Spectral Shaping of Circuit Errors in Digital-to-Analog Converters

Ian Galton, *Member, IEEE*

***Abstract*—Recently, various multibit noise-shaping digital-to-analog converters (DAC's) have been proposed that use digital signal processing techniques to cause the DAC noise arising from analog component mismatches to be spectrally shaped. Such DAC's have the potential to significantly increase the present precision limits of $\Delta\Sigma$ data converters by eliminating the need for one-bit quantization in delta-sigma modulators. This paper extends the practicality of the noise-shaping DAC approach by presenting a general noise-shaping DAC architecture along with two special-case configurations that achieve first- and second-order noise-shaping, respectively. The second-order DAC configuration, in particular, is the least complex of those currently known to the author. Additionally, the paper provides a rigorous explanation of the apparent paradox of how the DAC noise can be spectrally shaped even though the sources of the DAC noise—the errors introduced by the analog circuitry—are not known to the noise-shaping algorithm.**

***Index Terms*—ADC, analog-to-digital, DAC, data converter, delta–sigma, digital-to-analog, mismatch shaping, noise-shaping, sigma–delta.**

I. Introduction

DELTA–SIGMA analog-to-digital converters ($\Delta\Sigma$ADC's) and digital-to-analog converters ($\Delta\Sigma$DAC's), collectively referred to as $\Delta\Sigma$ data converters, are widely used in high-precision, low bandwidth applications such as digital audio processing. Through the use of oversampling, coarse internal quantization, and quantization noise-shaping they are able to perform high-precision data conversion yet can be implemented in VLSI technology optimized mainly for digital circuitry [1]. The coarse quantization simplifies the analog processing, but most of the resulting error is kept out of the signal band as a result of the quantization noise-shaping and oversampling. The primary impediment to their use in higher-rate applications such as digital radio and video processing is the oversampling requirement. Oversampling necessitates the use of circuitry with a significantly higher bandwidth than the signal to be converted, and in higher-rate applications this is often not practical.

In principle, the oversampling requirement can be reduced by increasing the order of the quantization noise-shaping. However, practical issues such as analog component matching and system stability have so far limited the extent to which the noise-shaping order can be usefully increased [2]–[7]. For example, in theory it is possible to design a $\Delta\Sigma$ data converter with multibit quantization that achieves any given order of quantization noise-shaping. Unfortunately, the application of such $\Delta\Sigma$ data converters has been limited to date because of their high sensitivity to analog component mismatches in their internal DAC's. Typically, the internal DAC in a $\Delta\Sigma$ data converter operates on a highly quantized digital sequence, so it does not require a large number of output levels. However, it must be nearly linear and introduce very little signal-band circuit noise if high-precision data conversion is required of the overall $\Delta\Sigma$ data converter. In principle, this can be achieved if the *steps* between adjacent output levels of each DAC have very nearly the same magnitude. Unfortunately, the required matching precision is on the order of the desired precision of the overall data converter and this is often beyond the practical limits of present VLSI technology. For this reason, the majority of $\Delta\Sigma$ data converters use two-level (i.e., one-bit) DAC's; in this case, there is only one step so matching the sizes of multiple steps is not an issue. While one-bit DAC's effectively avoid the step matching problem, they necessitate one-bit quantization in $\Delta\Sigma$ data converters and this forces design tradeoffs to be made that, for a given level of oversampling, significantly reduce data conversion signal-to-noise ratios (SNR's) below what would be possible if multibit quantization could be used [2]–[4], [8].

With this in mind, a number of investigators have recently proposed various multibit noise-shaping DAC's for use in $\Delta\Sigma$ data converters [9]–[17]. The noise-shaping DAC's all use digital signal processing techniques to cause the error resulting from step-size mismatches to lie outside of the $\Delta\Sigma$ data converter signal band. Thus, instead of trying simply to minimize mismatches through improvements in the design or layout of the analog circuits, the mismatches are accepted as inevitable, but their negative effects are largely avoided though a signal processing innovation. Consequently, the noise-shaping DAC's allow for multibit quantization within $\Delta\Sigma$ data converters but do not suffer from the extreme matching problems associated with conventional multibit DAC's.

This paper presents a general noise-shaping DAC architecture along with two special-case configurations that achieve first-order and second-order noise-shaping, respectively. The first-order DAC configuration has similar complexity to that presented in [11], but offers the advantage that it can be used in conjunction with a simple dithering technique that suppresses spurious tones. The second-order DAC is by far

Manuscript received December 27, 1995; revised April 4, 1997. This work was supported by the California MICRO program in conjunction with the Rockwell International Corporation Science Center. This paper was recommended by Associate Editor F. Larsen.

The author is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: galton@ece.ucsd.edu).

Publisher Item Identifier S 1057-7130(97)07643-X.

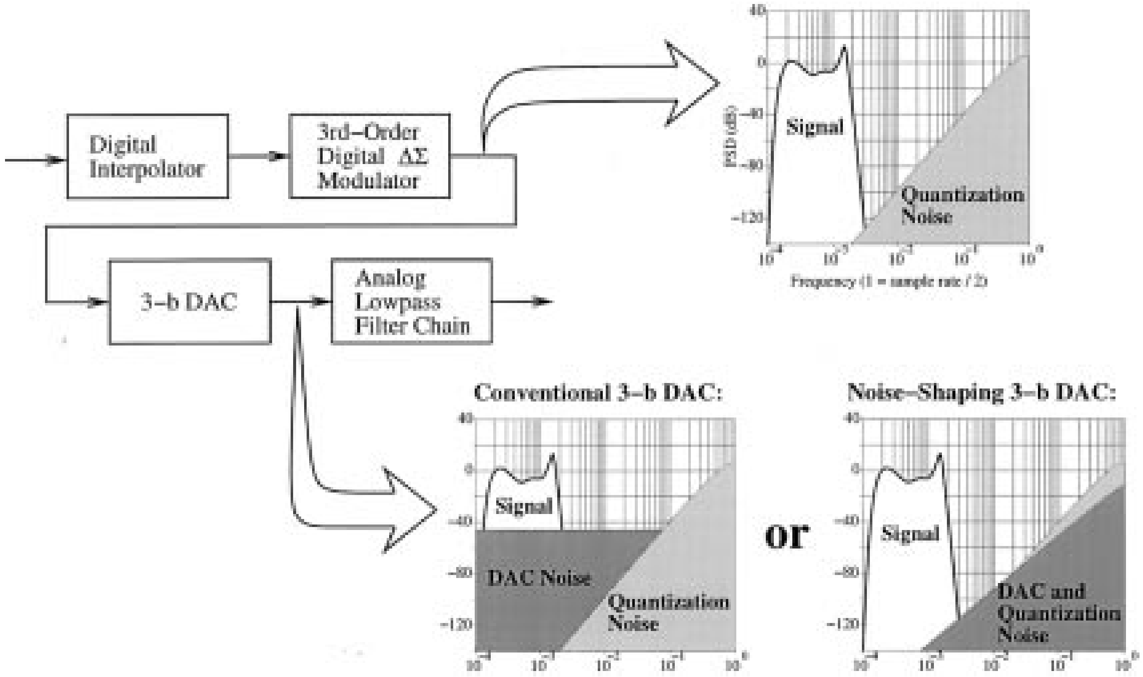


Fig. 1. Example of an oversampling 3-bit $\Delta\Sigma$DAC and (an artist's rendition of) typical power spectral densities of selected signals in the processing chain.

the least complex of those currently known to the author. Additionally, the paper provides rigorous theory that explains the apparent paradox of how it is possible to spectrally shape the DAC noise even though the sources of the DAC noise—the errors introduced by the analog circuitry—are not known to the noise-shaping algorithm. As part of the derivation, a set of sufficient conditions for achieving any given noise-shaping characteristics is developed.

