A Duty-Cycle-Error-Immune Reference Frequency Doubling Technique for Fractional-*N* Digital PLLs

Amr I. Eissa, Enrique Alvarez-Fontecilla, Colin Weltin-Wu, *Member, IEEE*, and Ian Galton, *Fellow, IEEE*

***Abstract*— Increasing a PLL's reference frequency offers significant performance advantages, but doing so by increasing the PLL's crystal oscillator frequency is not a viable option in many applications. Instead, a frequency doubler can be used to derive a reference signal with twice the frequency of the crystal oscillator, but conventional PLLs are highly sensitive to the crystal oscillator's duty cycle error in such cases. Prior solutions to this problem involve calibration techniques which impose convergence speed versus accuracy tradeoffs. In contrast, this paper proposes a system modification which makes a PLL immune to such duty cycle errors without the need for calibration. The technique is presented and analyzed in the context of a delta-sigma frequency-to-digital converter (ΔΣ-FDC) based PLL. Analysis and behavioral simulations with nonideal circuit parameters show that the worst-case convergence time is at least 10 times faster than that of the prior techniques. Additionally, the proposed ΔΣ-FDC includes other modifications which improve its performance relative to comparable prior ΔΣ-FDCs.**

***Index Terms*— Background calibration, crystal oscillators, digital phase-locked loops (PLLs), duty-cycle error, fractional-*N* PLL, frequency doubler, frequency-to-digital converter (FDC).**

I. Introduction

PHASE locked loops (PLLs) are critical components in communication systems, and their performance requirements continue to increase as communication system standards evolve. In particular, the demand for PLLs with sub-100-fs rms jitter is increasing to enable higher data rates in wireless and wireline communication systems [1], [2], [3], [4], [5], [6]. Furthermore, reciprocal-mixing requirements in some wireless applications require PLLs with reference spurs below –80 dBc.

A PLL's phase noise spectrum usually is dominated by the phase noise of its controlled oscillator above the PLL's bandwidth and by noise from all other circuitry within the PLL's bandwidth.[1] The PLL's in-band phase noise consists of white and highpass shaped components, that are essentially sampled at the reference frequency, f_{ref}, so doubling f_{ref} for a given PLL bandwidth reduces the contribution to the PLL's phase noise of the white and first-order highpass shaped noise components by 3 dB and 9 dB, respectively [7]. This reduces the PLL's jitter by reducing the in-band noise, and makes it possible to further reduce the jitter by increasing the PLL's bandwidth to suppress the controlled oscillator's phase noise contribution over the wider bandwidth.

However, a PLL is generally but one component of a larger system, and its reference frequency is typically derived from the system's crystal oscillator. Unfortunately, the crystal oscillator frequency, f_{crystal}, is usually dictated by cost and system-level constraints, so increasing f_{crystal} is rarely an option when designing the PLL. Instead, a frequency doubler (FD), which uses the rising and falling edges of the crystal oscillator to generate a double-frequency reference signal, can be used to effectively double f_{ref}. The drawback of the approach is that crystal oscillators typically have duty cycle errors of 5 to 10% across process, voltage, and temperature (PVT) variations, and conventional PLLs with FDs are highly sensitive to such errors [8], [9], [10], [11], [12]. The duty-cycle error results in large spurs at integer-multiples of f_{crystal} and increase the PLL's jitter.

For example, a PLL with closed-loop bandwidth of 1.5 MHz with a 20dB/decade roll-off up to 76.8 MHz, a 10 GHz output frequency, and a 153.6 MHz reference signal from a 76.8 MHz crystal-oscillator with 5% duty-cycle error followed by an FD would have a −21.8 dBc spur at 76.8 MHz. This corresponds to 1.82 ps of jitter, not including any other error sources. Furthermore, the duty cycle error increases the dynamic range requirements of several of the PLL's circuit blocks, which generally increases their contributions to the PLL's phase noise and spurs.

Previously published techniques that address this problem rely on estimating the duty-cycle error in the analog or digital domains and canceling it through the crystal oscillator and FD analog circuitry, as in [13], [14], [15], [16], [17], and [18], or through the PLL's multi-modulus divider (MMD) as in [11], [12], [19], and [20]. However, these techniques are subject to a fundamental trade-off between convergence speed and accuracy. To sufficiently reduce noise, the error estimation circuitry must have a small bandwidth, which leads to long

[1] The controlled oscillator is a digitally-controlled oscillator in the case of a digital PLL and a voltage-controlled oscillator in the case of an analog PLL.

Manuscript received 19 March 2024; revised 12 June 2024; accepted 30 July 2024. Date of publication 28 August 2024; date of current version 2 October 2024. This work was supported in part by Qualcomm, in part by Analog Devices, and in part by the National Science Foundation under Award 1617545. This article was recommended by Associate Editor R. Garg. *(Corresponding author: Ian Galton.)*

Amr I. Eissa is with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92092 USA, and also with the Electronics and Communications Engineering Department, Ain Shams University, Cairo 11517, Egypt (e-mail: aeissa@ucsd.edu).

Enrique Alvarez-Fontecilla is with Analog Devices Inc., San Diego, CA 92128 USA.

Colin Weltin-Wu is with Aeonsemi, Inc., Santa Clara, CA 95134 USA.

Ian Galton is with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92092 USA (e-mail: galton@ucsd.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2024.3439210.

