

# Delta-Sigma Fractional- $N$ Phase-Locked Loops

Ian Galton

**Abstract**—This paper presents a tutorial on delta-sigma fractional- $N$  PLLs for frequency synthesis. The presentation assumes the reader has a working knowledge of integer- $N$  PLLs. It builds on this knowledge by introducing the additional concepts required to understand  $\Delta\Sigma$  fractional- $N$  PLLs. After explaining the limitations of integer- $N$  PLLs with respect to tuning resolution, the paper introduces the delta-sigma fractional- $N$  PLL as a means of avoiding these limitations. It then presents a self-contained explanation of the relevant aspects of delta-sigma modulation, an extension of the well known integer- $N$  PLL linearized model to delta-sigma fractional- $N$  PLLs, a design example, and techniques for wideband digital modulation of the VCO within a delta-sigma fractional- $N$  PLL.

## I. INTRODUCTION

Over the last decade, delta-sigma ( $\Delta\Sigma$ ) fractional- $N$  phase locked loops (PLLs) have become widely used for frequency synthesis in consumer-oriented electronic communications products such as cellular phones and wireless LANs. Unlike an integer- $N$  PLL, the output frequency of a  $\Delta\Sigma$  fractional- $N$  PLL is not limited to integer multiples of a reference frequency. The core of a  $\Delta\Sigma$  fractional- $N$  PLL is similar to an integer- $N$  PLL, but it incorporates additional digital circuitry that allows it to accurately interpolate between integer multiples of the reference frequency. The tuning resolution depends only on the complexity of the digital circuitry, so considerable flexibility and programmability is achieved. A single  $\Delta\Sigma$  fractional- $N$  PLL often can be used for local oscillator generation in applications that would otherwise require a cascade of two or more integer- $N$  PLLs. Moreover, the fine tuning resolution makes it possible to perform digitally-controlled frequency modulation for generation of continuous-phase (e.g., FSK and MSK) transmit signals, thereby simplifying wireless transmitters. These benefits come at the expense of increased digital complexity and somewhat increased phase noise relative to integer- $N$  PLLs. However, with the relentless progress in silicon VLSI technology optimized for digital circuitry, this tradeoff is increasingly attractive, especially in consumer products which tend to favor cost reduction over performance.

This paper presents a tutorial on  $\Delta\Sigma$  fractional- $N$  PLLs. It is assumed that the reader has a working knowledge of integer- $N$  PLLs. The paper builds on this knowledge by presenting the additional concepts required to understand  $\Delta\Sigma$  fractional- $N$  PLLs. The limitations of integer- $N$  PLLs with respect to tuning resolution are described in Section II. The key ideas underlying fractional- $N$  PLLs in general and  $\Delta\Sigma$  fractional- $N$

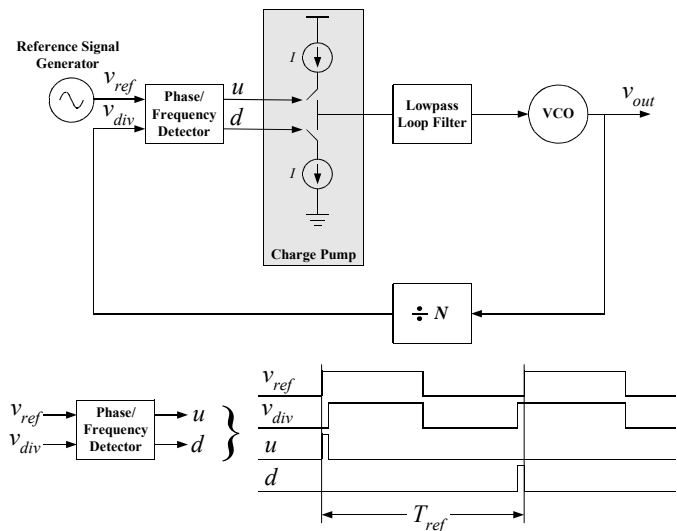


Figure 1: A typical integer- $N$  PLL.

PLLs in particular are presented in Section III. The primary innovation in  $\Delta\Sigma$  fractional- $N$  PLLs relative to other types of fractional- $N$  PLLs is the use of  $\Delta\Sigma$  modulation. Therefore, a self-contained introduction to  $\Delta\Sigma$  modulation as it relates to  $\Delta\Sigma$  fractional- $N$  PLLs is presented in Section IV. A  $\Delta\Sigma$  fractional- $N$  PLL linearized model is derived in Section V and compared to the corresponding model for integer- $N$  PLLs. A design example is presented to demonstrate how the model is used in practice. Design issues that arise in  $\Delta\Sigma$  fractional- $N$  PLLs but not integer- $N$  PLLs are presented in Section VI, and recently developed enhancements to  $\Delta\Sigma$  fractional- $N$  PLLs that allow wideband digital modulation of the VCO are presented in Section VII.

## II. INTEGER- $N$ PLL LIMITATIONS

An example of a typical integer- $N$  PLL for frequency synthesis is shown in Figure 1 [1], [2]. Its purpose is to generate a spectrally pure periodic output signal with a frequency of  $N f_{ref}$ , where  $N$  is an integer, and  $f_{ref}$  is the frequency of the reference signal. The example PLL consists of a phase-frequency detector (PFD), a charge pump, a lowpass loop filter, a voltage controlled oscillator (VCO), and an  $N$ -fold digital divider. The PFD compares the positive-going edges of the reference signal to those from the divider and causes the charge pump to drive the loop filter with current pulses whose widths are proportional to the phase difference between the two signals. The pulses are lowpass filtered by the loop filter and the resulting waveform drives the VCO. Within the loop bandwidth phase noise from the VCO is suppressed and outside the loop bandwidth most of the other noise sources are suppressed, so the PLL can be designed to generate a spectrally pure output sig-

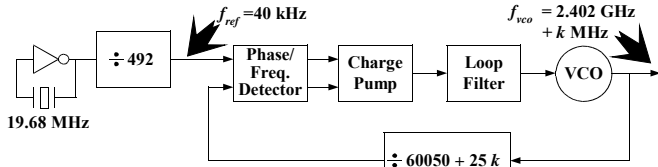


Figure 2: An example integer- $N$  PLL for generation of the Bluetooth wireless LAN RF channel frequencies.

nal at any integer multiple of the reference frequency,  $f_{ref}$ .

As indicated by the timing diagram in Figure 1, the loop filter is updated by the charge pump once every reference period. This discrete-time behavior places an upper limit on the loop bandwidth of approximately  $f_{ref}/10$  above which the PLL tends to be unstable [1]. In integrated circuit PLLs, it is common to further limit the bandwidth to approximately  $f_{ref}/20$  to allow for process and temperature variations.

The output frequency can be changed by changing  $N$ , but  $N$  must be an integer, so the output frequency can be changed only by integer multiples of the reference frequency. If finer tuning resolution is required the only option is to reduce the reference frequency. Unfortunately, this tends to reduce the maximum practical loop bandwidth, thereby increasing the settling time of the PLL, the noise contributed by the VCO, and the in-band portions of the noise contributed by the reference source, the PFD, the charge pump, and the divider.

This fundamental tradeoff between bandwidth and tuning resolution in integer- $N$  PLLs creates problems in many applications. For example, a PLL that can be tuned from 2.402 GHz to 2.480 GHz in steps of 1 MHz is required to generate the local oscillator signal in a direct conversion Bluetooth transceiver [3]. An integer- $N$  PLL capable of generating the local oscillator signal from a commonly used crystal oscillator frequency, 19.68 MHz, is shown in Figure 2. A reference frequency of  $f_{ref} = 40$  kHz—the greatest common divisor of the crystal frequency and the set of desired output frequencies—is obtained by dividing the crystal oscillator signal by 492. The resulting PLL output frequency is  $60050 + 25k$  times the reference frequency, where  $k$  is an integer used to select the desired frequency step.

The PLL achieves the desired output frequencies, but its bandwidth is limited to approximately 2 kHz, i.e.,  $f_{ref}/20$ . Unfortunately, with such a low bandwidth the settling time exceeds the 200  $\mu$ s limit specified in the Bluetooth standard, and the phase noise contributed by the VCO would be unacceptably high if it were implemented in present-day CMOS technology. One solution is to use a 1 MHz reference signal, but this requires the crystal frequency to be an integer multiple of 1 MHz, or another PLL to generate a 1 MHz reference frequency. Unfortunately, in low cost consumer electronics applications such as Bluetooth, it is often desirable to be compatible with all of the popular crystal frequencies, so restricting the crystal frequencies to multiples of 1 MHz is not always an option. In such cases, an additional PLL capable of generating the 1 MHz reference signal with very little phase noise from any of the crystal frequencies is required, or, as described in the next section, a single fractional- $N$  PLL can be used.

