Second and Third-Order Noise Shaping Digital Quantizers for Low Phase Noise and Nonlinearity-Induced Spurious Tones in Fractional-N PLLs

Eythan Familier, *Member, IEEE*, and Ian Galton, *Fellow, IEEE*

***Abstract*—Noise shaping digital quantizers, most commonly digital delta-sigma $(\Delta\Sigma)$ modulators, are used in fractional-N phase-locked loops (PLLs) to enable fractional frequency tuning. Unfortunately, their quantization noise is subjected to nonlinear distortion because of the PLL's inevitable non-ideal analog circuit behavior, which induces spurious tones in the PLL's phase error. Successive requantizers have been proposed as $\Delta\Sigma$ modulator replacements with the advantage that they reduce the power of these spurious tones. However, the quantization noise from previously published successive requantizers is only first-order highpass shaped, so it usually causes more PLL phase noise than that from the second-order and third-order $\Delta\Sigma$ modulators commonly used in PLLs. This paper presents second-order and third-order successive requantizers to address this limitation. Additionally, successive requantizer design options are presented that result in either lower-power spurious tones or lower phase noise compared to $\Delta\Sigma$ modulators when used in PLLs.**

***Index Terms*—DC-free quantization noise, noise-shaping quantizers, spurious tones.**

I. Introduction

FRACTIONAL-N phase-locked loops (PLLs) typically incorporate all-digital delta-sigma $(\Delta\Sigma)$ modulators to enable fractional frequency tuning [1]–[3]. A $\Delta\Sigma$ modulator's output sequence can be written as the sum of its input sequence plus *quantization noise*. The quantization noise causes the PLL's phase error to contain a component proportional to a lowpass filtered version of the running sum of the quantization noise [4].[1] In practice, non-ideal analog circuit behavior in the PLL causes the PLL's phase error to also contain components proportional to nonlinearly distorted versions of both the quantization noise and its running sum. Unfortunately, these nonlinearly distorted sequences contain spurious tones, even when the quantization noise and its running sum are free of spurious tones [5]–[15]. This is problematic in high-performance applications such as wireless communication systems which tend to be extremely sensitive to spurious tones.

The successive requantizer was proposed in [7] as a digital $\Delta\Sigma$ modulator replacement to address this issue. The nonlinearities to which the quantization noise and its running sum are subjected in PLLs tend to be well approximated by truncated memoryless power series [7], [8]. Therefore, the successive requantizer in [7] was designed to produce quantization noise, $s[n]$, with the property that $s^p[n]$ for $p = 1, 2, 3, 4,$ and 5 are free of spurious tones, and such that its running sum, $t[n]$, has the property that $t^p[n]$ for $p = 1, 2,$ and 3 are free of spurious tones. The successive requantizer was demonstrated in a PLL with record-setting spurious tone performance in [8]. Unfortunately, its quantization noise is only first-order highpass shaped, whereas most $\Delta\Sigma$ modulators used in PLLs have second-order or third-order highpass shaped quantization noise to reduce the quantization noise contribution to the PLL's phase noise [4].[2] This issue was addressed in [8] via a quantization noise cancelation technique at the expense of increased PLL circuit area and power consumption.

This paper presents extensions of previously published results that enable successive requantizers with second-order and third-order highpass shaped quantization noise to address this limitation. It also presents design techniques that optimize the successive requantizers to either minimize PLL phase noise or spurious tone power depending on the PLL's target specifications.

In both cases, the successive requantizers achieve higher than first-order quantization noise shaping in return for not ensuring that $s^p[n]$ for $p \geq 2$ is free of spurious tones. In practice, this is not a significant limitation because in most PLLs the frequency divider output edges are resynchronized to voltage controlled oscillator edges which tends to make the nonlinear distortion applied to $s[n]$ negligible [4]. Therefore, the design option presented in the paper to minimize spurious tones focuses on

Manuscript received October 11, 2015; revised January 22, 2016; accepted February 15, 2016. Date of current version July 15, 2016. This work was supported by the National Science Foundation under Award 1343389, by Analog Devices, and by corporate members of the UCSD Center for Wireless Communications. This paper was recommended by Associate Editor A. Mazzanti.

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: eythanfc@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2016.2571598

[1] In this paper, a PLL's *phase error* refers to the difference between the PLL's actual and ideal phases and a PLL's *phase noise* refers to all stochastic components of its phase error (e.g., such as those caused by thermal noise).

[2] A sequence is said to be first-order highpass shaped if its running sum is bounded but the running sum of its running sum (i.e., its double running sum) is not bounded. Similarly, a sequence is said to be second-order highpass shaped if both its running sum and its double running sum are bounded but its triple running sum is not bounded.

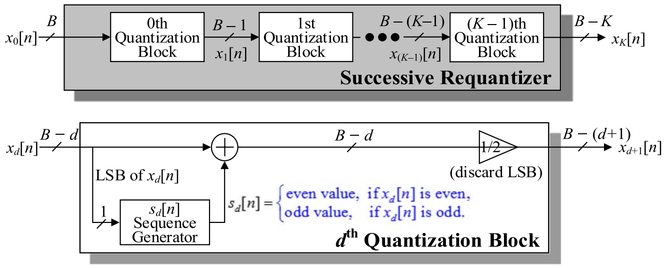

B
$x_0[n]$
0th Quantization Block
$B-1$
$x_1[n]$
1st Quantization Block
$B-(K-1)$
$x_{(K-1)}[n]$
$(K-1)$th Quantization Block
$B-K$
$x_K[n]$
Successive Requantizer
$B-d$
$x_d[n]$
$B-d$
1/2
$B-(d+1)$
$x_{d+1}[n]$
(discard LSB)
LSB of $x_d[n]$
1
$s_d[n]$ Sequence Generator
$s_d[n] = \begin{cases} \text{even value, if } x_d[n] \text{ is even,} \\ \text{odd value, if } x_d[n] \text{ is odd.} \end{cases}$
d^{th} Quantization Block

Fig. 1. High-level diagram of a successive requantizer.

nonlinearity applied to the quantization noise running sum at the expense of only a slight increase in PLL phase noise. Specifically, it ensures that $t^p[n]$ for $p = 1, 2, \ldots, h$ are free of spurious tones, where h is a positive integer. The other design option results in successive requantizers that, like $\Delta\Sigma$ modulators, do not ensure that $t^p[n]$ is free of spurious tones for $p \geq 2$. Instead, they offer the advantage of introducing lower PLL phase noise than their $\Delta\Sigma$ modulator counterparts when used in typical PLLs.

Therefore, the contribution of this paper is a family of replacements for the commonly-used second-order and third-order $\Delta\Sigma$ modulators in fractional-N PLLs, each member of which either improves PLL spurious-tone performance at the expense of slightly higher PLL phase noise or lowers PLL phase noise. Unlike previous work, this is achieved without the high area and power consumption of phase noise cancellation techniques.

The paper consists of four main sections. Section II presents the second-order successive requantizer architecture. Section III presents two second-order successive requantizer designs which highlight a tradeoff between nonlinearity-induced spurious tone power and low-frequency quantization noise power. Section IV presents a second-order successive requantizer with lower low-frequency quantization noise power than a second-order $\Delta\Sigma$ modulator. Section V presents a third-order successive requantizer with lower low-frequency quantization noise power than a third-order $\Delta\Sigma$ modulator.

II. Second-Order Successive Requantizer Architecture

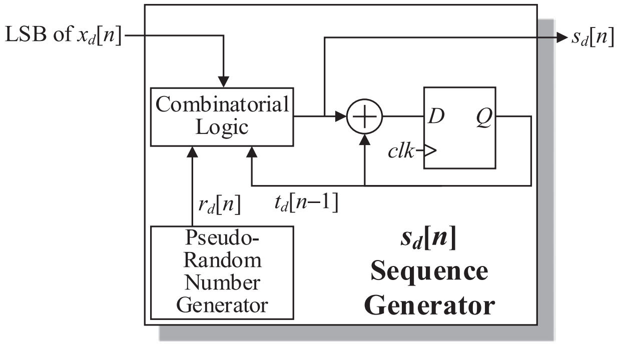


Fig. 2. Block diagram of a first-order $s_d[n]$ sequence generator.

A high-level diagram of a successive requantizer is shown in Fig. 1. Its sequences are all integer-valued and represented in two's complement format. It processes a B-bit input sequence $x_0[n]$ through K serially-connected *quantization blocks* to produce a $(B-K)$-bit output sequence $x_K[n]$. The dth quantization block, for each $d = 0, 1, \ldots, K-1$, divides its input, $x_d[n]$, by two and quantizes the result by one bit such that its output sequence has the form

$$x_{d+1}[n] = \frac{x_d[n] + s_d[n]}{2} \tag{1}$$

where $s_d[n]/2$ can be viewed as quantization noise. The $s_d[n]$ *sequence generator* generates $s_d[n]$ to have the same parity as $x_d[n]$ for all n (otherwise $x_{d+1}[n]$ would not be integer-valued) and with a small enough magnitude that $x_{d+1}[n]$ can be represented with one less bit than $x_d[n]$. As explained in [7], it follows that the output of the successive requantizer can be written as

$$x_K[n] = 2^{-K} x_0[n] + s[n] \tag{2}$$

where

$$s[n] = \sum_{d=0}^{K-1} 2^{d-K} s_d[n] \tag{3}$$

is the quantization noise of the successive requantizer.

The running sums of $s_d[n]$ and $s[n]$ are defined as

$$t_d[n] = \sum_{k=0}^{n} s_d[k], \quad t[n] = \sum_{k=0}^{n} s[k] \tag{4}$$

respectively. Therefore, (3) implies that

$$t[n] = \sum_{d=0}^{K-1} 2^{d-K} t_d[n]. \tag{5}$$

It follows from (3)–(5) that the statistical properties of $s[n]$ and $t[n]$ are determined by the behavior of the $s_d[n]$ sequence generators.