The paper consists of two main sections and an Appendix. The general DAC architecture along with the first and second-order configurations are described in detail in Section II. In Section III, the goal is to provide an intuitively appealing, yet mathematically complete, explanation of the noise-shaping phenomenon. The results of Section III are extended in the Appendix to develop an expression that quantifies noise-shaping performance in terms of specific circuit errors.

II. The Noise-Shaping DAC Architecture and Performance

A. The Noise-Shaping DAC Approach

To illustrate the concepts outlined above and define terms that will be used throughout the paper, consider the example $\Delta\Sigma$DAC shown in Fig. 1. It consists of a digital interpolator, a third-order digital delta-sigma modulator, a three-bit DAC, and an analog lowpass filter (usually implemented as a chain of analog filters). As illustrated by the upper power spectral density (PSD) plot in the figure, the three-bit output of the $\Delta\Sigma$ modulator consists of a signal component restricted to a low frequency *signal band* and high frequency quantization noise, most of which falls outside of the signal band. Thus, in the absence of any errors introduced by the 3-bit DAC, the analog filter would remove most of the quantization noise, and the result would be a high-precision analog representation of the original digital input sequence. However, as illustrated in the lower left PSD plot in Fig. 1, step-size mismatches in conventional multibit DAC's result in *DAC noise* of which a substantial component resides within the signal band. Thus, even though the analog filter removes the components of the quantization and DAC noise outside the signal band, much of the DAC noise remains and spoils the overall conversion precision. In contrast, noise-shaping DAC's result in DAC noise that resides primarily outside the signal band as illustrated in the lower right PSD plot of Fig. 1, so the analog filter removes much of the DAC noise along with the quantization noise.

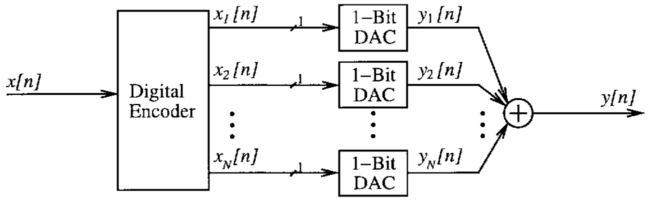


Fig. 2. The high-level architecture of most of the noise-shaping DAC's to date.

Most of the noise-shaping DAC architectures presented to date have the general form shown in Fig. 2 [9]–[15], [17]. Each consists of a *digital encoder* and N one-bit DAC's referred to as *unit DAC-elements*. The digital input sequence, $x[n]$, is taken to be a sequence of nonnegative integers less than or equal to N. The digital encoder maps each input sample to N output bits such that the sum of the N bits is equal to $x[n]$:

$$x_1[n] + x_2[n] + \cdots + x_N[n] = x[n]. \tag{1}$$

The unit DAC-elements operate according to

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

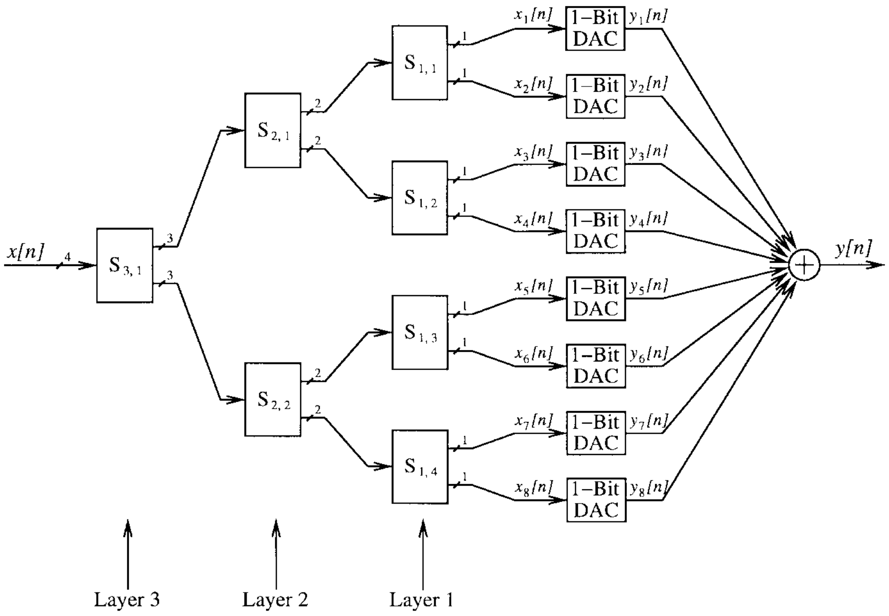


Fig. 3. The DAC architecture at the switching block level shown for the case of a 3-bit DAC.

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.[1]

The coarse DAC's required for $\Delta\Sigma$ data converters are typically implemented using switched-capacitor unit DAC-elements that each dump an ideally fixed amount of charge into the summing node of an op-amp based integrator during each sample interval in which the input bit is high [1], [18]–[22]. To a large extent, the relative errors introduced by the unit DAC-elements arise from mismatches in fabricated capacitor dimensions. Consequently, e_{h_r} and e_{l_r} are generally assumed to be time-invariant, but otherwise arbitrary, and are referred to as *static DAC-element errors*.

It follows that the DAC noise is a function of both $x[n]$ and the set of static DAC-element errors. All of the noise-shaping DAC architectures exploit the fact that the digital encoder has many ways that it can choose its output bits and still satisfy (1). Thus, the digital encoder can modulate the DAC noise component of $y[n]$ without affecting the signal component. The noise-shaping DAC's are able to perform this modulation such that the DAC noise is spectrally shaped as in the example associated with Fig. 1. Amazingly, they do this without requiring any knowledge of the specific static DAC-element error values.

B. The Switching Block DAC Architecture

A high-level view of the noise-shaping DAC architecture proposed in this paper is shown in Fig. 3. For simplicity, a 3-bit version is shown, but the architecture and all of the results outlined below are easily generalized to any number of bits.[2] The architecture is a special case of that shown in Fig. 2. The digital encoder consists of three *layers* of digital devices called *switching blocks* each of which is labeled $S_{k,\,r}$, where k denotes the layer number and r denotes the position of the switching block in the layer.