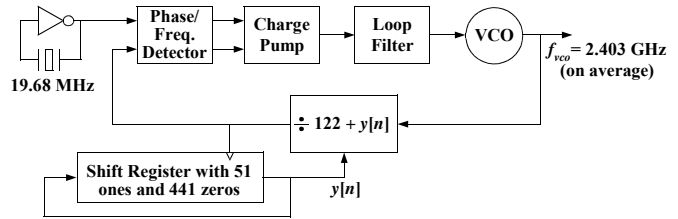


Figure 3: A fractional- $N$  PLL that generates non-integer multiples of the reference frequency, but has phase noise consisting of large spurious tones.

### III. THE IDEA BEHIND $\Delta\Sigma$ FRACTIONAL- $N$ PLLS

In this section, the example problem of generating the second Bluetooth channel frequency, 2.403 GHz, with a reference frequency of 19.68 MHz is used as a vehicle with which to explain the idea behind  $\Delta\Sigma$  fractional- $N$  PLLs. First, a pair of “bad” fractional- $N$  PLLs are presented that achieve the desired frequency but have poor phase noise performance. Then the  $\Delta\Sigma$  fractional- $N$  PLL technique is presented as a means of improving the phase noise performance.

The output frequency of an integer- $N$  PLL with a reference frequency of 19.68 MHz is 2.40096 GHz when the divider modulus,  $N$ , is set to 122 and 2.42064 GHz when  $N$  is set to 123. The problem is that to achieve the desired frequency of 2.403 GHz,  $N$  would have to be set to the non-integer value of  $122 + 51/492$ . This cannot be implemented directly because the divider modulus must be an integer value. However the divider modulus can be updated each reference period, so one option is to switch between  $N = 122$  and  $N = 123$  such that the average modulus over many reference periods converges to  $122 + 51/492$ . In this case, the resulting average PLL output frequency is 2.403 GHz as desired. This is the fundamental idea behind most fractional- $N$  PLLs [4].

While dynamically switching the divider modulus solves the problem of achieving non-integer multiples of the reference frequency, a price is paid in the form of increased phase noise. During each reference period the difference between the actual divider modulus and the average, i.e., ideal, divider modulus represents error that gets injected into the PLL and results in increased phase noise. As described below, the amount by which the phase noise is increased depends upon the characteristics of the sequence of divider moduli.

For example, in the fractional- $N$  PLL shown in Figure 3, the divider modulus is set each reference period to 122 or 123 such that over each set of 492 consecutive reference periods it is set to 122 a total of 441 times and 123 a total of 51 times. Thus, the average modulus is  $122 + 51/492$  as required. The sequence of moduli is periodic with a period of 492, so it repeats at a rate of 40 kHz. Consequently, the difference between the actual divider moduli and their average is a periodic sequence with a repeat rate of 40 kHz, so the resulting phase noise is periodic and is comprised of spurious tones at integer multiples of 40 kHz. Many of the spurious tones occur at low frequencies, and they can be very large. Unfortunately, the only way to suppress the tones is have a very small PLL bandwidth, which negates the potential benefit of the fraction-

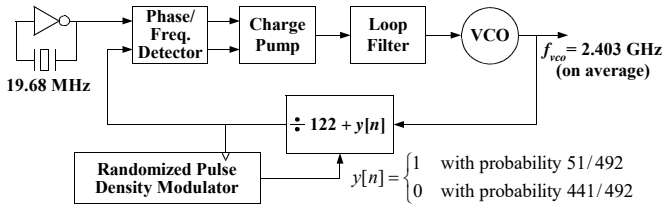


Figure 4: A fractional- $N$  PLL that generates non-integer multiples of the reference frequency, but has a large amount of in-band phase noise.

al- $N$  technique.

One way to eliminate spurious tones is to introduce randomness to break up the periodicity in the sequence of moduli while still achieving the desired average modulus. For example, as shown in Figure 4, a digital block can be used to generate a sequence,  $y[n]$ , that approximates a sampled sequence of independent random variables that take on values of 0 and 1 with probabilities 441/492 and 51/492, respectively. During the  $n^{\text{th}}$  reference period the divider modulus is set to  $122 + y[n]$ , so the sequence of moduli has the desired average yet its power spectral density (PSD) is that of white noise. Thus, instead of contributing spurious tones, the modified technique introduces white noise. Unfortunately, the portion of the white noise within the PLL's bandwidth is integrated by the PLL transfer function, so the overall phase noise contribution again can be significant unless the PLL bandwidth is small.

In each fractional- $N$  PLL example presented above, the sequence,  $y[n]$ , can be written as  $y[n] = x + e_m[n]$ , where  $x$  is the desired fractional part of the modulus, i.e.,  $x = 51/492$ , and  $e_m[n]$  is undesired zero-mean *quantization noise* caused by using integer moduli in place of the ideal fractional value. In the first example,  $e_m[n]$  is periodic and therefore consists of spurious tones at multiples of 40 kHz. In the second example,  $e_m[n]$  is white noise. Each PLL attenuates the portion of  $e_m[n]$  outside its bandwidth, but the portion within its bandwidth is not significantly attenuated. Unfortunately, in each example  $e_m[n]$  contains significant power at low frequencies, so it contributes substantial phase noise unless the PLL bandwidth is very low.

A  $\Delta\Sigma$  fractional- $N$  PLL avoids this problem by generating the sequence of moduli such that the quantization noise has most of its power in a frequency band well above the desired bandwidth of the PLL [5], [6], [7]. An example  $\Delta\Sigma$  fractional- $N$  PLL is shown in Figure 5. The PLL core is similar to those of the previous fractional- $N$  PLL examples, but in this case  $y[n]$  is generated by a digital  $\Delta\Sigma$  modulator. The details of how the  $\Delta\Sigma$  modulator works are presented in the next section, but its purpose is to coarsely quantize its input sequence,  $x[n]$ , such that  $y[n]$  is integer-valued and has the form:  $y[n] = x[n - 2] + e_m[n]$ , where  $e_m[n]$  is dc-free quantization noise with most of its power outside the PLL bandwidth. In this example,  $x[n]$  consists of the desired fractional modulus value, 51/492, plus a small, pseudo-random, 1-bit sequence. As described in the next section, the pseudo-random sequence is necessary to avoid spurious tones in the  $\Delta\Sigma$  modulator's quantization noise, but its amplitude is very small so it does not appreciably increase the phase noise of the PLL.

Also shown in Figure 5 are PSD plots of the output phase

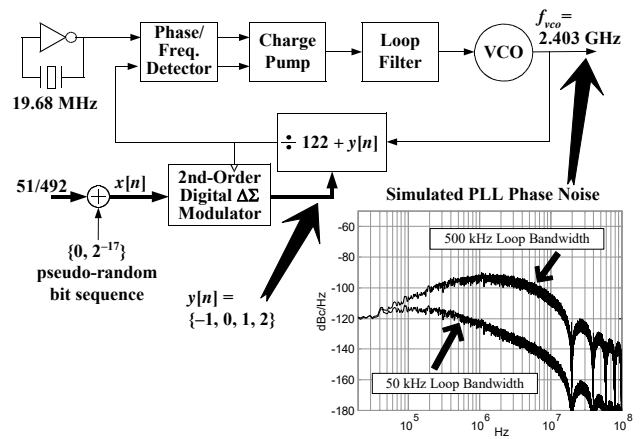


Figure 5: A  $\Delta\Sigma$  fractional- $N$  PLL example.

noise arising from  $\Delta\Sigma$  modulator quantization noise,  $e_m[n]$ , in two computer simulated versions of the example  $\Delta\Sigma$  fractional- $N$  PLL, one with a 50 kHz loop bandwidth and the other with a 500 kHz loop bandwidth. As shown in the next section, the PSD of  $e_m[n]$  increases with frequency, so the phase noise PSD corresponding to the 50 kHz bandwidth PLL is significantly smaller than that corresponding to the 500 kHz bandwidth PLL. For example, the former easily meets the requirements for a local oscillator in a direct conversion Bluetooth transceiver, but the latter falls short of the requirements by at least 23 dB.