A first-order $s_d[n]$ sequence generator, i.e., one in which $s_d[n]$ is first-order highpass shaped, is shown in Fig. 2 [7]. It contains a pseudo-random number generator that outputs a sequence of independent, identically and uniformly distributed pseudo-random variables $r_d[n]$, a delayed accumulator block that takes $s_d[n]$ and outputs $t_d[n-1]$, and a combinatorial logic block that generates $s_d[n]$ as a function of the lowest significant bit (LSB) of $x_d[n]$, $t_d[n-1]$, and $r_d[n]$ such that $t_d[n]$ is bounded for all n. It follows that $t_d[n]$ is a deterministic function of $t_d[n-1]$, the parity of $x_d[n]$, and $r_d[n]$.

Fig. 3 shows the proposed second-order $s_d[n]$ sequence generator. It contains a pseudo-random number generator that outputs $r_d[n]$ as in the first-order case, a combinatorial logic block, and two difference blocks. As shown in the figure, the combinatorial logic block generates a bounded sequence $u_d[n]$ conditioned on its delayed version $u_d[n-1]$, a *parity sequence* $o_d[n]$, and $r_d[n]$. As can be seen from Fig. 3

$$t_d[n] = u_d[n] - u_d[n-1] \tag{6}$$

$$s_d[n] = t_d[n] - t_d[n-1] \tag{7}$$

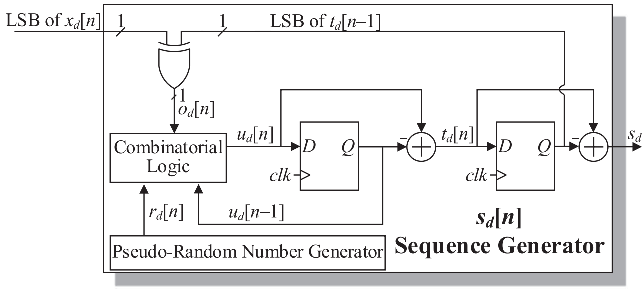


Fig. 3. Block diagram of a second-order $s_d[n]$ sequence generator.

so $u_d[n]$ is the running sum of $t_d[n]$ and the double running sum of $s_d[n]$. Unlike in the first-order $s_d[n]$ sequence generator, in the second-order $s_d[n]$ sequence generator both the running sum and the double running sum of $s_d[n]$ are bounded, so $s_d[n]$ is second-order highpass shaped.

As described above, $s_d[n]$ must have the same parity as $x_d[n]$ for all n, i.e.,

$$x_d[n] \bmod 2 = s_d[n] \bmod 2. \tag{8}$$

This imposes some restrictions on the combinatorial logic. It can be seen from Fig. 3 that

$$o_d[n] = (x_d[n] + t_d[n-1]) \bmod 2 \tag{9}$$

which, with (6), can be written as

$$o_d[n] = (x_d[n] + u_d[n-1] - u_d[n-2]) \bmod 2. \tag{10}$$

Equations (6) and (7) imply that

$$s_d[n] = u_d[n] - 2u_d[n-1] + u_d[n-2]. \tag{11}$$

For any integers a and b, $[(a \bmod 2) + (b \bmod 2)] \bmod 2 = (a+b) \bmod 2$, and $(2a) \bmod 2 = 0$, so it follows from (8), (10), and (11) that the combinatorial logic must generate $u_d[n]$ such that

$$(u_d[n] + u_d[n-1]) \bmod 2 = o_d[n]. \tag{12}$$

The number of bits in the successive requantizer output is determined by the range of values covered by $x_0[n]$ and by the $s_d[n]$ sequences. It follows from (1) that for each $0 \le m \le K-1$:

$$x_m[n] + s_m[n] = 2^{-m}x_0[n] + \sum_{d=0}^{m} 2^{d-m}s_d[n] \tag{13}$$

$$x_{m+1}[n] = 2^{-m-1}x_0[n] + \sum_{d=0}^{m} 2^{d-m-1}s_d[n]. \tag{14}$$

Let N_s be the smallest positive integer greater than or equal to 2 such that

$$|s_d[n]| \le N_s \text{ for all } n \text{ and each } d \in \{0, 1, \ldots, K-1\}. \tag{15}$$

If the range of the input to the successive requantizer is restricted as

$$-2^K \le x_0[n] \le 0 \tag{16}$$

it follows from (13)–(15) that:

$$-2N_s - 2^{K-m} < x_m[n] + s_m[n] < 2N_s \text{ for } 0 \le m \le K-1 \tag{17}$$

$$-N_s - 2^{K-m} < x_m[n] < N_s \text{ for } 1 \le m \le K. \tag{18}$$

This implies that $x_m[n] + s_m[n]$ for $0 \le m \le K-1$ and $x_m[n]$ for $1 \le m \le K$ can be represented with $\lceil \log_2(2^{K-m+1}N_s) \rceil$ bits, where, for any value x, $\lceil x \rceil$ is the smallest integer greater than or equal to x. The dth quantization block represents $x_d[n]$ and $x_d[n] + s_d[n]$ with $B-d$ bits and $x_{d+1}[n]$ with $B-(d+1)$ bits, so it follows that the successive requantizer requires at least $B = K + 1 + \lceil \log_2(N_s) \rceil$ bits for its input, or, equivalently, $1 + \lceil \log_2(N_s) \rceil$ bits for its output.

For example, if $N_s = 8$, as in some of the example successive requantizers in the following section, and (16) holds, the successive requantizer input must have at least $B = K + 4$ bits, and its output must have at least 4 bits to cover the range $\{-8, -7, \ldots, 7\}$. A similar analysis yields that if the range of $x_0[n]$ is restricted as

$$-2^K \le x_0[n] \le 2^K \tag{19}$$

then the successive requantizer requires at least $B = K + 2 + \lceil \log_2(N_s) \rceil$ bits for its input, or, equivalently, $2 + \lceil \log_2(N_s) \rceil$ bits for its output.

As illustrated via examples in the next sections, the choice of N_s represents a tradeoff between PLL spurious tone performance and quantization noise power: increasing N_s provides flexibility which can be used to improve spurious tone performance, but it also tends to increase quantization noise power [12].

It follows from Figs. 1 and 3 that the computational complexity of the successive requantizer is a logarithmic function of N_s and a quadratic function of the number of quantization blocks K. The dth pseudo-random number generator can be implemented with a modified linear-feedback shift register (LFSR) that simultaneously generates multiple bits that are well-modeled as zero-mean, white, and independent of each other [16].

III. Second-Order Successive Requantizers With High Immunity to Spurious Tones

The combinatorial logic block of a second-order $s_d[n]$ sequence generator can be described by two *state transition matrices*, $\mathbf{A_e}$ and $\mathbf{A_o}$, which define the probability mass function (pmf) of $u_d[n]$ conditioned on $u_d[n-1]$ and $o_d[n]$ for each n. Specifically, if $u_d[n]$ takes values in $\{N_u, N_u - 1, \ldots, -N_u\}$, where N_u is a positive integer, then $\mathbf{A_e}$ and $\mathbf{A_o}$ are $(2N_u + 1) \times (2N_u + 1)$ matrices with elements

$$\mathbf{A_e}(i,j) = \Pr(u_d[n] = \mathbf{u}(j) | u_d[n-1] = \mathbf{u}(i), o_d[n] = 0)$$
$$\mathbf{A_o}(i,j) = \Pr(u_d[n] = \mathbf{u}(j) | u_d[n-1] = \mathbf{u}(i), o_d[n] = 1) \tag{20}$$

for all $i, j \in \{1, 2, \ldots, 2N_u + 1\}$, where[3]

$$\mathbf{u} = \begin{pmatrix} N_u & (N_u - 1) & \cdots & -N_u \end{pmatrix}^T . \tag{21}$$

The combinatorial logic block is deterministic, so the probabilities in (20) arise from the random sequence, $r_d[n]$, which is assumed, by design, to be uniformly distributed for all n. This implies that the probabilities in (20) must be of the form $k/2^b$, where $k \in \{0, 1, \ldots, 2^b\}$ and b is the number of bits used to represent $r_d[n]$.[4] The only other requirements the state transition matrices must satisfy are that their rows add to 1—so that the pmf is valid—and that

$$\begin{aligned} \mathbf{A_e}(i,j) &= 0 \quad \forall \quad i+j : \text{ odd and} \\ \mathbf{A_o}(i,j) &= 0 \quad \forall \quad i+j : \text{ even.} \end{aligned} \tag{22}$$

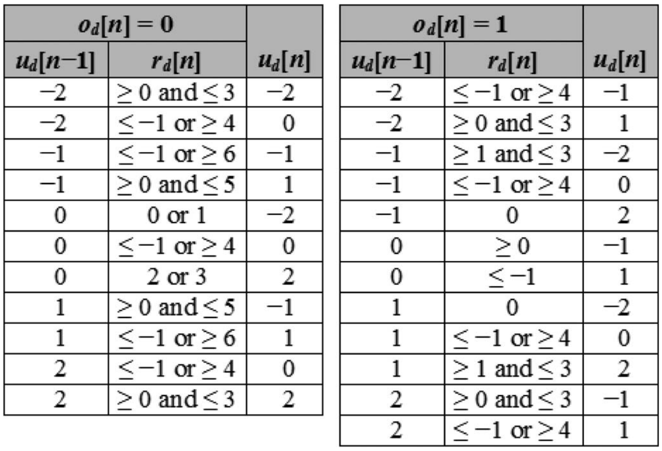
The last requirement is needed to satisfy (12). Equation (21) implies that if $i + j$ is even then $\mathbf{u}(i) + \mathbf{u}(j)$ is even, whereas if $i + j$ is odd then $\mathbf{u}(i) + \mathbf{u}(j)$ is odd. This and (20) imply that if $\mathbf{A_e}(i,j) \neq 0$ for some odd $i + j$, there is a non-zero probability that $u_d[n-1] + u_d[n]$ is odd when $o_d[n] = 0$, which contradicts (12). Similarly, if $\mathbf{A_o}(i,j) \neq 0$ for some even $i + j$, there is a non-zero probability that $u_d[n-1] + u_d[n]$ is even when $o_d[n] = 1$, which contradicts (12) as well.