The details of the switching blocks will be described shortly, but at the high level of Fig. 3, each digital input sample, $x[n]$, is mapped by the three layers of switching blocks to 8 bits, $x_1[n]$, $x_2[n]$, $\cdots$, $x_8[n]$, that satisfy (1) with $N = 8$. It can be easily verified that this will happen provided each switching block satisfies the following *number conservation rule*:

Number Conservation Rule: The two outputs of each switching block must be in the range $\{0,\, 1,\, \cdots,\, 2^{k-1}\}$ where k is the layer number, and their sum must equal the input to the switching block.

For example, if $x[n_0] = 6$, then the number conservation rule is satisfied by the Layer 3 switching block if its two outputs at time n_0 are any of the following pairs: (3, 3), (4, 2), or (2, 4). For this example, provided all of the switching blocks satisfy the number conservation rule, then six of the bits: $x_1[n_0]$, $\cdots$, $x_8[n_0]$, would equal one, and two would equal zero.

From a signal processing point-of-view, the switching blocks perform the operations shown in Fig. 4, wherein, $s_{k,\,r}[n]$ is a sequence that is generated within each switching block and will be described shortly. It follows from the figure that

$$x_{k-1,\,2r-1}[n] = \tfrac{1}{2}\left(x_{k,\,r}[n] + s_{k,\,r}[n]\right)$$

and

$$x_{k-1,\,2r}[n] = \tfrac{1}{2}\left(x_{k,\,r}[n] - s_{k,\,r}[n]\right). \tag{3}$$

Because the sequence $s_{k,\,r}[n]$ determines the difference between the two outputs of the $S_{k,\,r}$ switching block, it follows that $s_{k,\,r}[n]$ must satisfy certain conditions for the number conservation rule to be satisfied. Specifically, it can be easily

[1] The choice of one and zero as the ideal output levels of the unit DAC-elements is arbitrary. Any other distinct pair of numbers could be used. The choice simply affects the gain and offset of the overall DAC.

[2] The DAC shown is actually a 9-level DAC because there are 8 unit DAC-elements. Thus, its input is slightly wider than 3-bits.

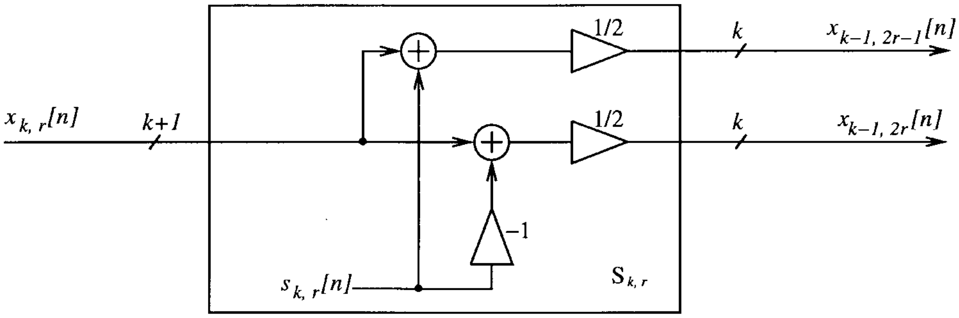


Fig. 4. The general form of all the switching blocks.

verified that if

$$s_{k,r}[n] = \begin{cases} \text{even if } x_{k,r}[n] \text{ is even} \\ \text{odd if } x_{k,r}[n] \text{ is odd} \end{cases} \tag{4}$$

and

$$|s_{k,r}[n]| \le \min\{x_{k,r}[n],\, 2^k - x_{k,r}[n]\} \tag{5}$$

for every k and r, then the switching blocks all satisfy the number conservation rule.

As derived in Section III and the Appendix, for a b-bit version (i.e., a b-layer version) of the DAC architecture shown in Fig. 3, wherein all the switching blocks operate as described by (3)–(5), the output of the DAC has the form $y[n] = \alpha x[n] + \beta + e[n]$, where

$$e[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r} s_{k,r}[n] \tag{6}$$

and α, β, and $\Delta_{k,r}$ are constants that depend only upon the static DAC-element errors. Specifically

$$\alpha = 1 + \frac{1}{2^b} \sum_{i=1}^{2^b} (e_{h_i} - e_{l_i}), \quad \beta = \sum_{i=1}^{2^b} e_{l_i} \tag{7}$$

and

$$\Delta_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} [(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})]. \tag{8}$$

Thus, α represents a constant gain, β represents a constant offset, and $e[n]$ can be viewed as the DAC noise. It follows from (6) that *if each switching block calculates $s_{k,r}[n]$ as an Lth-order shaped sequence that is uncorrelated from the $s_{k,r}[n]$ sequences of the other switching blocks, then the DAC noise will be an Lth-order shaped sequence*. The main difficulty here is that the $s_{k,r}[n]$ sequences must also be chosen to satisfy (4) and (5) in order to satisfy the number conservation rule.

C. The First- and Second-Order DAC Configurations

Structures that can be used to generate the $s_{k,r}[n]$ sequences for first- and second-order DAC's are shown in Fig. 5. Together with the structure in Fig. 4, the structures in Fig. 5(a) and (b) represent the signal processing operations performed by the complete $S_{k,r}$ switching block for first-order and second-order noise-shaping DAC configurations, respectively. As will be demonstrated shortly, simple gate-level implementations of these switching blocks exist, but the two structures shown in Fig. 5 will first be explained at the signal processing level.

The $S_{k,r}$ switching block associated with the first-order DAC generates $s_{k,r}[n]$ as shown in Fig. 5(a). The structure differs from a first-order $\Delta\Sigma$ modulator only in that it has no signal input and the hard limiter is followed by a least-significant-bit (LSB) multiplier that forces $s_{k,r}[n]$ to be zero whenever the LSB of the input to the switching block is zero (i.e., when the input is even). Because of the hard limiter, $s_{k,r}[n]$ is bounded in magnitude by unity, and the LSB multiplication causes (4) and (5) to be satisfied. Together, the hard limiter and LSB-multiplier can be viewed as introducing additive error, and the error is bounded in magnitude by unity because $|s_{k,r}[n]| \le 1$. Consequently, the structure behaves like a nonoverloading first-order $\Delta\Sigma$ modulator, so $s_{k,r}[n]$ is a first-order shaped version of the additive error. It follows from (6) that the resulting DAC noise is also first-order shaped noise.

The $S_{k,r}$ switching block associated with the second-order DAC generates $s_{k,r}[n]$ as shown in Fig. 5(b). The structure consists of two discrete-time integrators, a quantizer, and an amplitude limiter enclosed in a double feedback loop. The quantizer has a step-size of $\Delta_q = 2$ and performs midtread quantization when the LSB of the input to the switching block is zero and midrise quantization otherwise. In other words, the quantizer rounds to the nearest even integer when $x_{k,r}[n]$ is even and to the nearest odd integer otherwise. This quantizer arrangement forces $s_{k,r}[n]$ to satisfy (4). The amplitude limiter clips the output of the quantizer if necessary to force $s_{k,r}[n]$ to satisfy (5).