#### IV. DELTA-SIGMA MODULATION OVERVIEW

As mentioned above, a digital  $\Delta\Sigma$  modulator performs coarse quantization in such a way that the inevitable error introduced by the quantization process, i.e., the quantization noise, is attenuated in a specific frequency band of interest. There are many different  $\Delta\Sigma$  modulator architectures. Most use coarse uniform quantizers to perform the quantization with feedback around the quantizers to suppress the quantization noise in particular frequency bands. Therefore, to illustrate the  $\Delta\Sigma$  modulator concept, first a specific uniform quantizer example is considered in isolation, and then a specific  $\Delta\Sigma$  modulator architecture that incorporates the uniform quantizer is presented.

##### A. An Example Uniform Quantizer

The input-output characteristic of the example uniform quantizer is shown in Figure 6. It is a 9-level quantizer with integer valued output levels. For each input value with a magnitude less than 4.5, the quantizer generates the corresponding output sample by rounding the input value to the nearest integer. For each input value greater than 4.5 or less than  $-4.5$ , the quantizer sets its output to 4 or  $-4$ , respectively; such values are said to *overload* the quantizer. By defining the quantization noise as  $e_q[n] = y[n] - r[n]$ , the quantizer can be viewed without approximation as an additive noise source as illustrated in the figure.

To illustrate some properties of the example quantizer, consider a 48 Msample/s input sequence,  $x[n]$ , consisting of a

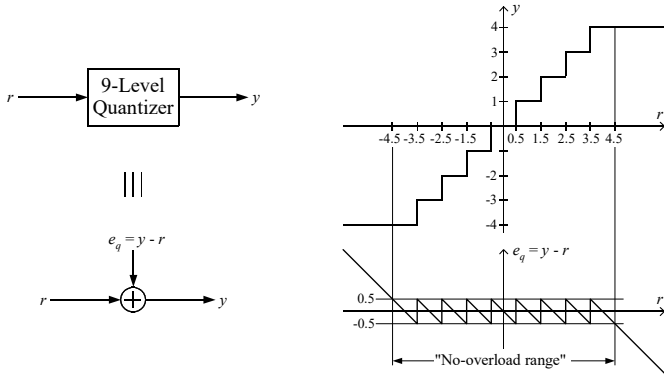


Figure 6: A 9-level quantizer example.

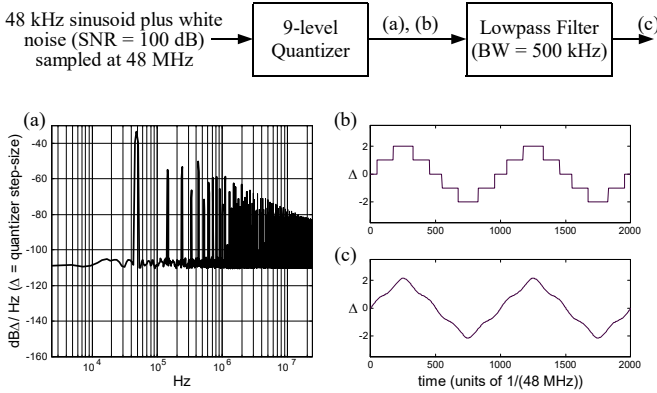


Figure 7: (a) A power spectral density plot of the quantizer output in dB, relative to the quantization step-size of  $\Delta = 1$ , per Hz, (b) a time domain plot of the quantizer output, and (c) a time domain plot of the quantizer output filtered by a sharp lowpass filter with a cutoff frequency of 500 kHz.

48 kHz sinusoid with an amplitude of 1.7 plus a small amount of white noise such that the input signal-to-noise ratio (SNR) is 100 dB. Figure 7(a) shows the PSD plot of the resulting quantizer output sequence, and Figure 7(b) shows a time domain plot of the quantizer output sequence over two periods of the sinusoid. Given the coarseness of the quantization, it is not surprising that the quantizer output sequence is not a precise representation of the quantizer input sequence. As evident in Figure 7(a), the quantization noise for this input sequence consists primarily of harmonic distortion as represented by the numerous spurious tones distributed over the entire discrete-time frequency band. Even in the relatively narrow frequency band below 500 kHz, significant harmonic distortion corrupts the desired signal. To illustrate this in the time domain, Figure 7(c) shows the sequence obtained by passing the quantizer output sequence through a sharp lowpass discrete-time filter with a cutoff frequency of 500 kHz. The significant quantization noise power in the zero to 500 kHz frequency band causes the sequence shown in Figure 7(c) to deviate significantly from the sinusoidal quantizer input sequence.

### B. An Example $\Delta\Sigma$ Modulator

The example  $\Delta\Sigma$  modulator architecture shown in Figure 8 can be used to circumvent this problem. The structure incorporates the same 9-level quantizer presented above, but in this

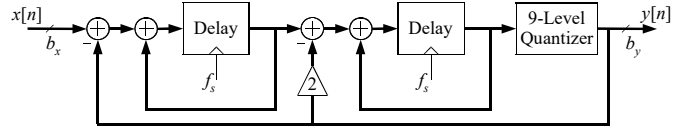


Figure 8: A  $\Delta\Sigma$  modulator example.

case the quantizer is preceded by two delaying discrete-time integrators (i.e., accumulators), and surrounded by two feedback loops [8], [9]. Each discrete-time integrator has a transfer function of  $z^{-1}/(1-z^{-1})$  which implies that its  $n^{\text{th}}$  output sample is the sum of all its input samples for times  $k < n$ . With the quantizer represented as an additive noise source as depicted in Figure 6, the  $\Delta\Sigma$  modulator can be viewed as a two-input, single-output, linear time-invariant, discrete-time system. It is straightforward to verify that

$$y[n] = x[n-2] + e_m[n], \quad (1)$$

where  $e_m[n]$  is the overall quantization noise of the  $\Delta\Sigma$  modulator and is given by

$$e_m[n] = e_q[n] - 2e_q[n-1] + e_q[n-2]. \quad (2)$$

To illustrate the behavior of the  $\Delta\Sigma$  modulator, suppose that the same 48 Msample/s input sequence considered above is applied to the input of the  $\Delta\Sigma$  modulator, and that the discrete-time integrators in the  $\Delta\Sigma$  modulator are clocked at 48 MHz. Figure 9(a) shows the PSD plot of the resulting  $\Delta\Sigma$  modulator output sequence,  $y[n]$ , and Figure 9(b) shows a time domain plot of  $y[n]$  over two periods of the sinusoid. Two important differences with respect to the uniform quantization example shown in Figure 7 are apparent: the quantization noise PSD is significantly attenuated at low frequencies, and no spurious tones are visible anywhere in the discrete-time spectrum. For instance, the SNR in the zero to 500 kHz frequency band is approximately 84 dB for this example as opposed to 14 dB for the uniform quantization example of Figure 7. Consequently, subjecting the  $\Delta\Sigma$  modulator output sequence to a lowpass filter with a cutoff frequency of 500 kHz results in a sequence that is very nearly equal to the  $\Delta\Sigma$  modulator input sequence as demonstrated in Figure 9(c).

Below about 120 kHz, the PSD shown in Figure 9(a) is dominated by the two components of the  $\Delta\Sigma$  modulator input sequence: the 48 kHz sinusoid component, and the input noise component. Above 120 kHz, the PSD is dominated by the  $\Delta\Sigma$  modulator quantization noise,  $e_m[n]$ , and rises with a slope of 40 dB per decade. It follows from (2) that  $e_m[n]$  can be viewed as the result of passing the additive noise from the quantizer,  $e_q[n]$ , through a discrete-time filter with transfer function  $(1-z^{-1})^2$ . Since this filter has two zeros at dc, the smooth 40 dB per decade increase of the PSD of  $e_m[n]$  indicates that  $e_q[n]$  is very nearly white noise, at least for the example shown in Figure 9.