As an example, if $N_u = 2$ and $r_d[n] \in \{-8, -7, \ldots, 7\}$, then

$$\mathbf{A_e} = \begin{pmatrix} \frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 \\ 0 & \frac{5}{8} & 0 & \frac{3}{8} & 0 \\ \frac{1}{8} & 0 & \frac{3}{4} & 0 & \frac{1}{8} \\ 0 & \frac{3}{8} & 0 & \frac{5}{8} & 0 \\ 0 & 0 & \frac{3}{4} & 0 & \frac{1}{4} \end{pmatrix} \text{ and}$$

$$\mathbf{A_o} = \begin{pmatrix} 0 & \frac{3}{4} & 0 & \frac{1}{4} & 0 \\ \frac{3}{16} & 0 & \frac{3}{4} & 0 & \frac{1}{16} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{16} & 0 & \frac{3}{4} & 0 & \frac{3}{16} \\ 0 & \frac{1}{4} & 0 & \frac{3}{4} & 0 \end{pmatrix} \tag{23}$$

describe valid behavior for the combinatorial logic block. A truth table for combinatorial logic that implements the behavior specified by (23) can be constructed from (20) and (21) with $N_u = 2$. For example, the elements in the third row of $\mathbf{A_e}$, i.e., 1/8, 0, 3/4, 0, and 1/8, are the probabilities that $u_d[n] = \boldsymbol{u}(1) = 2$, $u_d[n] = \boldsymbol{u}(2) = 1$, $u_d[n] = \boldsymbol{u}(3) = 0$, $u_d[n] = \boldsymbol{u}(4) = -1$, and $u_d[n] = \boldsymbol{u}(5) = -2$, respectively, conditioned on $u_d[n-1] = \boldsymbol{u}(3) = 0$ and $o_d[n] = 0$. Therefore, if $u_d[n-1] = 0$ and $o_d[n] = 0$, the combinatorial logic must set $u_d[n] = 2$ with probability 1/8, $u_d[n] = 0$ with probability 3/4, and $u_d[n] = -2$ with probability 1/8. Given that $r_d[n]$ is uniformly distributed among the sixteen integers from -8 to 7, one way to do this is to map two of these integers to $u_d[n] = 2$, another two to $u_d[n] = -2$, and the rest to $u_d[n] = 0$; e.g., set $u_d[n] = -2$ if $r_d[n] = 0$ or 1, $u_d[n] = 2$ if $r_d[n] = 2$ or 3, and $u_d[n] = 0$ otherwise. A complete truth table, an example of which is shown in Fig. 4, can be constructed by applying this procedure to every row of $\mathbf{A_e}$ and $\mathbf{A_o}$.



$o_d[n] = 0$		$u_d[n]$
$u_d[n-1]$	$r_d[n]$	
−2	≥ 0 and ≤ 3	−2
−2	≤ −1 or ≥ 4	0
−1	≤ −1 or ≥ 6	−1
−1	≥ 0 and ≤ 5	1
0	0 or 1	−2
0	≤ −1 or ≥ 4	0
0	2 or 3	2
1	≥ 0 and ≤ 5	−1
1	≤ −1 or ≥ 6	1
2	≤ −1 or ≥ 4	0
2	≥ 0 and ≤ 3	2

$o_d[n] = 1$		$u_d[n]$
$u_d[n-1]$	$r_d[n]$	
−2	≤ −1 or ≥ 4	−1
−2	≥ 0 and ≤ 3	1
−1	≥ 1 and ≤ 3	−2
−1	≤ −1 or ≥ 4	0
−1	0	2
0	≥ 0	−1
0	≤ −1	1
1	0	−2
1	≤ −1 or ≥ 4	0
1	≥ 1 and ≤ 3	2
2	≥ 0 and ≤ 3	−1
2	≤ −1 or ≥ 4	1

Fig. 4. Example truth table for the combinatorial logic block described by the state transition matrices in (23), with $r_d[n] \in \{-8, -7, \ldots, 7\}$.

Note that the rows in both $\mathbf{A_e}$ and $\mathbf{A_o}$ of (23) alternate between two types of vectors: vectors whose odd-indexed elements are zero, referred to as even-entries vectors, and vectors whose even-indexed elements are zero, referred to as odd-entries vectors. This is a consequence of (22) and holds for all valid state transition matrices.

State transition matrices such as those in (23) were used in [13] to describe the combinatorial logic block in first-order $s_d[n]$ sequence generators. For such $s_d[n]$ sequence generators, the state transition matrices define the pmf of $t_d[n]$ conditioned on $t_d[n-1]$ and a parity sequence. In contrast, the state transition matrices in second-order $s_d[n]$ sequence generators define the pmf of $u_d[n]$ conditioned on $u_d[n-1]$ and the parity sequence $o_d[n]$, as described by (20) and (21). Therefore, if the same state transition matrices are used to describe the combinatorial logic block of a first and a second-order $s_d[n]$ sequence generator, and if the parity sequences of both generators are equal, $t_d[n]$ in the first-order $s_d[n]$ sequence generator and $u_d[n]$ in the second-order $s_d[n]$ sequence generator are statistically equivalent. Furthermore, since $t_d[n]$ and $u_d[n]$ are defined as the running sums of $s_d[n]$ and $t_d[n]$, respectively, $s_d[n]$ in the first-order $s_d[n]$ sequence generator and $t_d[n]$ in the second-order $s_d[n]$ sequence generator are statistically equivalent as well.

For any positive integer h, a sequence $x[n]$ is said to be *immune to spurious tones up to order* h if $x^p[n]$ for $p = 1, 2, \ldots, h$ are free of spurious tones. In [13], conditions are presented on the state transition matrices of a first-order successive requantizer, i.e., a successive requantizer which uses first-order $s_d[n]$ sequence generators, that make $t_d[n]$ and $s_d[n]$ immune to spurious tones up to orders h_1 and h_2, respectively, for each d, where h_1 and h_2 are positive integers which do not depend on the parity sequences of the $s_d[n]$ sequence generators. It is also shown in [13] that such conditions make $t[n]$ and $s[n]$ immune to spurious tones up to orders h_1 and h_2, respectively. Therefore, in a second-order successive requantizer, such conditions make $u_d[n]$ and $t_d[n]$ immune to spurious tones up to orders h_1

[3] In this paper, the ith entry of a vector $\mathbf{v}$ is denoted by $\mathbf{v}(i)$, whereas the ith row, jth column entry of a matrix $\mathbf{M}$ is denoted by $\mathbf{M}(i,j)$.

[4] Since $u_d[n]$ conditioned on $u_d[n-1]$ and $o_d[n]$ is a deterministic function of $r_d[n]$, the 2^b values $r_d[n]$ can take map to $M \leq 2^b$ values of $u_d[n]$ for each $u_d[n-1]$ and $o_d[n]$. Since $r_d[n]$ is uniformly distributed, the probability that $u_d[n] = u$ equals $k/2^b$, where k is the number of different $r_d[n]$ values that map to u.

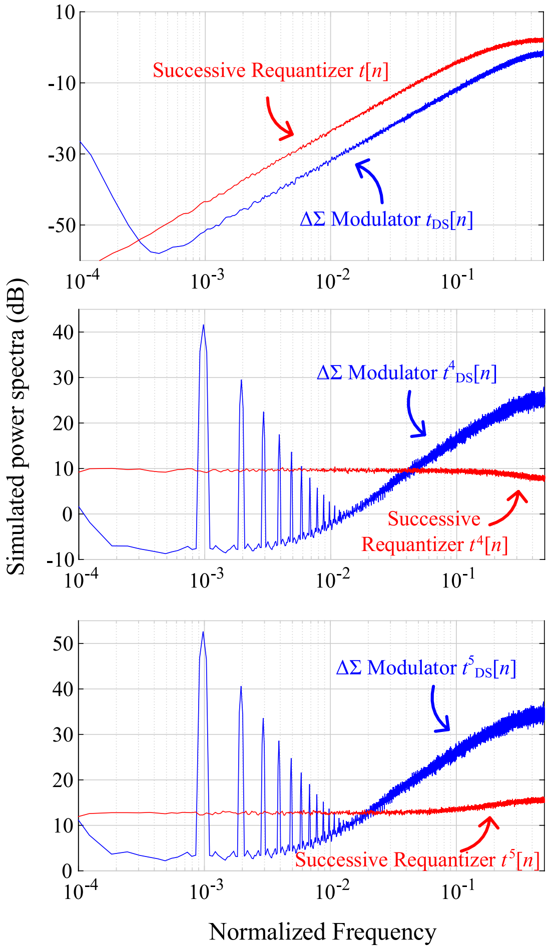


Fig. 5. Simulated power spectra of the running sum of the quantization noise of a second-order successive requantizer that implements the state transition matrices in (23) and a second-order ΔΣ modulator before and after the application of fourth and fifth-order nonlinear distortion.

and h_2, respectively, for each d. Additionally, as can be verified with identical reasoning to that used in [13] for the first-order successive requantizer, they make

$$u[n] = \sum_{k=0}^{n} t[k] \tag{24}$$

and $t[n]$ immune to spurious tones up to orders h_1 and h_2, respectively. For example, it was proven in [13] that, in a first-order successive requantizer, state transition matrices (23) make $s[n]$ immune to spurious tones up to order 5. Therefore, in a second-order successive requantizer, these state transition matrices make $t[n]$ immune to spurious tones up to order 5. Fig. 5 shows plots of simulated power spectra of $t[n]$ as generated with a second-order successive requantizer that implements the state transition matrices in (23) and of the running sum of the quantization noise of a dithered second-order ΔΣ modulator, $t_{\mathrm{DS}}[n]$, before and after the application of fourth and fifth-order nonlinear distortion. As expected, the simulated power spectra of $t^4[n]$ and $t^5[n]$ show no visible spurious tones, whereas those of $t^4_{\mathrm{DS}}[n]$ and $t^5_{\mathrm{DS}}[n]$ do.