The limiter and quantizer together can be viewed as introducing additive error. An analysis of the filtering performed by the structure of Fig. 5(b) indicates that $s_{k,r}[n]$ is a second-order shaped version of this error. However, note that aside from the unusual quantizer and limiter, the structure differs from a conventional second-order $\Delta\Sigma$ modulator in that it has a 1/4 gain element preceding the quantizer and an extra feed-forward path between the outputs of the two integrators. The purpose of these modifications is to reduce the variance of the input to the quantizer so as to reduce the likelihood of exceeding the range of the limiter at any given time.

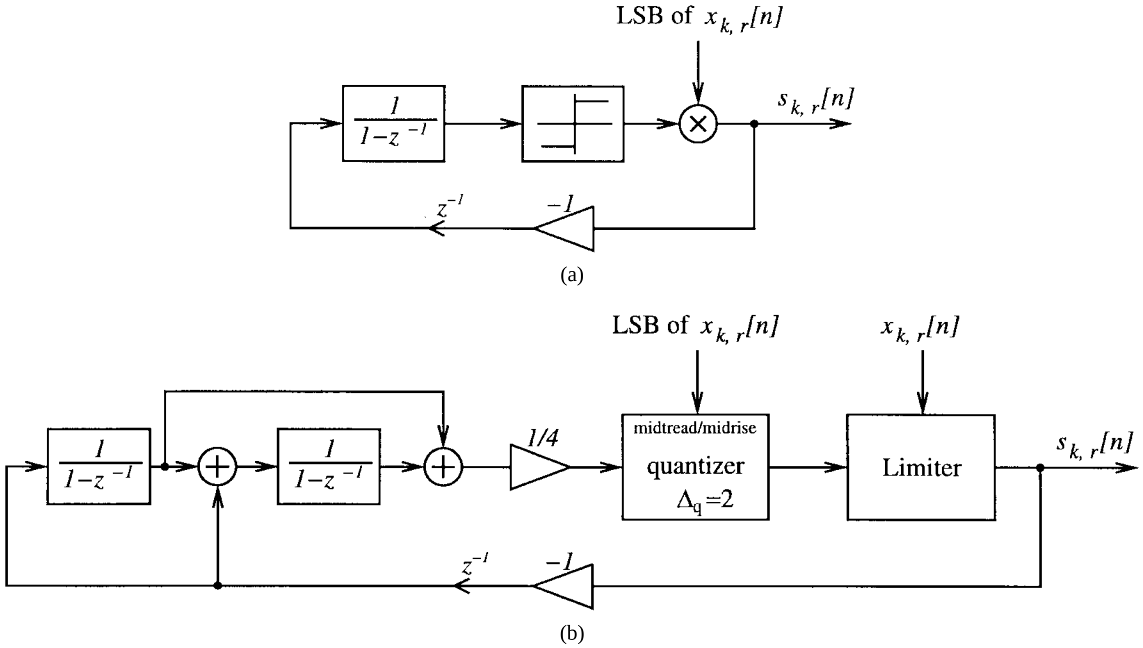


Fig. 5. Switching block calculation of $s_{k,r}[n]$ for (a) the first-order noise-shaping DAC, and (b) the second-order noise-shaping DAC.

Although the structures of Fig. 5 used by the switching blocks to generate the $s_{k,r}[n]$ sequences are similar to $\Delta\Sigma$ modulators, they differ in that they have no input sequences, so the discrete-time integrator inputs are always small integer values. If integer initial conditions are used, it follows that the state variables of the structures are always bounded integer values. Consequently, the structures can be implemented with low bit-width registers and arithmetic.

For example, one of many possible gate-level implementations of the switching block for the first-order DAC is shown in Fig. 6. It consists of two D flip-flops, and two LSB-adders (i.e., adders that each calculate the sum of a 1-bit number and a $k+1$-bit number). The LSB of the input to the switching block is used to gate the clock input of the two flip-flops which are connected in a four-state configuration. Each output of the switching block is taken as the k most-significant-bits of the sum of the input to the switching block and either the Q or $\overline{Q}$ output of the right-most flip-flop. Although the output of each adder is an odd number whenever the LSB of the input to the switching block is different from the corresponding output of the right-most flip-flop, the switching block outputs are never fractional because the divide-by-two operation is performed by the equivalent of right-shifting each adder output by one bit. Therefore, (4) is satisfied in effect.

It can be verified that the switching block performs the function of the system shown in Fig. 4 with $s_{k,r}[n]$ generated by the system shown in Fig. 5(a) for the appropriate choice of initial conditions. The two flip-flops in Fig. 6 play the role of the discrete-time integrator in the system of Fig. 5(a). Denoting the Q outputs of the two flip-flops as Q_1 and Q_2, respectively, it can be verified that the state ($Q_1=1$, $Q_2=1$) corresponds to an integrator state of 1, the state ($Q_1=0$, $Q_2=0$) corresponds to an integrator state of -1, and the states ($Q_1=0$, $Q_2=1$) and ($Q_1=1$, $Q_2=0$) both correspond to an integrator state of 0.

D. Spurious Tone Generation

From (6) it follows that any spurious tones contained in the switching block sequences, $s_{k,r}[n]$, will appear in the DAC noise. In the case of the second-order noise-shaping DAC, numerous simulations run by the author (an example of which is presented below) indicate that the spurious tones have extremely low amplitudes. However, the first-order noise-shaping DAC tends to generate larger-amplitude spurious tones for some inputs. A practical approach to suppressing these tones is as follows. For the structure of Fig. 5(a) with an initial condition of 0, it is easily verified that at each sample-time the input to the hard-limiter is either a 0, 1, or -1. By definition the output of the hard limiter is a 1 when its input is positive, and a -1 when its input is negative. Consequently, quantization error is introduced only when the input to the hard limiter is a 0 in which case the quantization error has a magnitude of 1. An input of 0 is midway between the two possible output values of the hard limiter. Thus, the total mean-squared error would not be changed if the hard limiter were replaced by a device that operates according to

$$v_{k,r}[n]=\begin{cases}1, & \text{if } u_{k,r}[n]=1;\\ -1, & \text{if } u_{k,r}[n]=-1;\\ r_{k,r}[n], & \text{if } u_{k,r}[n]=0\end{cases} \qquad (9)$$

where $v_{k,r}[n]$ is the output of the device, $u_{k,r}[n]$ is the input of the device, and $r_{k,r}[n]$ is a random ±1 sequence that is white, independent of $x[n]$, and uncorrelated with the $r_{k,r}[n]$ sequences in the other switching blocks. When the hard limiters in all the switching blocks are replaced by these devices, all spurious tones are eliminated.