It can be proven that  $e_q[n]$  is indeed white noise; it has a variance of 1/12 and is uncorrelated with the  $\Delta\Sigma$  modulator input sequence [10]. Moreover, this situation holds in general for the example  $\Delta\Sigma$  modulator architecture provided that the input sequence satisfies two conditions: 1) its magnitude is sufficiently small that the quantizer within the  $\Delta\Sigma$  modulator

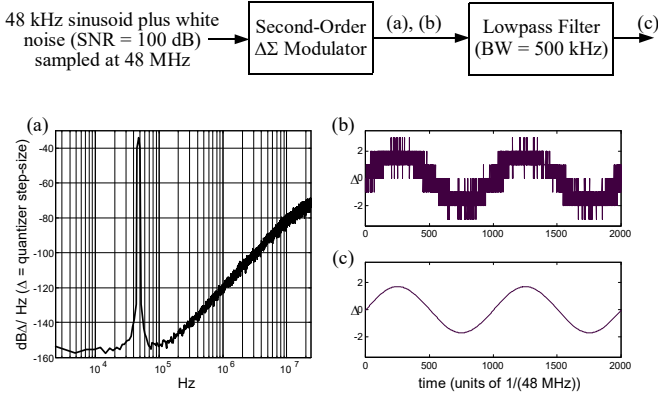


Figure 9: (a) A power spectral density plot of the  $\Delta\Sigma$  modulator output in dB, relative to the quantization step-size of  $\Delta = 1$ , per Hz, (b) a time domain plot of the  $\Delta\Sigma$  modulator output, and (c) a time domain plot of the  $\Delta\Sigma$  modulator output filtered by a sharp lowpass filter with a cutoff frequency of 500 kHz.

never overloads, and 2) it consists of a signal component plus a small amount of independent white noise. It can be shown that the first condition is satisfied if the input signal is bounded in magnitude by  $3\Delta$  where  $\Delta$  is the step-size of the quantizer (for this example,  $\Delta = 1$ ) [11]. Input sequences with values even slightly exceeding  $3\Delta$  in magnitude generally cause the quantizer to overload with the result that  $e_q[n]$  contains spurious tones and the SNR in the frequency band of interest is degraded. For this reason, the range between  $-3\Delta$  and  $3\Delta$  is said to be the *input no-overload range* of the  $\Delta\Sigma$  modulator. For the second condition to be satisfied, the power of the  $\Delta\Sigma$  modulator input sequence's white noise component may be arbitrarily small, but if it is absent altogether,  $e_q[n]$ , is not guaranteed to be white. For instance, in the example shown in Figure 9 the input sequence contains a white noise component with 100dB less power than the signal component. If this tiny noise component were not present, the resulting  $\Delta\Sigma$  modulator output PSD would contain numerous spurious tones. Since the  $\Delta\Sigma$  modulators used in  $\Delta\Sigma$  fractional- $N$  PLLs are all-digital devices, the noise must be added digitally. As shown in [12], it is sufficient to add a 1-bit, sub-LSB, independent, white noise *dither sequence* with zero mean at the input node. In practice, a 1-bit pseudo-random dither sequence is typically used in place of a truly random dither sequence. Such a sequence can be generated easily using a linear feedback shift register, and has the desired result with respect to the quantization noise despite not being truly random [13], [14].

### C. Other $\Delta\Sigma$ Modulator Options

To this point, the  $\Delta\Sigma$  modulation concept has been illustrated via the particular example  $\Delta\Sigma$  modulator architecture shown in Figure 8, namely a second-order multi-bit  $\Delta\Sigma$  modulator. While this type of  $\Delta\Sigma$  modulator is widely used in  $\Delta\Sigma$  fractional- $N$  PLLs, there exist other types of  $\Delta\Sigma$  modulators that can be applied to  $\Delta\Sigma$  fractional- $N$  PLLs. Most of the other architectures are higher-order  $\Delta\Sigma$  modulators that perform higher than second-order quantization noise shaping, thereby more aggressively suppressing quantization noise in particular frequency bands relative to the example second-order  $\Delta\Sigma$  modulator. Some of these higher-order  $\Delta\Sigma$  modulators incor-

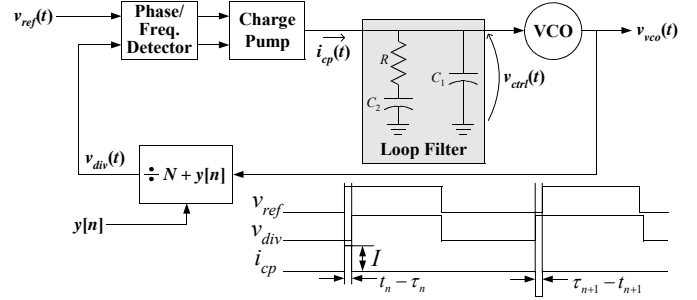


Figure 10: The  $\Delta\Sigma$  fractional- $N$  PLL with the details of a commonly used loop filter and a timing diagram relating to the charge pump output.

porate a higher than second-order loop filter (e.g., more than two discrete-time integrators) and a single quantizer surrounded by one or more feedback loops [15], [16]. In many cases, these  $\Delta\Sigma$  modulators are designed specifically to allow one-bit quantization [7], [17], [18]. This simplifies the design of the divider in that only two moduli are required, but such  $\Delta\Sigma$  modulators tend to have spurious tones in their quantization noise that cannot be completely suppressed even with elaborate dithering techniques. Others of these higher-order  $\Delta\Sigma$  modulators, often referred to as MASH, cascaded, or multi-stage  $\Delta\Sigma$  modulators, are comprised of multiple lower-order  $\Delta\Sigma$  modulators, such as the second-order  $\Delta\Sigma$  modulator presented above, cascaded to obtain the equivalent of a single higher-order  $\Delta\Sigma$  modulator [5], [19], [20].

## V. $\Delta\Sigma$ FRACTIONAL- $N$ PLL DYNAMICS

A  $\Delta\Sigma$  fractional- $N$  PLL linearized model is derived in this section in the form of a block diagram that describes the output phase noise in terms of the component parameters and noise sources in the PLL. As in the case of an integer- $N$  PLL the model provides an accurate tool with which to predict the total phase noise, bandwidth, and stability of the PLL.

### A. Derivation of a $\Delta\Sigma$ fractional- $N$ PLL Linearized Model

In PLL analyses it is common to assume that each periodic signal within the PLL has the form  $v(t) = A(t) \sin(\omega t + \theta(t))$ , where  $A(t)$  is a positive *amplitude* function,  $\omega$  is a constant *center frequency* in radians/sec, and  $\theta(t)$  is zero-mean *phase noise* in radians. In most cases of interest for PLL analysis, the amplitude is well modeled as a constant value, and the phase noise is very small relative to  $\pi$  with a bandwidth that is much lower than the center frequency. Solving for the time of the  $n^{\text{th}}$  positive-going zero crossing,  $\gamma_n$ , of  $v(t)$  gives  $\gamma_n = [n - \theta(\gamma_n)/(2\pi)] \cdot T$ , where  $T = 2\pi/\omega$  is the period of the signal. Therefore, the sequence,  $\gamma_n$ , is a sampled version of the phase noise with very little aliasing, so knowing the sequence and  $T$  is approximately equivalent to knowing the phase noise. This approximation is made throughout the following analysis.

The relationship between the charge pump output current and the PFD input signals is shown in Figure 10. Ideally, during the  $n^{\text{th}}$  reference period the charge pump output is a current pulse of amplitude  $I$  or  $-I$  and duration  $|t_n - \tau_n|$ , where  $t_n$  and  $\tau_n$  are the times of the charge pump output transitions trig-

gered by the positive-going edges of the divider output and reference signal, respectively. Therefore, the average current sourced or sunk by the charge pump during the  $n^{\text{th}}$  reference period is  $I(t_n - \tau_n)/T_{\text{ref}}$ . In practice, the PFD is usually designed such that, except for a possible constant offset, this result holds even though the current sources have finite rise and fall times [2].

