x^{(1)}[n] \quad (32)$$

where

$$w^{(1)}[n]=w^{(1)}_{h_1[n]} \quad \text{and} \quad h_1[n]=(1-c_1[n])+1$$

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 $w_2^{(1)}$, 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 $1\le b\le b'$. It will be shown that the claim holds for $b'+1$. Notice that (17) may be written as

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

where

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

and

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

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 Fig. 3 and (22) that $x_{b',1}[n]$ is either 0 or $2^{b'}$. Consequently, all the $x_{k,r}[n], 1\le k\le b', 1\le r\le 2^{b'-k}$ 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

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

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 $x^{(i)}[n], 1\le i\le b'+1$ in (36) with the coefficients determined from (26) with $b=b'+1$. The coefficient of $x^{(b'+1)}[n]$ in (36) is

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

which equals $w^{(b'+1)}[n]$, as asserted. Next, suppose $c_{b'+1}[n]=0$. The coefficient of $x^{(i)}[n], 1\le i\le b'$ 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

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

and

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

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

$$\begin{aligned}\mathrm{Var}\{e[n]\} = &\sum_{i=1}^{b} \mathrm{E}\{(w^{(i)}[n])^2\} x^{(i)}[n] \\ &+ 2 \sum_{j=1}^{b-1} \sum_{i=j+1}^{b} \mathrm{E}\{w^{(i)}[n] w^{(j)}[n]\} \\ &\cdot x^{(i)}[n] x^{(j)}[n] \end{aligned} \tag{40}$$

where use was made of the equality $\{x^{(i)}[n]\}^2 = x^{(i)}[n]$. To evaluate (40), consider first $\mathrm{E}\{(w^{(i)}[n])^2\}$. 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 $c_k[n], k = i, i+1, \cdots, b$. The two possible values of $c_k[n]$ are equiprobable 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 equiprobable. It then follows from (27) that the 2^{b-i+1} different values of $w^{(i)}[n]$ are equiprobable, i.e., $w^{(i)}[n] = w_j^{(i)}$ with probability $1/2^{b-i+1}$. Thus

$$\mathrm{E}\{(w^{(i)}[n])^2\} = \frac{1}{2^{b-i+1}} \sum_{j=1}^{2^{b-i+1}} [w_j^{(i)}]^2 \tag{41}$$

as asserted.

It will next be shown that

$$\mathrm{E}\{w^{(i)}[n] w^{(j)}[n]\} = \frac{1}{2^{b-j}} \sum_{k=1}^{2^{b-i}} w_{2k-1}^{(i)} w_{2k}^{(i)}. \tag{42}$$

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

$$\begin{aligned} w^{(i)}[n] = w_{2k-1+(1-r)}^{(i)}, \qquad & k \in \{1, 2, \cdots, 2^{b-i}\}, \\ & r \in \{0, 1\} \end{aligned} \tag{43}$$

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

$$w^{(j)}[n] \in \{w_{2^{i-j}(2k-2+r)+m}^{(j)} : m = 1, 2, \cdots, 2^{i-j}\} \tag{44}$$

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

$$\begin{aligned}&\mathrm{E}\{w^{(i)}[n] w^{(j)}[n]\} \\ &\quad = \frac{1}{2^{b-i+1}} \frac{1}{2^{i-j}} \sum_{r=0}^{1} \\ &\qquad \cdot \left\{ \sum_{k=1}^{2^{b-i}} w_{2k-1+(1-r)}^{(i)} \left[\sum_{m=1}^{2^{i-j}} w_{2^{i-j}(2k-2+r)+m}^{(j)} \right] \right\}. \end{aligned} \tag{45}$$

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

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

From (28) it follows that (46) can be verified by establishing the appropriate limits for the summation of the terms $(e_{l_m} - e_{l_m - \alpha + 1})$. 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

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

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

APPENDIX 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

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

and consequently

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

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 DAC-element errors are bounded.

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

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

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

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

where

$$\sigma^2[n] = \mathrm{E}\{e^2[n]\} = \mathrm{Var}\{e[n]\},$$

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 and G. A. Zimmerman, "Spur-reduced digital sinusoid synthesis," *IEEE Trans. Commun.*, vol. 43, pp. 2254–2262, July 1995.

[2] L. R. Carley, "A noise shaping coder topology for 15+ bits converters," *IEEE J. Solid-State Circuits*, vol. 24, pp. 267–273, Apr. 1989.

[3] L. R. Carley and J. Kenney, "A 16-bit 4th order noise-shaping D/A converter," *IEEE Proc. CICC*, pp. 21.7.1–27.7.4, 1988.

[4] I. Galton and P. Carbone, "A rigorous error analysis of D/A conversion with dynamic element matching," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 763–772, Dec. 1995.

[5] I. Galton, "Spectral shaping of circuit errors in D/A converters," *IEEE Trans. Circuits Syst. II*, to be published.

[6] R. M. Gray and T. G. Stockham, Jr., "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 805–812, May 1993.

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

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

Henrik T. Jensen (M'97) was born in Odense, Denmark, in 1968. He received the B.S. degree in computer engineering from Odense University, Odense, Denmark, in 1992, the M.S. degree in electrical and computer engineering from the University of California, Irvine, in 1994, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego, in 1997.

He currently holds a position as Postgraduate Research Engineer at the University of California, San Diego. His research interests include digital signal processing and data conversion techniques.

Ian Galton (M'92) received the Sc.B. degree in electrical engineering from Brown University in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology in 1989 and 1992, respectively.

Prior to 1988, he worked at Acuson, Inc., Mountain View, CA, on acoustic-beam-formation software for use with a medical ultrasound imaging system and as a software engineer for Mead Data Central, Menlo Park, CA. From 1992 through 1996, he worked as an Assistant Professor of Electrical Engineering at the University of California, Irvine. In 1996, he joined the faculty of the University of California, San Diego, as an Associate Professor of Electrical Engineering. His research interests involve integrated signal processing circuits and systems for communications and include the analysis and implementation of data converters and digital phase locked loops. He holds four patents.

Dr. Galton received the Caltech Charles Wilts doctoral thesis prize.