Fig. 5 also shows that $t[n]$ has significantly higher low-frequency power spectrum content than $t_{\mathrm{DS}}[n]$. One of the contributions of this paper is to reduce this content by modifying $\mathbf{A_e}$ and $\mathbf{A_o}$ subject to $t[n]$ maintaining a minimum desired order of immunity to spurious tones.

It follows from Theorem 2 and Lemmas 3 and 6 in [13] and the parallels of first and second-order successive requantizers described above that sufficient conditions to make $t[n]$ immune to spurious tones up to order h_t, where h_t is a positive integer, are that

C1) $\mathbf{A_e}$ and $\mathbf{A_o}$ are centrosymmetric[5] with all their odd-entries row vectors containing at least $1+\lfloor (N_u+1)/2 \rfloor$ nonzero entries and all their even-entries row vectors containing at least $1+\lfloor N_u/2 \rfloor$ nonzero entries (where $\lfloor x \rfloor$ denotes the greatest integer that is less than or equal to x), and that

C2) For each positive even integer $p \le h_t$

$$\begin{aligned}\mathbf{A_e T_e t}^{(p)} &= \mathbf{A_e T_o t}^{(p)} = \mathbf{A_o T_e t}^{(p)} \\ &= \mathbf{A_o T_o t}^{(p)} = c_p \mathbf{1}_{2N_u+1}\end{aligned} \tag{25}$$

where c_p is a constant

$$\mathbf{T_x}(i,j) = \begin{cases} \mathbf{A_x}(i, j+i-2N_u-1), & \text{if } 2N_u+2-i \le j \le 4N_u+2-i \\ 0, & \text{if } j \le 2N_u+1-i, j \ge 4N_u+3-i \end{cases} \tag{26}$$

for $\mathbf{x} = \mathbf{e}$ or $\mathbf{o}$

$$\mathbf{t}^{(p)} = \begin{pmatrix} (2N_u)^p & (2N_u-1)^p & \cdots & (-2N_u)^p \end{pmatrix}^T \tag{27}$$

and $\mathbf{1}_{2N_u+1}$ is a length-$(2N_u+1)$ vector whose elements are all 1.

For example, if $\mathbf{A_e}$ and $\mathbf{A_o}$ are given by (23), $\mathbf{T_e}$ and $\mathbf{T_o}$, computed using (26), are given by

$$\mathbf{T_e} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{8} & 0 & \frac{3}{8} & 0 & 0 \\ 0 & 0 & \frac{1}{8} & 0 & \frac{3}{4} & 0 & \frac{1}{8} & 0 & 0 \\ 0 & 0 & \frac{3}{8} & 0 & \frac{5}{8} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$\mathbf{T_o} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{3}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{3}{16} & 0 & \frac{3}{4} & 0 & \frac{1}{16} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{16} & 0 & \frac{3}{4} & 0 & \frac{3}{16} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{28}$$

The above conditions hold regardless of how the successive requantizer is initialized.

As explained above, the entries in $\mathbf{A_e}$ and $\mathbf{A_o}$ are constrained to be of the form $k/2^b$, where $k \in \{0, 1, \ldots, 2^b\}$ and b is the number of bits used to represent each $r_d[n]$. This implies that there is a finite number of matrices $\mathbf{A_e}$ and $\mathbf{A_o}$ which satisfy conditions **C1** and **C2** for each N_u and h_t. As justified in the

[5] An $N \times N$ centrosymmetric matrix $\mathbf{A}$ is a matrix for which $\mathbf{A}(i,j) = \mathbf{A}(N+1-i, N+1-j)$ for all i, j.

$o_d[n]=0$		$u_d[n]$
$u_d[n-1]$	$r_d[n]$	
−2	≥ −512 and ≤ 153	0
−2	≥ 154	2
−1	≥ 0 and ≤ 55	−1
−1	≤ −1 or ≥ 56	1
0	≥ 0 and ≤ 178	−2
0	≤ −1 or ≥ 358	0
0	≥ 179 and ≤ 357	2
1	≤ −1 or ≥ 56	−1
1	≥ 0 and ≤ 55	1
2	≥ 154	−2
2	≥ −512 and ≤ 153	0

$o_d[n]=1$		
$u_d[n-1]$	$r_d[n]$	$u_d[n]$
−2	≥ 0 and ≤ 6	−1
−2	≤ −1 or ≥ 7	1
−1	−512 or −511	−2
−1	≥ −510 and ≤ 155	0
−1	≥ 156	2
0	≥ 0	−1
0	≤ −1	1
1	≥ 156	−2
1	≥ −510 and ≤ 155	0
1	−512 or −511	2
2	≤ −1 or ≥ 7	−1
2	≥ 0 and ≤ 6	1

Fig. 6. Example truth table for the combinatorial logic block described by the state transition matrices in (31), with $r_d[n] \in \{-512, -511, \ldots, 511\}$.

Appendix, the low-frequency quantization noise can be reduced by choosing the $\mathbf{A_e}$ and $\mathbf{A_o}$ matrices that minimize

$$\lim_{H\to\infty} \sum_{m=-H}^{H} \sum_{i=1}^{2N_u+1} \left(\mathbf{A}^{|m|}\mathbf{u}\right)(i) \cdot \mathbf{u}(i) \cdot \left(\lim_{k\to\infty} \mathbf{A}^k\right)(1,i) \tag{29}$$

where

$$\mathbf{A} = \frac{(\mathbf{A_e} + \mathbf{A_o})}{2} \tag{30}$$

subject to the constraint that $\mathbf{A}$ be primitive, i.e., that there exists a positive integer n such that the entries of $\mathbf{A}^n$ are all greater than zero. Because $\mathbf{A}$ is stochastic, a necessary and sufficient condition for $\mathbf{A}$ to be primitive is that all entries of $\mathbf{A}^{4N_u+1}$ are greater than zero [17]. In this paper, a computer program was written which cycles through all $\mathbf{A_e}$ and $\mathbf{A_o}$ matrices that satisfy conditions **C1** and **C2** for $N_u = 2$ and $h_t = 3$ and picks those which minimize (29), where the limits in (29) are approximated with a suitable number of terms (i.e., a number high enough that increasing it has no visible effect on the simulated power spectrum of $t[n]$). To keep hardware implementation requirements modest, $r_d[n]$ was restricted to values that can be represented with 10 bits, so the entries in $\mathbf{A_e}$ and $\mathbf{A_o}$ are restricted to be of the form $k/1024$, where $k \in \{0, \ldots, 1024\}$. Note, however, that larger values of N_u and h_t might require $r_d[n]$ to be represented with more than 10 bits. The resulting state transition matrices are

$$\mathbf{A_e} = \begin{pmatrix} 0 & 0 & \frac{333}{512} & 0 & \frac{179}{512} \\ 0 & \frac{7}{128} & 0 & \frac{121}{128} & 0 \\ \frac{179}{1024} & 0 & \frac{333}{512} & 0 & \frac{179}{1024} \\ 0 & \frac{121}{128} & 0 & \frac{7}{128} & 0 \\ \frac{179}{512} & 0 & \frac{333}{512} & 0 & 0 \end{pmatrix}$$

$$\mathbf{A_o} = \begin{pmatrix} 0 & \frac{7}{1024} & 0 & \frac{1017}{1024} & 0 \\ \frac{1}{512} & 0 & \frac{333}{512} & 0 & \frac{89}{256} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{89}{256} & 0 & \frac{333}{512} & 0 & \frac{1}{512} \\ 0 & \frac{1017}{1024} & 0 & \frac{7}{1024} & 0 \end{pmatrix}. \tag{31}$$

An example combinatorial logic block truth table of a second-order $s_d[n]$ sequence generator that implements these state transition matrices is shown in Fig. 6.

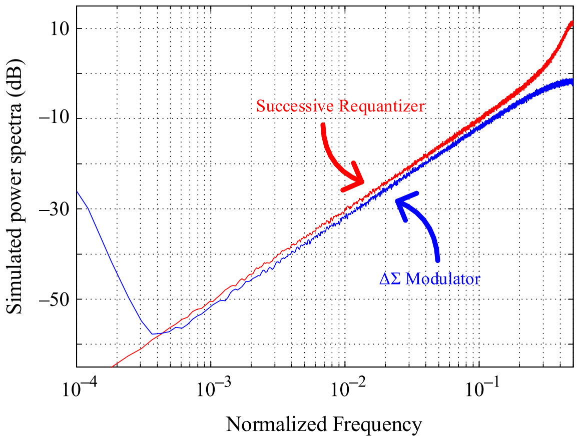


Fig. 7. Simulated power spectra of the running sum of the quantization noise of a second-order successive requantizer that implements the state transition matrices in (31) and of a second-order ΔΣ modulator.