The first step in deriving the model is to develop an expression for  $t_n - \tau_n$ . Ideally,  $\tau_n = nT_{\text{ref}}$ , but phase noise introduced by the reference source and PFD cause it to have the form

$$\tau_n = nT_{\text{ref}} - \frac{T_{\text{ref}}}{2\pi} [\theta_{\text{ref}}(\tau_n) + \theta_{\text{PFD}}(\tau_n)], \quad (3)$$

where  $\theta_{\text{ref}}(t)$  and  $\theta_{\text{PFD}}(t)$  are the reference source and PFD phase noise functions, respectively. If the VCO output were ideal its positive-going edges would be spaced at uniform intervals of  $T_{\text{ref}} / (N + \alpha)$ , where  $\alpha$  is the fractional part of the modulus (e.g.,  $\alpha = 51/492$  in Figure 5). Therefore, ideally,

$$t_n = \frac{T_{\text{ref}}}{N + \alpha} \sum_{k=0}^{n-1} (N + y[k]),$$

but in practice it deviates because of VCO phase noise,  $\theta_{\text{VCO}}(t)$ , divider phase noise,  $\theta_{\text{div}}(t)$ , and instantaneous deviations of the VCO control voltage from its ideal average value of  $\bar{v}_{\text{ctrl}} = (N + \alpha)/(T_{\text{ref}} K_{\text{VCO}})$ , where  $K_{\text{VCO}}$  is the VCO gain in units of Hz/Volt. As a result,

$$t_n = \frac{T_{\text{ref}}}{N + \alpha} \left[ \sum_{k=0}^{n-1} (N + y[k]) - k_{\text{VCO}} \int_0^{t_n} (v_{\text{ctrl}}(t) - \bar{v}_{\text{ctrl}}) dt - \frac{\theta_{\text{VCO}}(t_n)}{2\pi} \right] - \frac{T_{\text{ref}}}{2\pi} \theta_{\text{div}}(t_n),$$

which reduces to

$$t_n = nT_{\text{ref}} + \frac{T_{\text{ref}}}{N + \alpha} \left[ \sum_{k=0}^{n-1} (y[k] - \alpha) - k_{\text{VCO}} \int_0^{t_n} (v_{\text{ctrl}}(t) - \bar{v}_{\text{ctrl}}) dt - \frac{\theta_{\text{VCO}}(t_n)}{2\pi} \right] - \frac{T_{\text{ref}}}{2\pi} \theta_{\text{div}}(t_n). \quad (4)$$

Subtracting (3) from (4) yields an expression for the average current sourced or sunk by the charge pump during the  $n^{\text{th}}$  reference period:

$$I(t_n - \tau_n)/T_{\text{ref}} = I \left[ \frac{\sum_{k=0}^{n-1} (y[k] - \alpha) - k_{\text{VCO}} \int_0^{t_n} (v_{\text{ctrl}}(t) - \bar{v}_{\text{ctrl}}) dt - \frac{\theta_{\text{VCO}}(t_n)}{2\pi}}{N + \alpha} - \frac{\theta_{\text{div}}(t_n)}{2\pi} + \frac{\theta_{\text{ref}}(\tau_n)}{2\pi} + \frac{\theta_{\text{PFD}}(\tau_n)}{2\pi} \right]. \quad (5)$$

As mentioned above, the phase noise terms are assumed to have bandwidths that are much smaller than the reference frequency. Consequently, the sampling of the phase noise functions in (5) can be neglected, and the charge pump output can be modeled as a smoothly varying function of time with an average value over each reference period equal to that of (5).

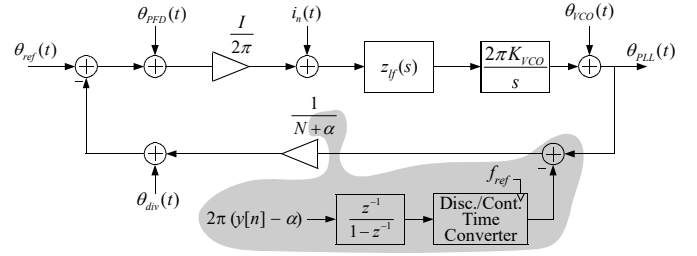


Figure 11: The  $\Delta\Sigma$  fractional- $N$  PLL linearized model. Except for the shaded region the model is identical to the corresponding integer- $N$  PLL model.

With these approximations, (5) implies that

$$i_{\text{cp}}(t) = I \left[ \frac{u_m(t) - k_{\text{VCO}} \int_0^t (v_{\text{ctrl}}(t) - \bar{v}_{\text{ctrl}}) dt - \frac{\theta_{\text{VCO}}(t)}{2\pi}}{N + \alpha} - \frac{\theta_{\text{div}}(t)}{2\pi} + \frac{\theta_{\text{ref}}(t)}{2\pi} + \frac{\theta_{\text{PFD}}(t)}{2\pi} \right], \quad (6)$$

where  $u_m(t)$  is the result of discrete-time integrating and converting to continuous-time the quantity,  $y[n] - \alpha$ .

The  $\Delta\Sigma$  fractional- $N$  PLL linearized model follows directly from (6) and Figure 10. It is shown in Figure 11, where  $i_{\text{cp}}(t)$  represents the noise contributed by the charge pump current sources and the loop filter, and  $z_f(s)$  is the transfer function of the loop filter. The model specifies the phase noise transfer functions and loop dynamics of the PLL. For example, the model implies that

$$\frac{\theta_{\text{PLL}}(s)}{\theta_{\text{ref}}(s)} = (N + \alpha) \frac{T(s)}{1 + T(s)}, \quad \text{and} \quad \frac{\theta_{\text{PLL}}(s)}{\theta_{\text{VCO}}(s)} = \frac{1}{1 + T(s)} \quad (7)$$

where

$$T(s) = \frac{IK_{\text{VCO}} z_f(s)}{s(N + \alpha)} \quad (8)$$

is the *loop gain* of the PLL. For the loop filter shown in Figure 10, the transfer function is

$$z_f(s) = \frac{1}{C_1 + C_2} \frac{1 + sRC_2}{s[1 + sRC_1C_2/(C_1 + C_2)]}. \quad (9)$$

### B. Differences Between the $\Delta\Sigma$ Fractional- $N$ and Integer- $N$ PLL Models

The shaded region in Figure 11 indicates the part of the model that is specific to  $\Delta\Sigma$  fractional- $N$  PLLs; except for the shaded region the model is identical to the corresponding model for integer- $N$  PLLs. Therefore, each phase noise transfer function in an integer- $N$  PLL is identical to the corresponding phase noise transfer function in a  $\Delta\Sigma$  fractional- $N$  PLL, except every occurrence of  $N$  in the former is replaced by  $N + \alpha$  in the latter. In most cases,  $N \gg 1$  and  $\alpha < 1$ , so  $N + \alpha \approx N$  and the corresponding transfer functions in integer- $N$  and  $\Delta\Sigma$  fractional- $N$  PLLs are nearly identical in practice. Similarly, the loop dynamics and stability issues are nearly the same in  $\Delta\Sigma$  fractional- $N$  PLLs and integer- $N$  PLLs.

The primary difference between the  $\Delta\Sigma$  fractional- $N$  and integer- $N$  PLL models is the signal path corresponding to the  $\Delta\Sigma$  modulator shown in the shaded region of Figure 11. The

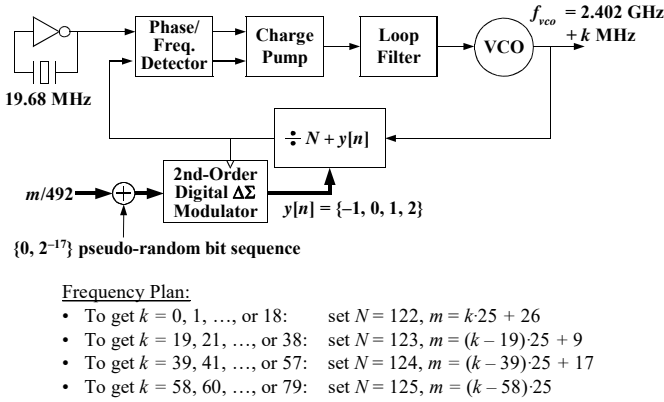


Figure 12: The example  $\Delta\Sigma$  fractional- $N$  PLL and frequency plan for generation of the Bluetooth wireless LAN RF channel frequencies.

sequence,  $y[n] - \alpha$ , consists of  $\Delta\Sigma$  modulator quantization noise,  $e_m[n]$ , which, as described previously, gives rise to phase error in the PLL output. For the example second-order  $\Delta\Sigma$  modulator it follows from the results presented in Section IV and the  $\Delta\Sigma$  fractional- $N$  PLL model equations presented above that the PLL phase noise component resulting from  $e_m[n]$  has a PSD given by

$$S_{\theta_{PLL}}(f) \Big|_{\Delta\Sigma \text{ only}} = 10 \cdot \log \left[ \frac{(2\pi)^2}{12f_{ref}} \left[ 2 \cdot \sin \left( \frac{\pi f}{f_{ref}} \right) \right]^2 \left| \frac{1}{N + \alpha} \cdot \frac{\theta_{PLL}(j2\pi f)}{\theta_{ref}(j2\pi f)} \right|^2 \right] \text{ dBc/Hz.} \quad (10)$$

The argument of the log function has the form of a highpass function times a lowpass function, which is consistent with the claim in Section III that the PLL lowpass filters the primarily high frequency quantization noise from the  $\Delta\Sigma$  modulator. It follows from (10) that the phase noise resulting from  $e_m[n]$  can be decreased by reducing the PLL bandwidth or increasing the reference frequency. If a higher-order  $\Delta\Sigma$  modulator is used, an equation similar to (10) results except that the exponent of the sinusoid is greater than two. This reduces the in-band portion of the quantization noise, but increases the out-of-band portion, which, depending upon the loop parameters of the PLL, can result in a somewhat lower overall phase noise. However, the PLL loop filter is highly constrained to maintain PLL stability, so the phase noise reduction that can be achieved by increasing the order of the  $\Delta\Sigma$  modulator is limited in most applications [16].