In practice, this *dithering* scheme is relatively easy to implement. For example, in the switching block implementation shown in Fig. 6 an exclusive-OR gate with inputs Q_1 and Q_2 can be used to detect the equivalent of a discrete-time

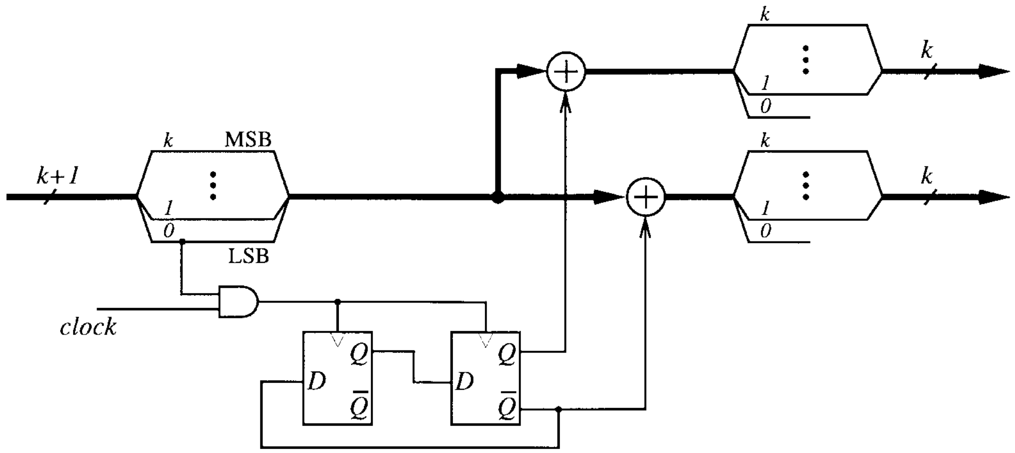


Fig. 6. A hardware-efficient implementation of the switching block for the first-order DAC.

integrator output of 0. When the output of the gate is high, simple logic can be used to randomly invert or not invert Q_2 and $\overline{Q}_2$. Random ±1 sequences that well-approximate the desired statistics are simple to generate using feedback shift registers [23].

A partial implementation of the dithering scheme can be used as a compromise between hardware complexity and spurious tone suppression. Specifically, if dithering is used in only some of the switching blocks, then some tone suppression is achieved and the resulting DAC is more hardware efficient than had dithering been used in all the switching blocks. For example, in the first-order noise-shaping DAC simulated below, only the $S_{3,1}$ switching block was dithered.

D. Simulation Results

The above claims will now be demonstrated via simulation results. For comparison, a *conventional DAC* and an *ideal DAC* are also simulated. The conventional DAC has the form of Fig. 2 with the digital encoder implemented as a so-called *thermometer encoder*. Hence, in the conventional DAC

$$x_k[n] = \begin{cases} 1, & \text{if } k \le x[n]; \\ 0, & \text{otherwise.} \end{cases}$$

The ideal DAC is introduces no DAC noise whatsoever.

Fig. 7(a) shows simulation results representing the output PSD's of a conventional DAC, a first-order noise-shaping DAC, a second-order noise-shaping DAC, and an ideal DAC, all driven by the same 3-bit third-order digital $\Delta\Sigma$ modulator with a sinusoidal excitation. All but the ideal DAC had the same set of static DAC-element errors which were chosen from a Gaussian distribution with a standard deviation of 1%.[3] The PSD's are in units of dB relative to the nominal LSB value of the DAC's (this is also the quantization step-size of the $\Delta\Sigma$ modulator), and the frequency scale is normalized such that unity corresponds to half the sample rate of the DAC output.

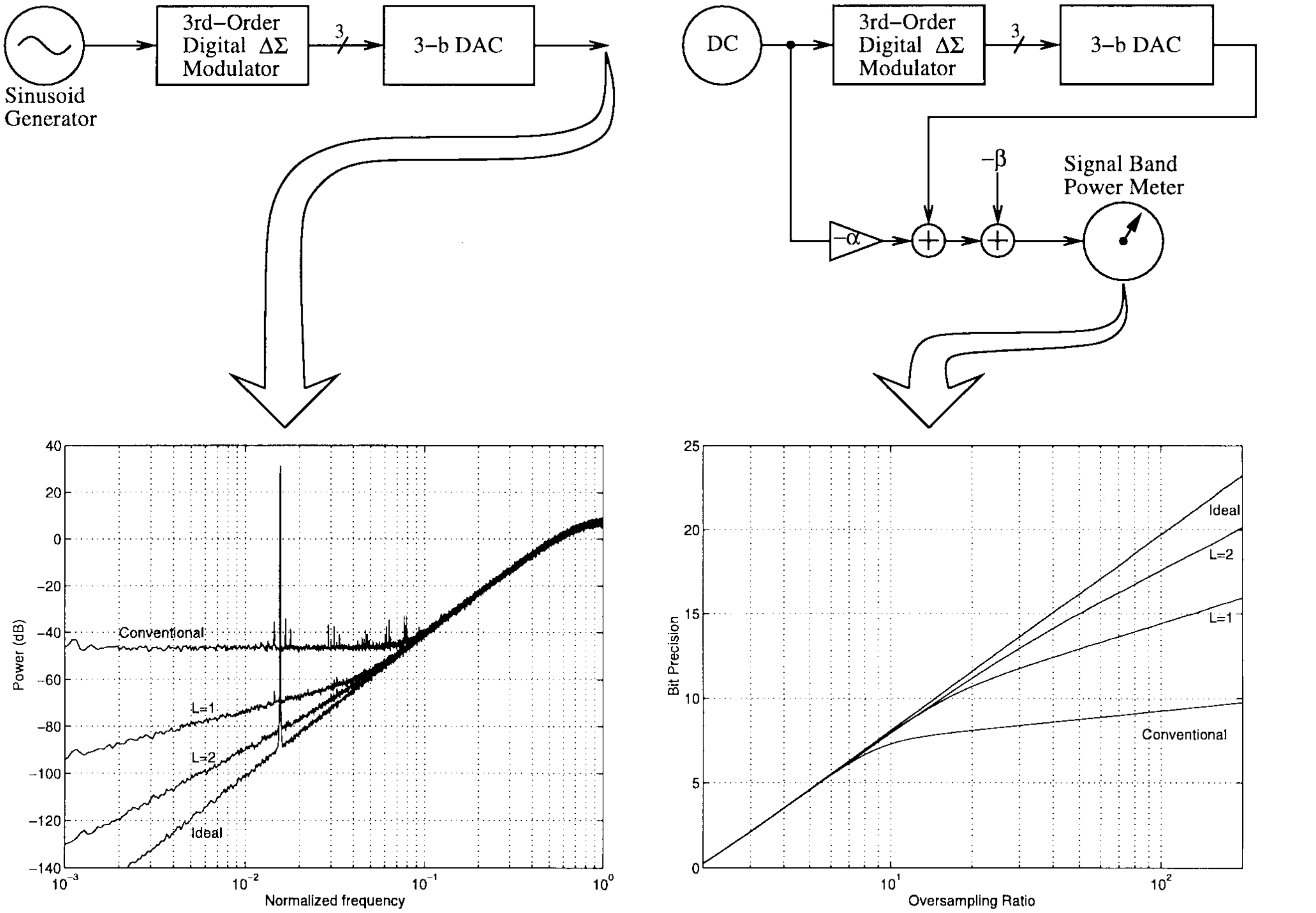
No DAC noise was introduced by the ideal DAC, so the PSD of the ideal DAC output is equal to that of the 3-bit, third-order digital $\Delta\Sigma$ modulator output. Accordingly, the PSD consists of third-order shaped quantization noise with a discrete spectral frequency component corresponding to the sinusoidal excitation of the $\Delta\Sigma$ modulator. As expected from well-known $\Delta\Sigma$ modulator results, the quantization noise component decreases by 18 dB per octave decrease in frequency. Each of the PSD's associated with the other three DAC's differs from that of the ideal DAC because of an additional component corresponding to the DAC noise. As is evident from Fig. 7(a), for the conventional DAC this component gives rise to a flat noise-floor at about −45 dB with a considerable spurious tone content. For the first-order and second-order DAC's the component gives rise to noise-floors that decrease by 6 and 12 dB per octave decrease in frequency, respectively, with very little spurious tone content. Thus, the simulations clearly support the assertions made above that the first-order and second-order DAC's give rise to first-order and second-order shaped DAC noise, respectively.