Fig. 7 shows the simulated power spectrum of $t[n]$ from a 20-bit input second-order successive requantizer that implements the state transition matrices in (31). Given that $u_d[n]$ is the double running sum of $s_d[n]$ and is bounded by $N_u = 2$, $s_d[n]$ is bounded by $N_s = 8$. Therefore, as explained in the previous section, for inputs restricted as in (16), this successive requantizer's output ranges from -8 to 7. For comparison, the figure also shows the simulated power spectrum of the running sum of the quantization noise of a dithered second-order ΔΣ modulator, $t_{\text{DS}}[n]$. At low frequencies, the simulated power spectrum of $t[n]$ is only approximately 1.5 dB above that of $t_{\text{DS}}[n]$. At high frequencies the power spectrum of $t[n]$ is higher than that of $t_{\text{DS}}[n]$, but as explained shortly this difference negligibly affects phase noise in typical fractional-N PLL designs. The range covered by $t[n]$ is $(-4, 4)$, which is higher than that covered by $t_{\text{DS}}[n]$. However, $t_{\text{DS}}[n]$ is only immune to spurious tones up to order 1, whereas $t[n]$ is immune to spurious tones up to order 3. Additionally, as can be seen in the figure, $t_{\text{DS}}[n]$ contains an integrated white noise component due to the ΔΣ modulator's dither, whereas $t[n]$ does not.

To evaluate the feasibility of using this successive requantizer in a PLL, an event-driven PLL simulator was written in C to compare the power spectrum of the PLL phase error when using the second-order successive requantizer against when using a dithered second-order ΔΣ modulator as the PLL's digital quantizer. The simulator models the PLL shown in Fig. 8 which consists of a phase-frequency detector (PFD), a charge pump (CP) with nominal branch currents of 1 mA, a third-order loop filter with component values of $C_1 = 67$ pF, $R_1 = 8.67$ kΩ, $C_2 = 2.02$ nF, $R_2 = 8.67$ kΩ, and $C_3 = 67$ pF, a voltage-controlled oscillator (VCO) with a gain of 5 MHz/V, two digital quantizers—a second-order ΔΣ modulator and the second-order successive requantizer described above—with an input, α, of -2^{-11}, and a multi-modulus frequency divider with a modulus of 137 plus the output of one of the digital quantizers [4]. The PLL has a reference frequency f_{ref} of 26 MHz, an output frequency f_{out} of 3.56 GHz, a bandwidth of 45 kHz, and a phase margin of 55°. The circuit noise sources modeled in the simulation are $1/f^2$ noise from the VCO (-125 dBc/Hz at 1 MHz) and white noise from the reference

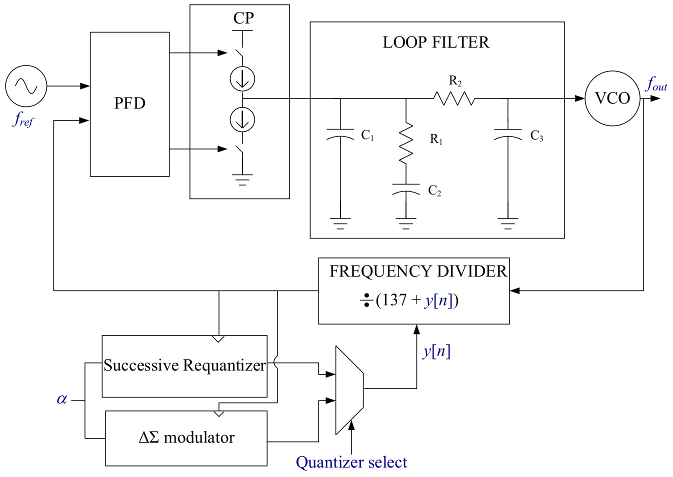


Fig. 8. Block diagram of the PLL used in phase error simulations.

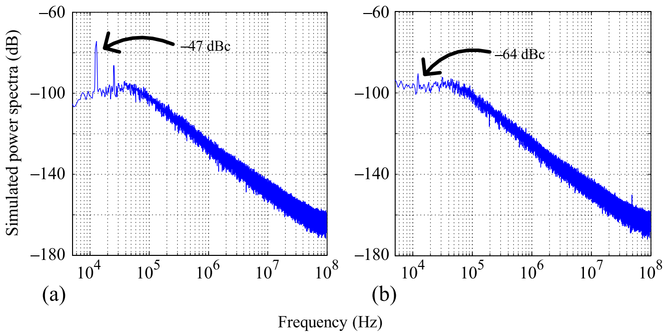


Fig. 9. Simulated power spectra of the phase error of a 3.56 GHz output frequency, 45 kHz bandwidth PLL when its digital quantizer is implemented as (a) a second-order $\Delta\Sigma$ modulator and (b) a second-order successive requantizer that implements the state transition matrices in (31).

oscillator (-150 dBc/Hz). The only non-ideality modeled is a 1% mismatch between the charge pump branch currents, which introduces nonlinear distortion.

Fig. 9 shows the simulated power spectrum of the phase error generated using each digital quantizer. As the figure shows, compared to the $\Delta\Sigma$ modulator, the successive requantizer results in significantly lower-power fractional spurious tones in the PLL's phase error: it reduces the largest fractional spurious tone power by 17 dB. The figure also shows that the successive requantizer introduces slightly higher in-band phase noise than the $\Delta\Sigma$ modulator. This is because the low-frequency portion of the nonlinearly distorted quantization noise power of the successive requantizer is slightly higher than that of the $\Delta\Sigma$ modulator. The noise penalty in a real PLL, however, is usually not significant, as $1/f$ noise dominates the phase noise power spectrum at low frequencies. Due to the relatively low bandwidth of the PLL and the third-order loop filter (both of which are typical design choices for analog fractional-N PLLs), the phase noise contributed by the digital quantizer is not dominant at high frequencies, so the higher high-frequency quantization noise from the successive requantizer does not significantly affect the PLL's phase noise performance compared to the $\Delta\Sigma$ modulator. Therefore, in typical fractional-N PLLs, spurious tones can be reduced significantly by using a successive requantizer at the expense of a slight increase in phase noise power.

A digital implementation of the second-order successive requantizer in 65 nm CMOS technology with $K = 16$ quantization blocks has 1500 digital gates, occupies an area of 6000 μm^2, and has an average power consumption of 100 μW. In contrast, a 16-bit input, second-order $\Delta\Sigma$ modulator implementation in the same technology has 120 digital gates, occupies an area of 1000 μm^2, and has an average power consumption of 20 μW. Given that the digital quantizer's area and average power consumption represent a small portion of the total area and average power consumption of typical fractional-N PLLs, the larger area and higher power consumption of the successive requantizer are not significant in practice.

IV. Second-Order Successive Requantizers With Reduced Quantization Noise at Low-Frequencies

The successive requantizer examples in the previous section highlight a design tradeoff between low-frequency quantization noise power and immunity to spurious tones: the successive requantizer that implements the state transition matrices in (31) has lower immunity to spurious tones $(h_t = 3)$ than that which implements the state transition matrices in (23) $(h_t = 5)$, but it has lower quantization noise power at low frequencies. This motivates finding state transition matrices for which $h_t = 1$ to reduce the low-frequency quantization noise power further. As can be verified with the proof of Lemma 3 and Theorem 1 in [13] and from the parallels of first and second-order successive requantizers, if $\mathbf{A_e}$ and $\mathbf{A_o}$ are 3×3 centrosymmetric state transition matrices with five and four non-zero elements, respectively, then $t[n]$ is free of spurious tones. Suppose that $\mathbf{A_e}$ and $\mathbf{A_o}$ satisfy the above condition and that their entries are of the form $k/1024$, where $k \in \{0, 1, \ldots, 1024\}$, as in the previous section. Then, the low-frequency quantization noise can be reduced by choosing the $\mathbf{A_e}$ and $\mathbf{A_o}$ matrices that minimize (29).[6] A computer program was written which cycles through all $\mathbf{A_e}$ and $\mathbf{A_o}$ satisfying the above conditions and picks those which minimize (29), where the limits in (29) are approximated with a suitable number of terms as in the previous section. The resulting state transition matrices are

$$\mathbf{A_e} = \begin{pmatrix} \frac{1}{1024} & 0 & \frac{1023}{1024} \\ 0 & 1 & 0 \\ \frac{1023}{1024} & 0 & \frac{1}{1024} \end{pmatrix}, \quad \mathbf{A_o} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix}. \tag{32}$$

As can be verified from simulation, $t^2[n]$ has spurious tones, so $h_t = 1$.

Fig. 10 shows the simulated power spectrum of $t[n]$ from a 20-bit input second-order successive requantizer which implements the state transition matrices in (32) and of the running sum of the quantization noise of a dithered second-order $\Delta\Sigma$ modulator, $t_{\text{DS}}[n]$. As seen in the figure, the power spectrum of $t[n]$ is lower by approximately 2 dB than that of $t_{\text{DS}}[n]$ at low frequencies. However, the range of $t[n]$ is $(-2, 2)$, which is double to that of $t_{\text{DS}}[n]$. Additionally, as in the previous section,

[6] The same low-frequency quantization noise reduction method from the previous section can be used because, as can be verified, all results derived in the Appendix hold if condition C1 is replaced by the above conditions for $N_u = 1$ $\mathbf{A_e}$ and $\mathbf{A_o}$ state transition matrices.

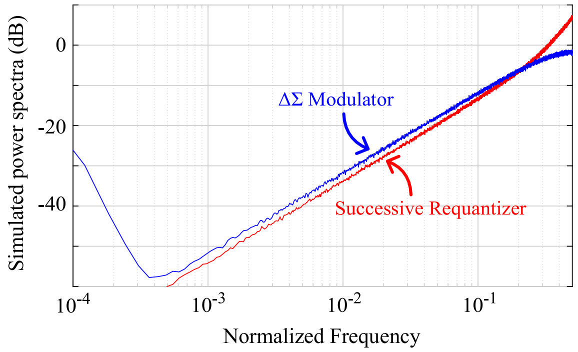


Fig. 10. Simulated power spectra of the running sum of the quantization noise of a second-order successive requantizer that implements the state transition matrices in (32) and a second-order $\Delta\Sigma$ modulator.