### C. A System Design Example

The PLL bandwidth and the phase margin both depend upon the loop gain,  $T(s)$ , which, for the loop filter shown in Figure 10, depends upon the parameters  $f_{ref}$ ,  $N$ ,  $I$ ,  $K_{VCO}$ ,  $R$ ,  $C_1$ , and  $C_2$ . Usually,  $f_{ref}$  and  $N$  are dictated by the application, and  $I$  and  $K_{VCO}$  are, at least partially, dictated by circuit design choices. This leaves the loop filter components as the main variables with which to set the desired PLL bandwidth, phase margin, and  $\Delta\Sigma$  modulator quantization noise suppression.

The process is demonstrated below for the  $\Delta\Sigma$  fractional- $N$  PLL presented in Section III to generate the local oscillator

frequencies in a direct conversion Bluetooth wireless LAN transceiver. The PLL is shown in Figure 12 with additional detail regarding the frequency plan. As described previously, the desired output frequencies are  $f_{VCO} = 2.402$  GHz +  $k$  MHz for  $k = 0, \dots, 78$ , and the crystal reference frequency is 19.68 MHz. Each of the 79 possible output frequencies is chosen by selecting  $m$  and  $N$  as indicated in the figure. In each case, the divider modulus is restricted to the set of four integers  $\{N - 1, N, N + 1, N + 2\}$ . The combinations of  $m$  and  $N$  were chosen to achieve the desired output frequencies yet keep the signals at the input of the  $\Delta\Sigma$  modulator sufficiently small so as not to overload the  $\Delta\Sigma$  modulator [11].

Typical requirements for such a PLL are that the loop bandwidth must be greater than 40 kHz, the phase margin must be greater than  $60^\circ$ , and the PLL phase noise be less than  $-120$  dBc/Hz at offsets from the carrier of 3 MHz and above. Assume that the VCO, divider, PFD, and charge pump circuits have been designed such that the overall PLL phase noise specification can be met provided the phase noise contributed by the  $\Delta\Sigma$  modulator and loop filter are each less than  $-130$  dBc/Hz at offsets from the carrier of 3 MHz and above. Furthermore, assume that the VCO and charge pump circuits are such that  $K_{VCO}$  and  $I$  are 200 MHz/V and 200  $\mu$ A, respectively, and that the loop filter has the form shown in Figure 10. Thus, the remaining design task is to choose the loop filter components such that the bandwidth, phase margin, and phase noise specifications are met.

The PLL phase margin, bandwidth, and phase noise arising from  $\Delta\Sigma$  modulator quantization noise can be derived from the linearized model equations, (7) through (10). While this can be done directly, it involves the solution of third order equations which can be messy. Alternatively, approximate solutions of the equations can be derived that provide better intuition [21]. A particularly convenient set of approximate solutions are

$$PM = \tan^{-1} \left( \frac{b-1}{2\sqrt{b}} \right), \quad (11)$$

$$f_{BW} = \frac{IK_{VCO}R}{2\pi N} \cdot \frac{b-1}{b}, \quad (12)$$

$$RC_2 = \frac{\sqrt{b}}{2\pi f_{BW}}, \quad (13)$$

and

$$S_{\theta_{PLL}}(f) \Big|_{\Delta\Sigma \text{ only}} \approx 10 \cdot \log \left[ \frac{4\pi^2 b}{3f_{ref}} \sin^2 \left( \frac{\pi f}{f_{ref}} \right) \left( \frac{f_{BW}}{f} \right)^4 \right] \text{ dBc/Hz,} \quad (14)$$

where  $PM$  is the phase margin of the PLL,  $f_{BW}$  is the 3 dB bandwidth of the PLL, and  $b = 1 + C_2/C_1$  is a measure of the separation between the two loop filter capacitors [22]. The derivations assume that  $b$  is greater than about 10, and (14) is valid for frequencies greater than  $(C_2 + C_1)/(2\pi RC_2 C_1)$ .

These equations are sufficient to determine appropriate loop filter component values. For example, suppose  $b$  is set to 49, so, as indicated by (11), the phase margin is approximately  $70^\circ$ . Solving (14) with the phase noise set to  $-130$  dBc/Hz at  $f = 3$  MHz indicates that  $f_{BW} \approx 50$  kHz. Therefore, the phase

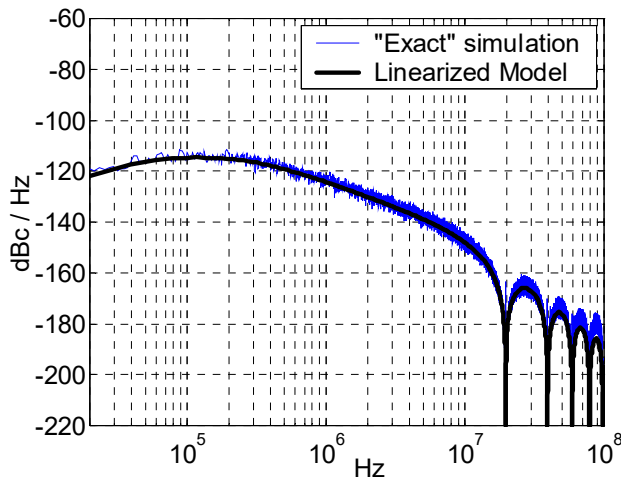


Figure 13: Simulated and calculated PSD plots of the phase noise arising from  $\Delta\Sigma$  modulator quantization noise for the example  $\Delta\Sigma$  fractional- $N$  PLL.

noise resulting from  $\Delta\Sigma$  modulator quantization noise is sufficiently suppressed with a 50 kHz bandwidth and a phase margin of  $70^\circ$ . With this information (12) can be solved to find  $R = 960 \Omega$  with which (13) and the definition of  $b$  can be used to calculate  $C_2 = 23 \text{ nF}$  and  $C_1 = 480 \text{ pF}$ . It is straightforward to verify that the phase noise introduced by the loop filter resistor (the only noise source in the loop filter) is well below  $-130 \text{ dBc/Hz}$  at offsets from the carrier of 3 MHz and above as required.

Figure 13 shows PSD plots of the phase noise arising from  $\Delta\Sigma$  modulator quantization noise for the example PLL with the loop filter component values derived above. The heavy curve was calculated directly from the linearized model equations (7) through (10). The light curve was obtained through a behavioral computer simulation of the PLL. As is evident from the figure, the two curves agree very well which suggests that the approximations made in obtaining the linearized model are reasonable.

An effect that does not have a counterpart in integer- $N$  PLLs is the presence of zeros in the PSD of the phase noise arising from  $\Delta\Sigma$  modulator quantization noise at multiples of the reference frequency. These zeros are a result of the discrete-to-continuous-time conversion of the  $\Delta\Sigma$  modulator quantization noise; each zero is a sampling image of the dc zero imposed on the quantization noise by the  $\Delta\Sigma$  modulator.