Fig. 7(b) shows attainable bit precisions versus oversampling ratios between 2 and 200. As shown schematically in the figure, each bit precision curve was obtained by driving the $\Delta\Sigma$ modulator followed by one of the four DAC's with a constant input, subtracting the constant input and DAC offset from the DAC output (using the expressions for α and β given above), and then calculating the power within the signal band of the resulting sequence for the range of oversampling ratios. A constant input was used to calculate the achievable bit precisions only to avoid causing the conventional DAC to introduce spurious tones. Other input sequences, e.g., sinusoidal input sequences, would have yielded very similar results for all but the conventional DAC.

III. Theoretical Explanation of the Noise-Shaping DAC's

In this section, the basic theoretical idea underlying the noise-shaping DAC's of this paper is explained. Although the presentation is mathematically rigorous, the objective of this

[3] The choice of a Gaussian distribution here is incidental. As proven in Section III and the Appendix, the noise-shaping does not depend upon the distribution or correlation properties of the static DAC-element errors.



(a) (b)

Fig. 7. (a) Simulated PSD's of conventional, first-order noise-shaping, second-order noise-shaping, and ideal DAC's all driven by the same 3-bit third-order digital $\Delta\Sigma$ modulator. (b) The corresponding bit precision limits versus oversampling ratio.

section is to provide intuition and insight into the theory. The results of this section are further developed in the Appendix to arrive at (6)–(8).

A functionally equivalent version of the DAC architecture of Fig. 3 is shown in Fig. 8(a) from which the recursive nature of the system is evident. Each subsystem enclosed in a dashed box is a DAC in its own right and, thus, is referred to as $\mathrm{DAC}_{k,\,r}$ where k and r are the indexes of the left-most switching block in $\mathrm{DAC}_{k,\,r}$. The recursive form of $\mathrm{DAC}_{k,\,r}$ in terms of $\mathrm{DAC}_{k-1,\,2r-1}$ and $\mathrm{DAC}_{k-1,\,2r}$ is shown in Fig. 8(b). From a comparison of the notation in Figs. 3 and 8, it follows that $\mathrm{DAC}_{0,\,r}$ is the rth unit DAC-element, and $\mathrm{DAC}_{3,\,1}$ is the 3-bit DAC itself.

By analyzing the recursive form of the DAC shown in Fig. 8(b) and using the principle of induction, the noise-shaping operation of the DAC can be explained and quantified. Although the analysis is not difficult, the subscripts become confusing so it is convenient to temporarily redraw the recursive DAC structure of Fig. 8(b) using the simplified notation shown in Fig. 9. The boxes labeled S, DAC_1, and DAC_2 in Fig. 9 correspond to those labeled $S_{k,\,r}$, $\mathrm{DAC}_{k-1,\,2r-1}$, and $\mathrm{DAC}_{k-1,\,2r}$ in Fig. 8(b). Thus, $x'[n] = x_{k,\,r}[n]$, $y'[n] = y_{k,\,r}[n]$, and so on. With this simplified notation, (3) implies

$$s[n] = x_1'[n] - x_2'[n] \tag{10}$$

and the number conservation rule implies

$$x'[n] = x_1'[n] + x_2'[n]. \tag{11}$$

Any nonideal DAC (or any single-input, single-output device, for that matter) can be viewed from a signal processing point-of-view as a device that introduces a constant gain, a dc offset, and zero-mean DAC noise. For convenience in the derivation that follows, the dc offset and zero-mean DAC noise are lumped together and referred to as *additive noise*. Thus, the outputs of the two DAC's in Fig. 9 can be written as

$$y_i'[n] = \alpha_i x_i'[n] + \epsilon_i[n] \tag{12}$$

for $i = 1$ and $i = 2$, where α_i and $\epsilon_i[n]$ are the gain and additive noise, respectively, associated with DAC_i.

From (12) and Fig. 9, it follows that

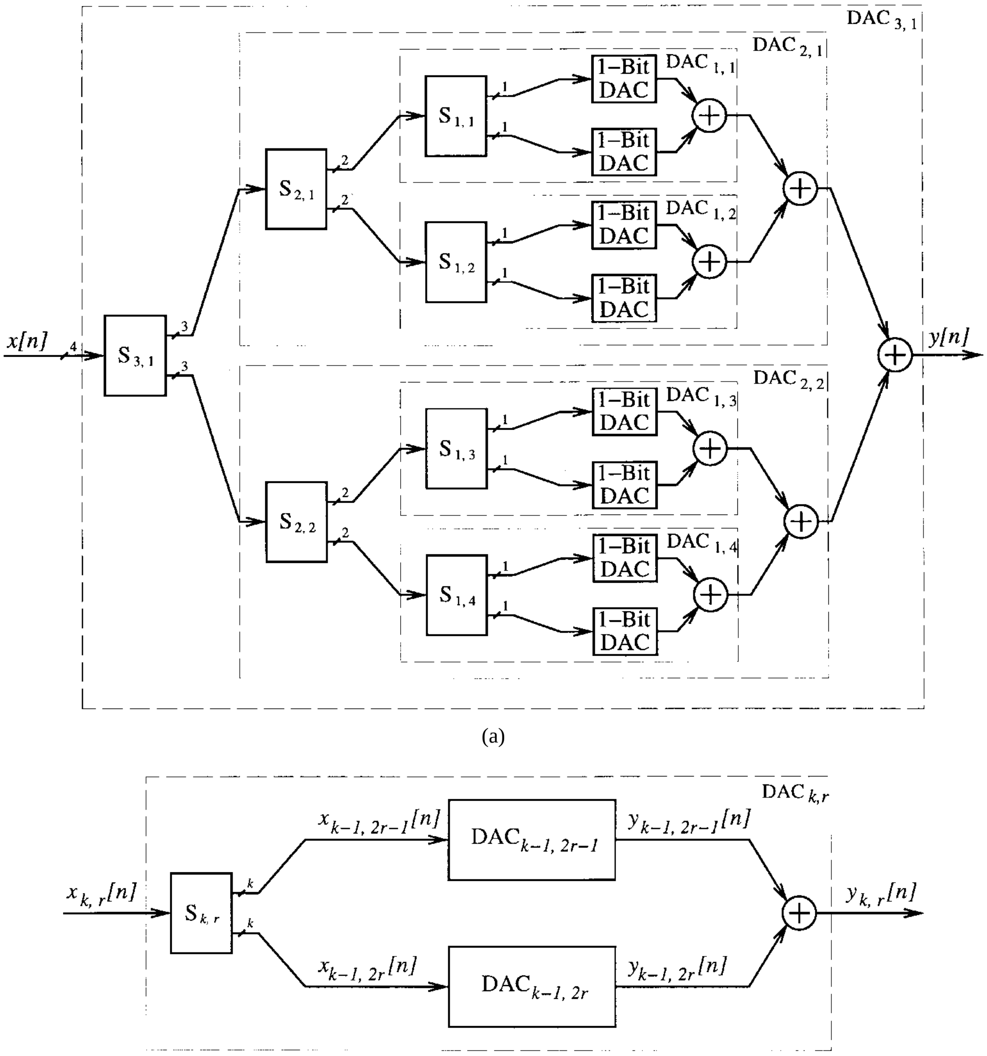
$$y'[n] = \alpha_1 x_1'[n] + \alpha_2 x_2'[n] + \epsilon_1[n] + \epsilon_2[n]$$