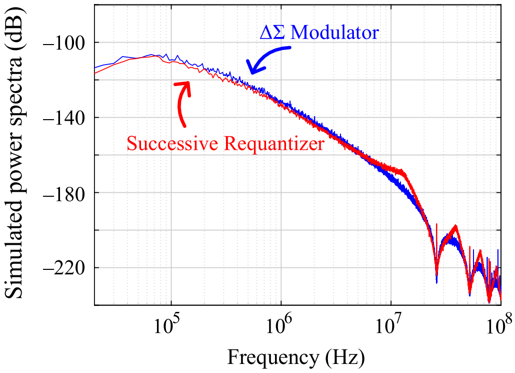


Fig. 11. Simulated power spectra of the phase error of a 3.56 GHz output frequency, 45 kHz bandwidth PLL when its digital quantizer is implemented as a second-order $\Delta\Sigma$ modulator and a second-order successive requantizer that implements the state transition matrices in (32).

$t_{\mathrm{DS}}[n]$ contains an integrated white noise component due to the $\Delta\Sigma$ modulator's dither, whereas $t[n]$ does not.

Fig. 11 shows the simulated power spectrum of the phase error of a PLL when using a second-order $\Delta\Sigma$ modulator and when using the second-order successive requantizer which implements the state transition matrices in (32). The PLL is identical to that described in Section III but with its reference and VCO noise sources turned off to show the quantization noise contribution to the PLL phase error at all frequencies. It also has perfectly-matched charge pump branch currents, as opposed to the modeled PLL in Section III. This allows comparison of the digital quantizers' contribution to the PLL phase noise in the ideal case in which the nonlinearly distorted versions of the quantization noise do not corrupt the low-frequency portion of the PLL phase noise, as in the PLL phase noise PSD shown in Fig. 9(b). As seen in the figure and as predicted by Fig. 10, the PLL phase noise at frequencies below 5 MHz is lower when the successive requantizer is used.

V. Third-Order Successive Requantizers With Reduced Quantization Noise at Low Frequencies

The $s_d[n]$ sequence generator in the successive requantizer can be modified to produce higher-order highpass shaped quantization noise by applying more difference operations to the combinatorial logic block output and by adding logic to ensure $x_d[n]$ and $s_d[n]$ have equal parity for all n. For example, a third-order successive requantizer can be implemented with the $s_d[n]$ sequence generator from Fig. 12. The combinatorial logic block can be described by state transition matrices $\mathbf{A_e}$ and $\mathbf{A_o}$ as before, but since its output is now $v_d[n]$-defined as the triple running sum of $s_d[n]$-instead of $u_d[n]$, where $v_d[n] \in \{N_v, N_v - 1, \ldots, -N_v\}$ for some positive integer N_v, $\mathbf{A_e}$ and $\mathbf{A_o}$ are now defined by

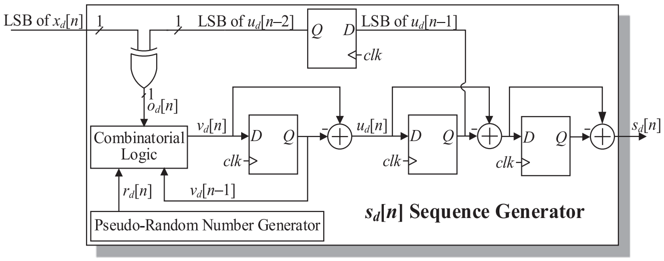


Fig. 12. Block diagram of a third-order $s_d[n]$ sequence generator.

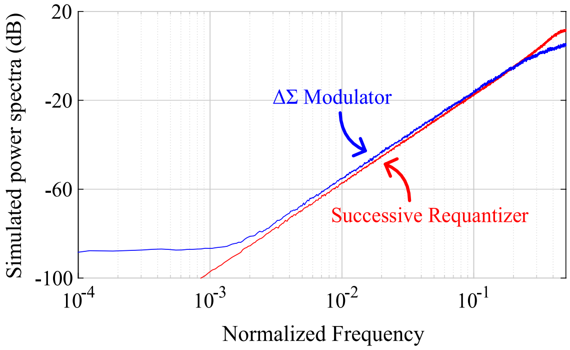


Fig. 13. Simulated power spectra of the running sum of the quantization noise of a third-order successive requantizer that implements the state transition matrices in (32) and a third-order $\Delta\Sigma$ modulator.

$$\begin{aligned} \mathbf{A_e}(i,j) &= \Pr(v_d[n] = \mathbf{v}(j) | v_d[n-1] = \mathbf{v}(i), o_d[n] = 0) \\ \mathbf{A_o}(i,j) &= \Pr(v_d[n] = \mathbf{v}(j) | v_d[n-1] = \mathbf{v}(i), o_d[n] = 1) \end{aligned} \quad (33)$$

where

$$\mathbf{v} = (N_v \quad (N_v - 1) \quad \cdots \quad -N_v)^T \quad (34)$$

in analogy to (20) and (21). Reasoning similar to that from (8) to (12) can be used to verify that, for $x_d[n]$ and $s_d[n]$ to have equal parity, the combinatorial logic block must generate $v_d[n]$ such that $v_d[n] - v_d[n-1]$ and $o_d[n]$ have equal parity, where $o_d[n]$ is generated as in Fig. 12.

Fig. 13 shows the simulated power spectrum of the $t[n]$ of a third-order successive requantizer that implements the state transition matrices in (32) and of the running sum of the quantization noise of a dithered third-order $\Delta\Sigma$ modulator, $t_{\mathrm{DS}}[n]$. The power spectrum of $t[n]$ is lower than that of $t_{\mathrm{DS}}[n]$ by approximately 1 to 2 dB at low frequencies, but the range of $t[n]$ is $(-4, 4)$, which is double that of $t_{\mathrm{DS}}[n]$. Fig. 14(a) shows the simulated power spectrum of the phase error of a PLL when using the third-order $\Delta\Sigma$ modulator and when using the third-order successive requantizer. The PLL modeled is that shown in Fig. 8, where the charge pump (CP) has branch currents of 500 μA, the loop filter has component values of $C_1 = 3.7$ pF, $R_1 = 26$ kΩ, $C_2 = 112$ pF, $R_2 = 26$ kΩ, and $C_3 = 3.7$ pF, the voltage-controlled oscillator (VCO) has a gain

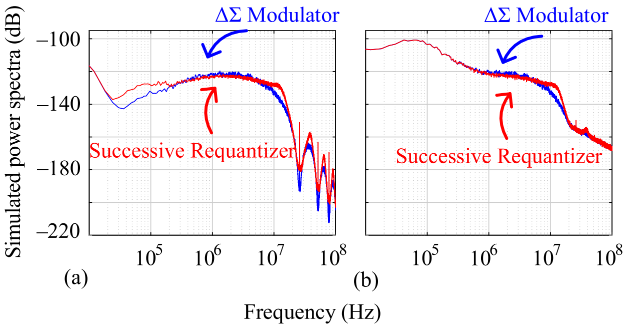


Fig. 14. Simulated power spectra of the phase error of a 3.56 GHz output frequency, 85 kHz bandwidth PLL when its digital quantizer is implemented as a third-order $\Delta\Sigma$ modulator and a third-order successive requantizer that implements the state transition matrices in (32) when (a) there are no circuit noise sources modeled and when (b) reference and VCO noise are modeled.

of 5 MHz/V, the input to the digital quantizers, α, is -2^{-11}, the reference frequency f_{ref} is 26 MHz, the output frequency f_{out} is 3.56 GHz, and the PLL bandwidth is 85 kHz. As seen in the figure, the successive requantizer contributes more low-frequency phase noise than the $\Delta\Sigma$ modulator. This is due to the inherent nonlinear behavior of the PLL and shows that, at low frequencies, the power of nonlinearly distorted versions of the quantization noise running sum of the successive requantizer is higher than that of the $\Delta\Sigma$ modulator. At frequencies from 400 kHz to 4 MHz the phase noise is lower by 0.5 to 1.5 dB when the successive requantizer is used, whereas at higher frequencies the $\Delta\Sigma$ modulator results in lower phase noise. The worse low-frequency behavior of the successive requantizer when used in a PLL is not an issue in practice, as low-frequency phase noise is typically dominated by reference, charge pump, and VCO noise. Fig. 14(b) shows the same phase noise simulation but with $1/f^2$ noise from the VCO (-127 dBc/Hz at 1 MHz) and white noise from the reference oscillator (-150 dBc/Hz). As seen in the figure, when these circuit noise sources are included, the phase noise performance when using either digital quantizer is similar at frequencies below 400 kHz. Therefore, in this PLL using a successive requantizer improves PLL spot noise from 400 kHz to 4 MHz at the detriment of PLL spot noise at higher frequencies.

VI. Conclusion

Second and third-order successive requantizers that can replace the commonly-used $\Delta\Sigma$ modulator in fractional-N PLLs to improve PLL phase error performance have been presented. Specifically, a second-order successive requantizer has been presented which can drastically improve spurious tone performance when used in typical fractional-N PLLs by producing quantization noise that is free of spurious tones even when subjected to the PLL's nonlinear distortion. A phase noise simulation of one such PLL shows a 17 dB power reduction in the largest fractional spurious tone when the digital quantizer is implemented as the second-order successive requantizer instead of as a second-order $\Delta\Sigma$ modulator while maintaining similar phase noise performance. Additionally, second and third-order successive requantizers have been presented which produce lower-power low-frequency quantization noise compared to their $\Delta\Sigma$ modulator counterparts. These successive requantizers are not optimized for nonlinearity-induced spurious tone reduction, but are rather intended for PLLs requiring low phase noise at low or mid frequencies.