## VI. $\Delta\Sigma$ FRACTIONAL- $N$ PLL SPECIFIC PROBLEMS

One of the most significant problems specific to  $\Delta\Sigma$  fractional- $N$  PLLs is that they can be sensitive to modulus-dependent divider delays. In practice, each positive-going divider edge is separated from the VCO edge that triggered it by a propagation delay. Ideally, this propagation delay is independent of the corresponding divider modulus, in which case it introduces a constant phase offset but does not otherwise contribute to the phase noise. However, if the propagation delay depends upon the divider modulus and the number of  $\Delta\Sigma$  modulator output levels is greater than two, the effect is that of a hard non-linearity applied to the  $\Delta\Sigma$  modulator quan-

tization noise. This tends to fold out-of-band  $\Delta\Sigma$  modulator quantization noise to low frequencies and introduce spurious tones, which can significantly increase the PLL phase noise. The problem is analogous to that of multi-bit digital-to-analog converter step-size mismatches in analog  $\Delta\Sigma$  data converters [23]. Unfortunately, circuit simulations are required to evaluate the severity of the problem on a case by case basis as both the extent of any modulus-dependent delays and their affect on the PLL phase noise are difficult to predict using hand analysis.

There are two well-known solutions to this problem. One solution is to resynchronize the divider output to the nearest VCO edge or at least a higher-frequency edge obtained from within the divider circuitry [22], [24]. The resynchronization erases memory of modulus-dependent delays and noise introduced within the divider circuitry, but care must be taken to ensure that the signal used for resynchronization is itself free of modulus dependent delays. The primary drawback of the approach is that it increases power consumption.

The other solution is to use a  $\Delta\Sigma$  modulator with single-bit (i.e., two level) quantization. In this case, modulus-dependent delays give rise to phase error at the output of the divider that consists of a constant offset plus a scaled version of the  $\Delta\Sigma$  modulator quantization noise. Since, by design, the  $\Delta\Sigma$  modulator quantization noise has most of its power outside the PLL bandwidth, the modulus-dependent delays increase the phase noise only slightly. Unfortunately,  $\Delta\Sigma$  modulators with single-bit quantization tend not to perform as well as  $\Delta\Sigma$  modulators with multi-bit (i.e., more than two-level) quantization. For example, if the 9-level quantizer in the 48 Msample/s  $\Delta\Sigma$  modulator example presented in Section IV were replaced by a one-bit quantizer, the dynamic range of the  $\Delta\Sigma$  modulator in the zero to 500 kHz band would be reduced from 88.5 dB to approximately 65 dB. Moreover, unlike the 9-level quantizer case, the additive noise from the single-bit quantizer would not be white and would be correlated with the input sequence. Its variance would be input dependent and it would contain spurious tones.

These problems can be mitigated by using a higher-order  $\Delta\Sigma$  modulator architecture to more aggressively suppress the in-band portion of the additive noise from the two-level quantizer. However, to maintain stability in a higher-order  $\Delta\Sigma$  modulator with single-bit quantization, the useful input range of the  $\Delta\Sigma$  modulator input signal must be reduced and more poles and zeros must be introduced within the feedback loop as compared to a multi-bit design with a comparable dynamic range. Even then, the problem of spurious tones persists, and it is difficult to predict where they will appear except through extensive simulation. Furthermore, to compensate for the restricted input range of the  $\Delta\Sigma$  modulator the reference frequency must be large enough that all of the desired PLL output frequencies can be achieved. This can severely limit design flexibility. For example, if the magnitude of the  $\Delta\Sigma$  modulator input signal were limited to less than 0.5 in the case of the Bluetooth local oscillator application considered above, the reference frequency would have to be greater than 79 MHz. Otherwise, it would not be possible to generate all the



Bluetooth channel frequencies.

Another issue specific to  $\Delta\Sigma$  fractional- $N$  PLLs is that modulus switching increases the average duration over which the charge pump current sources are turned on each period relative to integer- $N$  PLLs. For comparison, consider a  $\Delta\Sigma$  fractional- $N$  PLL and an integer- $N$  PLL with the same  $N$  (where  $N \gg \alpha$ ), the same  $f_{ref}$ , and identical loop components. It follows from (5) that

$$(t_n - \tau_n) \Big|_{\text{Fractional-}N} = (t_n - \tau_n) \Big|_{\text{Integer-}N} + \frac{T_{ref}}{N + \alpha} \sum_{k=0}^{n-1} (y[k] - \alpha). \quad (15)$$

The last term in (15), which is caused by having the  $\Delta\Sigma$  modulator switch the divider modulus, represents a significant increase in the time during which the charge pump current sources are turned on each reference period. Consequently, the phase noise arising just from charge pump current source noise is larger in the  $\Delta\Sigma$  fractional- $N$  PLL by

$$A \cdot \log \left[ \frac{\text{Average fractional-}N \text{ PLL charge pump "on time"}}{\text{Average integer-}N \text{ PLL charge pump "on time"}} \right]$$

where  $A$  is a constant between 10 and 20. The value of  $A$  depends upon the autocorrelation of the charge pump current source noise. For example, if the current source noise in successive charge pump pulses is completely uncorrelated, then  $A$  is 10. Near the other extreme,  $A$  is close to 20.

## VII. TECHNIQUES TO WIDEN $\Delta\Sigma$ FRACTIONAL- $N$ PLL LOOP BANDWIDTHS

A transmitter with virtually any modulation format can be implemented using D/A conversion to generate analog baseband or IF signals and upconversion to generate the final RF signal. However, many of the commonly used modulation formats in wireless communication systems such as MSK and FSK involve only frequency or phase modulation of a single carrier [25]. In such cases, the transmitted signal can be generated by modulating a radio frequency (RF) VCO, thereby eliminating the need for conventional upconversion stages and much of the attendant analog filtering. At least two approaches have been successfully implemented in commercial wireless transmitters to date. One is based on open-loop VCO modulation, and the other is based on  $\Delta\Sigma$  fractional- $N$  synthesis.

An example of a commercial transmitter that uses the open-loop VCO modulation technique is presented in [26] and [27], in this case for a DECT cordless telephone. Between transmit bursts, the desired center frequency is set relative to a reference frequency by enclosing the VCO within a conventional PLL. During each transmit burst the VCO is switched out of the PLL and the desired frequency modulation is applied directly to its input. The primary limitation of the approach is that it tends to be highly sensitive to noise and interference from other circuits. For example, in [27], the required level of isolation precluded the implementation of a single-chip transmitter. Furthermore, the modulation index of the transmitted signal depends upon the absolute tolerances of the VCO components which are often difficult to control in low-

cost VLSI technologies and can also drift rapidly over time.

In principle,  $\Delta\Sigma$  fractional- $N$  PLLs can avoid these problems by modulating the VCO within the PLL. This can be done by driving the input of the digital  $\Delta\Sigma$  modulator with the desired frequency modulation of the transmitted signal. The primary limitation is that bandwidth of the PLL must be narrow enough that the quantization noise from the  $\Delta\Sigma$  modulator is sufficiently attenuated, but sufficiently high to allow for the modulation. For instance, the phase noise PSD of the example  $\Delta\Sigma$  fractional- $N$  PLL shown in Figure 5 with a 50 kHz loop bandwidth meets the necessary phase noise specifications when used as a local oscillator in a conventional upconversion stage within a Bluetooth wireless LAN transmitter. However, if the Bluetooth transmitter is to be implemented by modulating the VCO through the digital  $\Delta\Sigma$  modulator, then the loop bandwidth of the PLL must be approximately 500 kHz. Unfortunately, when the loop bandwidth of the fractional- $N$  PLL shown in Figure 5 is widened to 500 kHz, the resulting phase noise becomes too large to meet the Bluetooth transmit requirements.

Nevertheless, commercial transmitters with VCO modulation through  $\Delta\Sigma$  fractional- $N$  synthesizers are beginning to be deployed, especially in low-performance, low-cost wireless systems such as Bluetooth wireless LANs [28]. Facilitating this trend are various solutions that have been devised in recent years to allow for wideband VCO modulation in  $\Delta\Sigma$  fractional- $N$  PLLs without incurring the phase noise penalty mentioned above. One of the solutions is to keep the loop bandwidth relatively low, but pre-emphasize (i.e., highpass filter) the digital phase modulation signal prior to the digital  $\Delta\Sigma$  modulator [29]. Unfortunately, this approach requires the highpass response of the digital pre-emphasis filter to be a reasonably close match to the inverse of the closed-loop filtering imposed by the largely analog PLL. Another of the solutions is to use a high-order loop filter in the PLL with a sharp lowpass response [30]. Increasing the order of the loop filter increases the attenuation of out-of-band quantization noise which allows for higher-order  $\Delta\Sigma$  modulation to reduce in-band quantization noise thereby allowing the loop bandwidth to be increased without increasing the total phase noise. However, as described in [30], this necessitates the use of a Type 1 PLL which significantly complicates the design of the phase detector. Yet another solution is to use a narrow loop bandwidth but modulate the VCO both through the digital  $\Delta\Sigma$  modulator and through an auxiliary modulation port at the VCO input [28]. The idea is to apply the low-frequency modulation components at the  $\Delta\Sigma$  modulator input and the high frequency modulation components directly to the VCO. Again, matching is an issue, but it has proven to be manageable at least for low-end applications such as Bluetooth transceivers.