which can be written as

$$y'[n] = \frac{\alpha_1 + \alpha_2}{2}\,(x_1'[n] + x_2'[n]) + \frac{\alpha_1 - \alpha_2}{2} \cdot (x_1'[n] - x_2'[n]) + \epsilon_1[n] + \epsilon_2[n].$$

By applying (10) and (11), and collecting terms, this becomes

$$y'[n] = \alpha x'[n] + \epsilon[n], \tag{13}$$



(a)

(b)

Fig. 8. (a) A redrawn but functionally equivalent version of the DAC shown in Fig. 2. (b) The contents of each dashed box in (a).

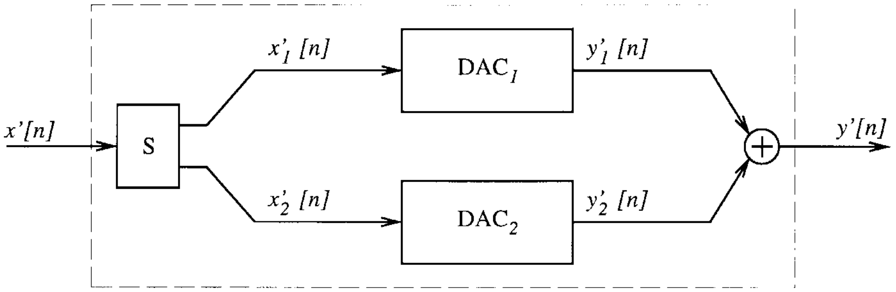


Fig. 9. The recursive DAC structure of Fig. 8(b) drawn with simplified notation.

where

$$\epsilon[n] = s[n]\Delta + \epsilon_1[n] + \epsilon_2[n] \tag{14}$$

$$\alpha = \frac{\alpha_1 + \alpha_2}{2}, \text{ and } \Delta = \frac{\alpha_1 - \alpha_2}{2}. \tag{15}$$

At this point, the key observation can be made.[4] Equation (14) indicates that the DAC noise of the recursive DAC of Fig. 9 consists of the DAC noise introduced by its two component DAC's and a component arising from the mismatch between the gains of its component DAC's. Suppose that Fig. 9 corresponds to $\text{DAC}_{1,1}$ in Fig. 8(a). That is, suppose S, DAC_1, and DAC_2 in Fig. 9 correspond to $S_{1,1}$ and the first and second unit DAC-elements in Fig. 8(a), respectively. From (12) and the definition of the static DAC-element errors, it is easily verified that for this case $\alpha_i = 1 + e_{h_i} - e_{l_i}$ and $\epsilon_i[n] = e_{l_i}$ for $i = 1$ and $i = 2$ so the additive noise from

[4] An excellent alternative explanation of this observation that applies to a three-level noise-shaping DAC is presented in [13].

each unit DAC-element is just a dc offset. Consequently, from (14), it follows that the only nonconstant component of the additive noise introduced by DAC$_{1,1}$ is contributed by the term $s[n]\Delta$. But Δ is a constant, so if $s[n] = s_{1,1}[n]$ is an Lth-order shaped sequence, it follows that the DAC noise will be an Lth-order shaped sequence regardless of the values of the static DAC-element errors. The same argument applies to DAC$_{1,2}$, DAC$_{1,3}$, and DAC$_{1,4}$ in Fig. 8(a).

Now suppose that Fig. 9 corresponds to DAC$_{2,1}$. That is, suppose S, DAC$_1$, and DAC$_2$ in Fig. 9 correspond to $S_{2,1}$, DAC$_{1,1}$, and DAC$_{1,2}$ in Fig. 8(a), respectively. By the argument made above, provided $s_{1,1}[n]$ and $s_{1,2}[n]$ are uncorrelated Lth-order shaped sequences, then the sum of the second two terms in (14) will also be an Lth-order shaped sequence. Consequently, provided $s[n] = s_{2,1}[n]$ is an Lth-order shaped sequence that is uncorrelated with $s_{1,1}[n]$ and $s_{1,2}[n]$, then $\epsilon[n]$, or, equivalently, the DAC noise introduced by DAC$_{2,1}$, is also an Lth-order shaped sequence. By induction, this argument applies to the entire DAC of Fig. 8(a), and more generally to DAC's of arbitrary size having the switching block architecture wherein all the switching blocks satisfy the number conservation rule.

IV. Conclusion

This paper presents a general DAC architecture that can be used to spectrally shape DAC noise arising from analog circuit mismatches, and practical first and second-order noise-shaping DAC configurations that are special cases of the general architecture. Additionally, it presents a theoretical derivation that explains the apparent paradox of how an all-digital algorithm that has no knowledge of the values of the analog circuit mismatches can result in well-controlled spectral shaping of the DAC noise caused by the mismatches.

The results of the paper extend the practicality of noise-shaping DAC's for use in $\Delta\Sigma$ data converters, and provide a theoretical foundation for the further development of noise-shaping DAC's. The first-order DAC configuration is as hardware-efficient as those presented previously, but offers the advantage that it is amenable to a simple dithering technique capable of eliminating spurious tones. The second-order DAC configuration is the most hardware-efficient of those currently known to the author, and generally introduces very little spurious tone content even without dithering. The theory presented provides sufficient conditions for achieving any given noise-shaping characteristics using the general architecture. Additional research is necessary to determine the extent to which these conditions can be satisfied for higher than second-order noise-shaping.