Appendix

This Appendix contains the derivation of the expression in (29) used to reduce the low-frequency power spectrum content of $t[n]$. The derivation is based on three assumptions: that condition **C1** from Section III holds, that each $o_d[n]$ takes on the values 0 and 1 with equal probability, and that $\mathbf{A}$ is primitive. The first assumption is a sufficient condition for $t[n]$ to be immune to spurious tones up to at least order one, as explained in Section III. The second assumption is based on the unpredictability of $o_d[n]$ for most quantization blocks. An exact expression for the pmf of $o_d[n]$ conditioned on the $x_0[n]$ and the $r_d[n]$ sequences is hard to find. However, as proven in Section III of [7], each $o_d[n]$ is a deterministic function of $\{x_0[m], m = 0, 1, \ldots, n\}$ and $\{r_k[m], k = 0, 1, \ldots, d-1, m = 0, 1, \ldots, n\}$, which motivates modeling it as being uniformly distributed. Simulations support the validity of this model for most values of d. The third assumption has been empirically found to hold for most matrices $\mathbf{A_e}$ and $\mathbf{A_o}$ satisfying condition **C1**, so it does not significantly limit the utility of (29). An additional assumption in the derivation is that each sequence $x[n]$ in the successive requantizer is such that $x[n] = 0$ for all $n \leq 0$. This assumption is not necessary to obtain (29) but is reasonable and simplifies the derivation.

The power spectrum of $t[n]$ can be estimated with the expected value of the L-length periodogram of $t[n]$

$$I_{t,L}(\omega) = \frac{1}{L}\left|\sum_{n=0}^{L-1} t[n]e^{-j\omega n}\right|^2 \qquad (35)$$

[7], [18]. It follows from (5) and (6) that this expected value can be written as:

$$E\left\{I_{t,L}(\omega)\right\}$$

$$= E\left\{\frac{1}{L}\left|\sum_{n=0}^{L-1}\left(\sum_{d=0}^{K-1} 2^{d-K}(u_d[n]-u_d[n-1])\right)e^{-j\omega n}\right|^2\right\}. \qquad (36)$$

Therefore, $E\{I_{t,L}(\omega)\}$ can be expanded as a finite sum of terms of the form

$$c_{a,b,n,m,L} E\left\{u_a[n]u_b[m]\right\} \qquad (37)$$

where, without loss of generality, $a \geq b$, $a, b \in \{0, 1, \ldots, K-1\}$, $n, m \in \{0, 1, \ldots, L-1\}$, and $c_{a,b,n,m,L}$ is a constant. Using the law of total expectation, the expectation in (37) can be written as

$$E\left\{u_a[n]u_b[m]\right\} = E\left\{u_b[m]E\left\{u_a[n]|u_b[m]\right\}\right\}. \qquad (38)$$

The inner expectation in (38) can be conditioned on additional variables as long as the outer expectation in (38) is computed over all possible values of those variables. Thus, (38) can be rewritten as

$$E\left\{u_a[n]u_b[m]\right\} = E\left\{u_b[m]E\left\{u_a[n]|u_b[m], u_a[0], u_b[0],\right.\right.$$
$$\left.\left. o_a[k], o_b[k]; k=1,2,\ldots,\max\{n,m\}\right\}\right\}. \quad (39)$$

In Section III of [7], it is proven that in a first-order $s_d[n]$ sequence generator $t_d[n]$ conditioned on $t_d[0], o_d[1], o_d[2], \ldots, o_d[n]$ is a deterministic function of $r_d[0], r_d[1], \ldots, r_d[n]$ for each $n > 0$. It follows from the parallels of first and second-order $s_d[n]$ sequence generators explained in Section III of this paper that in a second-order $s_d[n]$ sequence generator $u_d[n]$ conditioned on $u_d[0], o_d[1], o_d[2], \ldots, o_d[n]$ is a deterministic function of $r_d[0], r_d[1], \ldots, r_d[n]$. It follows from this, from the derivation in [7] proving that $o_d[n]$ is a deterministic function of $x_0[n]$ and $\{r_k[m], k=0,1,\ldots,d-1, m=0,1,\ldots,n\}$, and from the independence of the $r_d[n]$ sequences that, for $a \neq b$, (39) can be written as:

$$E\left\{u_a[n]u_b[m]\right\} = E\left\{u_b[m]E\left\{u_a[n]|u_a[0],\right.\right.$$
$$\left.\left. o_a[k]; k=1,2,\ldots,n\right\}\right\}. \quad (40)$$

Lemma 3 and equation (9) in [13] imply that in a first-order $s_d[n]$ sequence generator

$$E\left\{t_a[n]|t_a[0]=0, o_a[k]; k=1,2,\ldots,n\right\} = 0 \quad (41)$$

for each $a \in \{0,1,\ldots,K-1\}$. Therefore, in a second-order $s_d[n]$ sequence generator

$$E\left\{u_a[n]|u_a[0]=0, o_a[k]; k=1,2,\ldots,n\right\} = 0 \quad (42)$$

for each $a \in \{0,1,\ldots,K-1\}$. Since $u_a[0]=0$, it follows that the right side of (40) is zero.[7] Therefore

$$E\left\{u_a[n]u_b[m]\right\} = 0 \text{ when } a \neq b \quad (43)$$

so (36) can be rewritten as

$$E\left\{I_{t,L}(\omega)\right\}$$
$$= \sum_{d=0}^{K-1} 2^{d-K} E\left\{\frac{1}{L}\left|\sum_{n=0}^{L-1}\left(u_d[n]-u_d[n-1]\right)e^{-j\omega n}\right|^2\right\}. \quad (44)$$

This shows that $E\{I_{t,L}(\omega)\}$ can be reduced by reducing the expected value of the L-length periodogram of $t_d[n]$

$$I_{t_d,L}(\omega) = \frac{1}{L}\left|\sum_{n=0}^{L-1}\left(u_d[n]-u_d[n-1]\right)e^{-j\omega n}\right|^2 \quad (45)$$

for each $d \in \{0,1,\ldots,K-1\}$.

Equation (45) can be rewritten as

$$I_{t_d,L}(\omega) = \frac{1}{L}\left|(1-e^{-j\omega})\sum_{n=0}^{L-1}u_d[n]e^{-j\omega n} + u_d[L-1]e^{-j\omega L}\right|^2. \quad (46)$$

Let $I_{u_d,L}(\omega)$ be the L-length periodogram of $u_d[n]$. For any ω such that $I_{u_d,L}(\omega) \neq 0$, (46) can be rewritten as

$$I_{t_d,L}(\omega) = I_{u_d,L}(\omega)\left(|1-e^{-j\omega}|^2 + B(\omega,L)\right), \quad (47)$$

where

$$B(\omega,L)$$
$$= \frac{1}{L}\frac{2u_d[L-1]\mathrm{Re}\left\{(1-e^{-j\omega})e^{j\omega L}\sum_{n=0}^{L-1}u_d[n]e^{-j\omega n}\right\} + u_d^2[L-1]}{I_{u_d,L}(\omega)} \quad (48)$$

Using the proof of Theorem 1 in [7], Lemma 3 and Theorem 1 in [13] prove that in a first-order $s_d[n]$ sequence generator where condition **C1** holds, $I_{t_d,L}(\omega)$ is bounded in probability for all $L \geq 1$ and $0 < |\omega| \leq \pi$. Moreover, the proof of Lemma 3 in [13] shows that the constant C_{t^1} in (29) of the proof of Theorem 1 in [7] equals 0, which implies that $J_{2,2}$ in equation (30) of [7] equals 0 when $p=1$. It follows from this and from the rest of the proof of Theorem 1 in [7] that in a first-order $s_d[n]$ sequence generator $I_{t_d,L}(\omega)$ is uniformly bounded in probability for all $L \geq 1$ and $0 \leq |\omega| \leq \pi$. This implies that in the second-order $s_d[n]$ sequence generator $I_{u_d,L}(\omega)$ is uniformly bounded in probability for all $L \geq 1$ and $0 \leq |\omega| \leq \pi$.

The above implies that

$$\lim_{L\to\infty}\frac{1}{L}\sum_{n=0}^{L-1}u_d[n]e^{-j\omega n} = 0 \quad (49)$$

for all $0 \leq \omega \leq \pi$ almost surely. This and the fact that $u_d[n]$ is bounded for all n imply that

$$\lim_{L\to\infty} B(\omega,L) = 0 \quad (50)$$

almost surely, so it follows from (47) that for large enough values of L and for each $\omega \neq 0$ for which $E\{I_{u_d,L}(\omega)\} \neq 0$, $E\{I_{t_d,L}(\omega)\}$ can be reduced by reducing $E\{I_{u_d,L}(\omega)\}$. Additionally, the continuity of the L-length periodogram and the fact that $I_{u_d,L}(\omega)$ is uniformly bounded for all $L \geq 1$ and all $0 \leq |\omega| \leq \pi$ implies that the low-frequency power spectrum content of $u_d[n]$ can be reduced by reducing $E\{I_{u_d,L}(0)\}$.