## VIII. CONCLUSION

The additional concepts and issues associated with  $\Delta\Sigma$  fractional- $N$  PLLs for frequency synthesis relative to integer- $N$  PLLs have been presented. It has been shown that  $\Delta\Sigma$  fractional- $N$  PLLs provide tuning resolution limited only by digi-

tal logic complexity, and, in contrast to integer- $N$  PLLs, increased tuning resolution does not come at the expense of reduced bandwidth. Since one of the main innovations in a  $\Delta\Sigma$  fractional- $N$  PLL is the use of a  $\Delta\Sigma$  modulator to control the divider modulus, the relevant concepts underlying  $\Delta\Sigma$  modulation have been described in detail. A linearized model has been derived from first principles and a design example has been presented to illustrate how the model is used in practice. Techniques for wideband digital modulation of the VCO within a delta-sigma fractional- $N$  PLL have also been presented.

#### ACKNOWLEDGEMENTS

The author is grateful to Sudhakar Pamarti, Eric Siragusa, and Ashok Swaminathan for their helpful discussions and advice regarding this paper.

#### REFERENCES

1. P. M. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. COM-28, pp. 1849-1858, November 1980.
2. B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw Hill, 2001.
3. Bluetooth Wireless LAN Specification, Version 1.0, 2000.
4. U. L. Rohde, *Microvave and Wireless Synthesizers Theory and Design*, John Wiley & Sons, 1997.
5. B. Miller, B. Conley, "A multiple modulator fractional divider," *Annual IEEE Symposium on Frequency Control*, vol. 44, pp. 559-568, March 1990.
6. B. Miller, B. Conley, "A multiple modulator fractional divider," *IEEE Transactions on Instrumentation and Measurement*, vol. 40, no. 3, pp. 578-583, June 1991.
7. T. A. Riley, M. A. Copeland, T. A. Kwasniewski, "Delta-sigma modulation in fractional- $N$  frequency synthesis," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 5, pp. 553-559, May, 1993.
8. S. K. Tewksbury, R. W. Hallock, "Oversampled, linear predictive and noise-shaping coders of order  $N > 1$ ," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, pp. 436-447, July 1978.
9. G. Lainey, R. Saintlaurens, P. Senn, "Switched-capacitor second-order noise-shaping coder," *IEE Electronics Letters*, vol. 19, pp. 149-150, February 1983.
10. I. Galton, "Granular quantization noise in a class of delta-sigma modulators," *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 848-859, May 1994.
11. N. He, F. Kuhlmann, A. Buzo, "Multiloop sigma-delta quantization," *IEEE Transactions on Information Theory*, vol. 38, no.3, pp.1015-1028, May 1992.
12. I. Galton, "One-bit dithering in delta-sigma modulator-based D/A conversion," Proc. of the IEEE International Symposium on Circuits and Systems, 1993.
13. S. W. Golomb, *Shift Register Sequences*. Laguna Hills, CA: Aegean Park Press, 1982
14. E. J. McCluskey, *Logic Design Principles*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
15. S. K. Tewksbury, R. W. Hallock, "Oversampled, linear predictive and noise-shaping coders of order  $N > 1$ ," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, pp. 436-447, July 1978.
16. W. Rhee, B. S. Song, A. Ali, "A 1.1-GHz CMOS fractional- $N$  frequency synthesizer with a 3-b third-order  $\Delta\Sigma$  modulator," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 10, pp. 1453-1460, October 2000.
17. W. L. Lee, C. G. Sodini, "A topology for higher order interpolative coders," *Proceedings of the 1987 IEEE International Symposium on Circuits and Systems*, vol. 2, pp.459-462, May 1987.
18. K. C.-H. Chao, S. Nadeem, W. L. Lee, C. G. Sodini, "A higher order topology for interpolative modulators for oversampling A/D converters," *IEEE Transactions on Circuits and Systems*, vol. 37, no.3, p.309-318, March 1990.
19. Y. Matsuya, K. Uchimura, A. Iwata, T. Kobayashi, M. Ishikawa, T. Yoshitome, "A 16-bit oversampling A-to-D conversion technology using triple integration noise shaping," *IEEE Journal of Solid-State Circuits*, vol. SC-22, pp. 921-929, December 1987.
20. K. Uchimura, T. Hayashi, T. Kimura, A. Iwata, "Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. AASP-36, pp. 1899-1905, December 1988.
21. J. Craninckx, M. S. J. Steyaert, "A fully integrated CMOS DCS-1800 frequency synthesizer," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 2054-2065, December 1998.
22. S. Pamarti, "Techniques for Wideband Fractional- $N$  Phase-Locked Loops," PhD Dissertation, University of California, San Diego, 2003.
23. S. R. Norsworthy, R. Schreier, G. C. Temes, Eds. *Delta-Sigma Data Converters, Theory, Design, and Simulation*, New York: IEEE Press, 1997.
24. L. Lin, L. Tee, P. R. Gray, "A 1.4 GHz differential low-noise CMOS frequency synthesizer using a wideband PLL architecture", *IEEE ISSCC Digest of Technical Papers*, pp. 204-205, Feb. 2000.
25. J. G. Proakis, *Digital Communications*, fourth ed., McGraw Hill, 2000.
26. S. Heinen, S. Beyer, J. Fenk, "A 3.0 V 2 GHz transmitter IC for digital radio communication with integrated VCO's," Digest of Technical Papers, *IEEE International Solid-State Circuits Conference*, vol. 38, pp. 150-151, Feb. 1995.
27. S. Heinen, K. Hadjizada, U. Matter, W. Geppert, V. Thomas, S. Weber, S. Beyer, J. Fenk, E. Matshke, "A 2.7 V 2.5 GHz bipolar chipset for digital wireless communi-

- cation,” Digest of Technical Papers, *IEEE International Solid-State Circuits Conference*, vol. 40, pp. 306-307, Feb. 1997.
28. N. Filiol, et. al., “A 22 mW Bluetooth RF transceiver with direct RF modulation and on-chip IF filtering,” Digest of Technical Papers, *IEEE International Solid-State Circuits Conference*, vol. 43, pp. 202-203, Feb. 2001.
  29. M. H. Perrott, T. L. Tewksbury III, C. G. Sodini, “A 27-mW CMOS fractional-N synthesizer using digital compensation for 2.5-Mb/s GFSK modulation,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, pp. 2048-2059, Dec. 1997.
  30. S. Willingham, M. Perrott, B. Setterberg, A. Grzegorek, B. McFarland, “An integrated 2.5GHz  $\Sigma\Delta$  frequency synthesizer with 5 $\mu$ s settling and 2Mb/s closed loop modulation,” Digest of Technical Papers, *IEEE International Solid-State Circuits Conference*, vol. 43, pp. 200-201, Feb. 2000.



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

Since 1996 he has been a professor of electrical engineering at the University of California, San Diego where he teaches and conducts research in the field of mixed-signal integrated circuits and systems for communications. Prior to 1996 he was with UC Irvine, the NASA Jet Propulsion Laboratory, Acuson, and Mead Data Central. His research involves the invention, analysis, and integrated circuit implementation of key communication system blocks such as data converters, frequency synthesizers, and clock recovery systems. The emphasis of his research is on the development of digital signal processing techniques to mitigate the effects of non-ideal analog circuit behavior with the objective of generating enabling technology for highly integrated, low-cost, communication systems. In addition to his academic research, he regularly consults at several communications and semiconductor companies and teaches portions of various industry-oriented short courses on the design of data converters, PLLs, and wireless transceivers. He has served on a corporate Board of Directors and several corporate Technical Advisory Boards, and his is the Editor-in-Chief of the *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*.