Appendix

The purpose of this appendix is to derive (6)–(8) starting from the results of Section III. Specifically, (13)–(15) which were derived in Section III using the simplified notation of Fig. 9, are used to develop the nonrecursive formulations of the constant gain, dc offset, and DAC noise.

The first step is to rewrite (13)–(15) using the full notation of Fig. 8(b) as

$$y_{k,r}[n] = \alpha_{k,r}x_{k,r}[n] + \epsilon_{k,r}[n] \tag{16}$$

$$\epsilon_{k,r}[n] = s_{k,r}[n]\Delta_{k,r} + \epsilon_{k-1,2r-1}[n] + \epsilon_{k-1,2r}[n] \tag{17}$$

$$\alpha_{k,r} = \frac{\alpha_{k-1,2r-1}[n] + \alpha_{k-1,2r}[n]}{2}$$

and

$$\Delta_{k,r} = \frac{\alpha_{k-1,2r-1}[n] - \alpha_{k-1,2r}[n]}{2}. \tag{18}$$

It will now be shown that recursively applying the expression for $\alpha_{k,r}$ beginning with $k = 1$ and keeping r fixed gives

$$\alpha_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{r2^k} \alpha_{0,i}. \tag{19}$$

That this is true for $\alpha_{1,r}$ follows directly from (18). Suppose that (19) holds for some particular value of $k' > 1$. In this case, applying (18) gives

$$\alpha_{k'+1,r} = \frac{1}{2}\cdot\frac{1}{2^{k'}}\left\{\sum_{i=(2r-2)2^{k'}+1}^{(2r-1)2^{k'}} \alpha_{0,i} + \sum_{i=(2r-1)2^{k'}+1}^{2r2^{k'}} \alpha_{0,i}\right\}$$

which reduces to (19) for $k = k' + 1$. Consequently, by the principle of induction, (19) must hold for each positive integer k.

As mentioned above, $\alpha_{0,i}$ is just the gain introduced by the ith unit DAC-element which can be written as $\alpha_{0,i} = 1 + e_{h_i} - e_{l_i}$, so (19) becomes

$$\alpha_{k,r} = 1 + \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{r2^k} (e_{h_i} - e_{l_i}). \tag{20}$$

Substituting this into the expression for $\Delta_{k,r}$ above gives

$$\Delta_{k,r} = \frac{1}{2^k} \sum_{i=(r-1)2^k+1}^{(r-1)2^k+2^{k-1}} [(e_{h_i} - e_{l_i}) - (e_{h_{i+2^{k-1}}} - e_{l_{i+2^{k-1}}})]. \tag{21}$$

Finally, recursively applying (17) beginning with $k = 1$ gives

$$\epsilon_{b,1}[n] = \sum_{k=1}^{b} \sum_{r=1}^{2^{b-k}} \Delta_{k,r}s_{k,r}[n] + \sum_{i=1}^{2^b} e_{l_i}. \tag{22}$$

This can be verified by an induction argument similar to that used above to prove (19).

Equations (20)–(22) directly imply (6)–(8).

Acknowledgment

The author would like to thank S. Norsworthy of Motorola for introducing him to the problem considered in this paper, and for many helpful discussions related to the paper.

References

[1] *Delta-Sigma Data Converters, Theory, Design, and Simulation*, S. R. Norsworthy, R. Schreier, and G. C. Temes, Eds. New York: IEEE Press, 1997.

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

[3] K. C. H. Chao, S. Nadeem, W. L. Lee, and C. G. Sodini "A higher order topology for interpolative modulators for oversampling A/D converters," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 309–318, Mar. 1990.

[4] R. Schreier, "An empirical study of high-order, single-bit delta-sigma modulators," *IEEE Trans. Circuits Syst. II,* vol. 40, pp. 461–466, Aug. 1993.

[5] D. B. Ribner, "A comparison of modulator networks for high-order oversampled $\Sigma\Delta$ analog-to-digital converters," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 145–159, Feb. 1991.

[6] B. P. Brandt and B. A. Wooley, "A 50-MHz multibit sigma-delta modulator for 12-b 2-MHz A/D conversion," *IEEE J. Solid-State Circuits*, vol. 26, Dec. 1991.

[7] M. A. Alexander, H. Mohajeri, and J. O. Prayogo,"A 192 ks/s sigma-delta ADC with integrated decimation filters providing -97.4 dB THD," *IEEE ISSCC Dig. Tech. Papers*, vol. 37, pp. 190–191, 1994.

[8] M. Sarhang-Nejad and G. C. Temes, "A high-resolution multi-bit $\Sigma\Delta$ ADC with digital correction and relaxed amplifier requirements," *IEEE J. Solid-State Circuits,* vol. 28, June 1993.

[9] B. H. Leung and S. Sutarja, "Multi-bit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques," *IEEE Trans. Circuits Syst. II,* vol. 39, pp. 35–51, Jan. 1992.

[10] F. Chen and B. H. Leung, "A high resolution multibit sigma-delta modulator with individual level averaging," *IEEE J. Solid-State Circuits,* vol. 30, pp. 453–460, Apr. 1995.

[11] R. W. Adams and T. W. Kwan, "Data-directed scrambler for multi-bit noise shaping D/A converters," U.S. Patent 5 404 142, Apr. 4, 1995.

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

[13] ——, "A stereo multibit Sigma Delta DAC with asynchronous master-clock interface," *IEEE J. Solid-State Circuits,* vol. 31, pp. 1881–1887, Dec. 1996.

[14] R. T. Baird and T. S. Fiez, "Linearity enhancement of multi-bit $\Delta\Sigma$ A/D and D/A converters using data weighted averaging," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 753–762, Dec. 1995.

[15] R. Schreier and B. Zhang, "Noise-shaped multi-bit D/A converter employing unit elements," *Electron. Lett.,* vol. 31, no. 20, pp. 1712–1713, Sept. 28, 1995.

[16] R. K. Henderson and O. J. A. P. Nys, "Dynamic element matching techniques with arbitrary noise shaping functions," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1996.

[17] I. Galton, "Noise-shaping D/A converters for Delta Sigma modulation," in *Proc. IEEE Int. Symp. Circuits Syst.,* May 1996, vol. 1, pp. 441–444.

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

[19] M. Ismail and T. Fiez, *Analog VLSI.* New York: McGraw-Hill, 1994.

[20] T. Choi, R. Kaneshiro, P. Gray, W. Jett, and M. Wilcox, "High-frequency CMOS switched-capacitor filters for communications applications," *IEEE J. Solid-State Circuits,* vol. SC-18, pp. 652–664, Dec. 1983.

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

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

[23] S. W. Golomb, *Shift Register Sequences.* San Francisco, CA: Holden-Day, 1967.

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

Prior to 1988, he was with Acuson, Inc., Mountain View, CA, working 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, La Jolla, 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.

Dr. Galton holds four patents and received the Caltech Charles Wilts Doctoral Thesis Prize.