It can be shown that

$$I_{u_d,L}(\omega) = \sum_{m=-(L-1)}^{L-1} R_{u_d,L}[m]c_L[m]e^{-j\omega n} \quad (51)$$

[7] If $u_a[0] \neq 0$ it can be shown that the right side of (40) approaches zero as $n \to \infty$, which allows obtaining (29) but requires a longer derivation.

where

$$R_{u_d,L}[m]=\begin{cases}\frac{1}{L-|m|}\sum\limits_{n=0}^{L-1-|m|}u_d[n]u_d\left[n+|m|\right], & |m|\le L-1\\ 0, & \text{otherwise}\end{cases} \tag{52}$$

$$c_L[m]=\begin{cases}\frac{L-|m|}{L}, & |m|\le L-1\\ 0, & \text{otherwise.}\end{cases} \tag{53}$$

By assumption, $o_d[n]$ takes on the values 0 and 1 with equal probability, and as proven in [7], $o_d[n]$ is independent of $r_d[n]$, so the pmf of $u_d[n]$, as described by $\mathbf{A_e}$ and $\mathbf{A_o}$ in (20), depends only on $u_d[n-1]$ and not on n. This implies $u_d[n]$ is stationary. Therefore, it follows from (52) that:

$$E\left\{R_{u_d,L}[m]\right\}=R_{u_d}[m] \tag{54}$$

where

$$R_{u_d}[m]=E\left\{u_d[n]u_d[n+m]\right\} \tag{55}$$

is the autocorrelation of $u_d[n]$. It follows from the proof of Theorem 1 in [13] and the parallels between the first and second-order $s_d[n]$ sequence generators that $R_{u_d}[m]$ decreases exponentially with $|m|$. This, with (51), (53), and (54), implies that

$$\begin{aligned}\lim_{L\to\infty}E\left\{I_{u_d,L}(0)\right\}&=\lim_{L\to\infty}\sum_{m=-(L-1)}^{L-1}R_{u_d}[m]c_L[m]\\&=\lim_{L\to\infty}\sum_{m=-(L-1)}^{L-1}R_{u_d}[m].\end{aligned} \tag{56}$$

As proven in [7], the sequence whose pmf is described by $\mathbf{A_e}$ and $\mathbf{A_o}$ is a Markov process, and $\mathbf{A_e}$ and $\mathbf{A_o}$ are its state transition matrices conditioned on the value of $o_d[n]$ at each n. In the second-order successive requantizer, this Markov process is $u_d[n]$, as shown in (20). By assumption, $o_d[n]$ takes on the values 0 and 1 with equal probability, so the state transition matrix of $u_d[n]$ is given by $\mathbf{A}$ as defined in (30). This implies that

$$\mathbf{A}(N_u+1-i,N_u+1-u)=\Pr\left(u_d[n]=u|u_d[u-1]=i\right) \tag{57}$$

for all $i,u\in\{-N_u,\ldots,N_u\}$. Because $\mathbf{A}$ is primitive

$$\lim_{k\to\infty}\mathbf{A}^k \tag{58}$$

exists and

$$\left(\lim_{k\to\infty}\mathbf{A}^k\right)(N_u+1-i,N_u+1-u)=\Pr\left(u_d[n]=u\right) \tag{59}$$

for all $i,u\in\{-N_u,\ldots,N_u\}$ [19] (the left side of (59) should be interpreted as the (N_u+1-i)th row, (N_u+1-u)th column entry of the matrix inside the first parentheses). Additionally, using identical reasoning to that in [13] to derive equation (94) in that paper, it follows from (21) and (57) that:

$$\left(\mathbf{A}^{|m|}\mathbf{u}\right)(N_u+1-u)=E\left\{u_d[n+m]|u_d[n]=u\right\} \tag{60}$$

for all $u\in\{-N_u,\ldots,N_u\}$.

Using (55) and the law of total expectation, $R_{u_d}[m]$ can be evaluated as

$$\begin{aligned}R_{u_d}[m]&=\sum_{u=-N_u}^{N_u}E\{u_d[n]u_d[n+m]|u_d[n]=u\}\cdot\Pr(u_d[n]=u)\\&=\sum_{u=-N_u}^{N_u}E\{u_d[n+m]|u_d[n]=u\}\cdot u\cdot\Pr(u_d[n]=u)\,.\end{aligned} \tag{61}$$

With (21), (59), and (60), (61) can be rewritten as

$$\begin{aligned}R_{u_d}[m]&=\sum_{u=-N_u}^{N_u}\left(\mathbf{A}^{|m|}\mathbf{u}\right)(N_u+1-u)\cdot\mathbf{u}(N_u+1-u)\\&\quad\cdot\left(\lim_{k\to\infty}\mathbf{A}^k\right)(1,N_u+1-u)\\&=\sum_{i=1}^{2N_u+1}\left(\mathbf{A}^{|m|}\mathbf{u}\right)(i)\cdot\mathbf{u}(i)\cdot\left(\lim_{k\to\infty}\mathbf{A}^k\right)(1,i).\end{aligned} \tag{62}$$

With (56), this implies that

$$\begin{aligned}\lim_{L\to\infty}E\left\{I_{u_d,L}(0)\right\}=\sum_{m=-\infty}^{\infty}\sum_{i=1}^{2N_u+1}\left(\mathbf{A}^{|m|}\mathbf{u}\right)(i)\\\cdot\mathbf{u}(i)\cdot\left(\lim_{k\to\infty}\mathbf{A}^k\right)(1,i)\end{aligned} \tag{63}$$

which is the expression in (29).

Acknowledgment

The authors would like to thank Professor Peter Kennedy for his helpful comments relating to this work.

References

[1] B. Miller and B. Conley, "A multiple modulator fractional divider," in *Proc. Annu. IEEE Symp. Freq. Control*, Mar. 1990, vol. 44, pp. 559–568.

[2] B. Miller and B. Conley, "A multiple modulator fractional divider," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 3, pp. 578–583, Jun. 1991.

[3] T. A. Riley, M. A. Copeland, and T. A. Kwasniewski, "Delta-Sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553–559, May 1993.

[4] B. Razavi, *Phase-Locking in High-Performance Systems: From Devices to Architectures*. New York, NY, USA: Wiley-Interscience, 2003.

[5] B. De Muer and M. Steyaert, "A CMOS monolithic $\Delta\Sigma$-controlled fractional-N frequency synthesizer for DCS-1800," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 835 844, Jul. 2002.

[6] S. Pamarti, L. Jansson, and I. Galton, "A wideband 2.4 GHz $\Delta\Sigma$ fractional-N PLL with 1 Mb/s in-loop modulation," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 49–62, Jan. 2004.

[7] A. Swaminathan, A. Panigada, E. Masry, and I. Galton, "A digital requantizer with shaped requantization noise that remains well behaved after nonlinear distortion," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5382–5394, Nov. 2007.

[8] K. J. Wang, A. Swaminathan, and I. Galton, "Spurious tone suppression techniques applied to a wide-bandwidth 2.4 GHz fractional-N PLL," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2787–2797, Dec. 2008.

[9] H. Jian, Z. Xu, Y. Wu, and F. Chang, "A compact 0.8–6 GHz fractional-N PLL with binary-weighted D/A differentiator and offset-frequency $\Delta-\Sigma$ modulator for noise and spurs cancellation," in *Proc. Symp. VLSI Circuits*, Jun. 16–18, 2009, pp. 186–187.

[10] P. Su and S. Pamarti, "Mismatch shaping techniques to linearize charge pump errors in fractional-N PLLs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 1221–1230, Jun. 2010.

[11] K. Hosseini, B. Fitzgibbon, and M. P. Kennedy, "Observations concerning the generation of spurious tones in digital Delta-Sigma modulators followed by a memoryless nonlinearity," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 11, pp. 714–718, Nov. 2011.

[12] E. Familier and I. Galton, "A fundamental limitation of DC-free quantization noise with respect to nonlinearity-induced spurious tones," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4172–4180, Aug. 2013.

[13] E. Familier, C. Venerus, and I. Galton, "A class of quantizers with DC-free quantization noise and optimal immunity to nonlinearity-induced spurious tones," *IEEE Trans. Signal Process.*, vol. 61, no. 17, Sep. 2013.

[14] Z. Li, H. Mo, and M. P. Kennedy, "Comparative spur performance of a fractional-N frequency synthesizer with a nested MASH-SQ3 divider controller in the presence of memoryless piecewise-linear and polynomial nonlinearities," in *Proc. ISSC/CIICT 2014*, Jun. 26–27, 2014, pp. 374–379.

[15] M. P. Kennedy, H. Mo, Z. Li, G. Hu, P. Scognamiglio, and E. Napoli, "The noise and spur delusion in fractional-N frequency synthesizer design," in *Proc. ISCAS 2015*, May 24–27, 2015, pp. 2577–2580.

[16] E. Fogleman, I. Galton, W. Huff, and H. Jensen, "A 3.3-V single-poly CMOS audio ADC delta-sigma modulator with 98-dB peak SINAD and 105-dB peak SFDR," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, Mar. 2000.

[17] P. Perkins, "A theorem on regular matrices," *Pac. J. Math.*, vol. 11, no. 4, pp. 1529–1533, 1961.

[18] A. V. Oppenheim, R. W. Schafer, and J. R. Buck,*Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[19] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.

Eythan Familier (M'16) received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2010, and the M.S. degree in electrical engineering from the University of California at San Diego, La Jolla, CA, USA, in 2012. He is currently working toward the Ph.D. degree at the University of California at San Diego. For the last years he has been part of the Integrated Signal Processing Group, where he has worked on spurious tone mitigation techniques in phase-locked loops.

Ian Galton (F'09) received the Sc.B. degree from Brown University, Providence, RI, USA, in 1984, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, USA, in 1989 and 1992, respectively, all in electrical engineering.

Since 1996, he has been a Professor of Electrical Engineering at the University of California at San Diego, La Jolla, CA, USA, where he teaches and conducts research in the field of mixed-signal integrated circuits and systems for communications. Prior to 1996, he was with the University of California at Irvine, and prior to 1989, he was with Acuson and Mead Data Central. His research involves the invention, analysis, and integrated circuit implementation of critical communication system blocks such as data converters, frequency synthesizers, and clock recovery systems. In addition to his academic research, he regularly consults at several semiconductor companies and teaches industry-oriented short courses on the design of mixed-signal integrated circuits.

Dr. Galton has served on a corporate Board of Directors, on several corporate Technical Advisory Boards, as the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING, as a member of the IEEE Solid-State Circuits Society Administrative Committee, as a member of the IEEE Circuits and Systems Society Board of Governors, as a member of the IEEE International Solid-State Circuits Conference Technical Program Committee, and as a member of the IEEE Solid-State Circuits Society Distinguished Lecturer Program.