Digital Object Identifier 10.1109/TCSI.2024.3439210

convergence times. This tradeoff becomes more severe as the PLL's targeted jitter performance is improved.

A reference frequency-doubling (RFD) technique is presented in this paper which is immune to crystal oscillator duty-cycle error, so it is not subject to the speed-accuracy tradeoff of the prior solutions. It is presented and analyzed in the context of a delta-sigma frequency-to-digital converter ($\Delta\Sigma$-FDC) based PLL configured to achieve 75 fs rms jitter. Behavioral simulations with nonideal circuit parameters extracted from simulations of transistor-level PLL circuit blocks implemented in Global Foundries 22FDX 22 nm CMOS technology show that the worst-case convergence time is 412 reference cycles. This is at least 10× faster than that of the prior art with comparable initial duty-cycle errors and jitter. The presented $\Delta\Sigma$-FDC also includes a modified gain calibration technique and achieves reduced PFD and ADC spans after locking relative to comparable prior $\Delta\Sigma$-FDCs [21], [22].

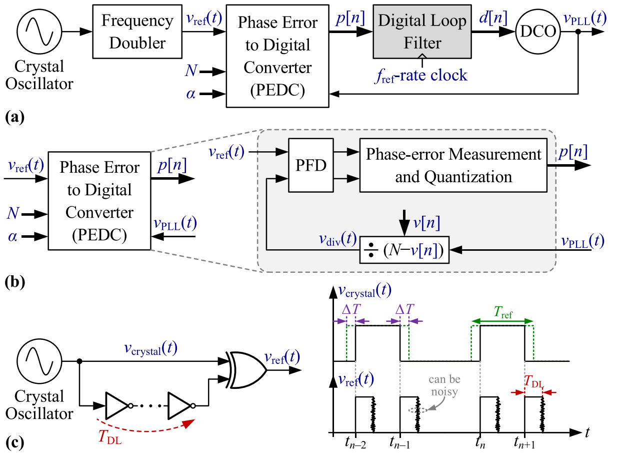

Fig. 1. (a) Generic fractional-N digital PLL using a reference frequency-doubler, (b) general form of commonly used PEDCs that use MMD, and (c) XOR-based frequency-doubler and associated waveforms.

II. Reference Frequency-Doubling in PLLs

Fig. 1(a) shows a top-level block diagram of a generic fractional-N digital PLL where the reference signal, $v_{\mathrm{ref}}(t)$, is generated by an FD, so the reference frequency, f_{ref}, is double that of the crystal frequency, f_{crystal}. The PLL is designed to generate a periodic output waveform, $v_{\mathrm{PLL}}(t)$, with frequency $f_{\mathrm{PLL}} = (N+\alpha)f_{\mathrm{ref}}$ where N is a positive integer and $-1/2 \le \alpha \le 1/2$. It consists of a phase-error to digital converter (PEDC), a digital loop filter (DLF), and a digitally controlled oscillator (DCO). The PEDC output, $p[n]$, is a quantized measure of the PLL's phase error and the DCO's frequency control sequence, $d[n]$, is a lowpass filtered version of $p[n]$.

The PEDC in many digital PLLs incorporates an MMD and a PFD as in Fig. 1(b). The nth and $(n+1)$th rising edges of the MMD output, $v_{\mathrm{div}}(t)$, are separated by $N-v[n]$ DCO periods, where $v[n]$ is an integer-valued sequence generated within the PEDC. The PFD output pulse width, which is equal to the time-difference between the rising edges of $v_{\mathrm{ref}}(t)$ and $v_{\mathrm{div}}(t)$, is measured and quantized by the phase-error measurement and quantization (PEMQ) circuitry. The sequence $v[n]$ can be generated by a digital re-quantizer such as in [23], [24], [25], and [26] or by linearly filtering $p[n]$ as in [21], [27], [28], [29], and [30]. In each case, the PLL settles such that the mean of $v[n]$ is $-\alpha$, so the PLL's mean output frequency is $(N+\alpha)f_{\mathrm{ref}}$.

Fig. 1(c) shows the details of a widely-used FD. It consists of an inverter-based delay-line and an XOR-gate arranged such that both the rising and falling edges of its input signal, $v_{\mathrm{crystal}}(t)$, cause rising edges in its output signal, $v_{\mathrm{ref}}(t)$. Usually, it is mainly the white portion of the phase noise of a crystal oscillator that contributes to a PLL's output phase noise as the crystal's high quality factor relegates other noise to a very low bandwidth. In addition, crystal oscillators typically include squaring-up buffers, which sharpen the oscillator's output edges to reduce the jitter added by subsequent circuitry [31]. Consequently, to the extent that the FD is ideal, the jitter of $v_{\mathrm{ref}}(t)$ is well-modeled as a white sequence and, as shown in [31], the phase noise power spectral density (PSD) of $v_{\mathrm{ref}}(t)$ in this case is 3 dB higher than that of $v_{\mathrm{crystal}}(t)$. Had the FD not been used in the system of Fig. 1, both N and α would have had to be doubled to achieve the same PLL output frequency, and as the PLL's phase noise transfer function from $v_{\mathrm{ref}}(t)$ is scaled by $(N+\alpha)^2$, it follows that the FD provides a net 3 dB reduction in the crystal oscillator's contribution to the PLL's phase noise.

The FD also reduces the PEDC's contribution to the PLL's phase noise. The PEDC's measurement noise is a combination of white and highpass shaped components, depending on the PEDC's design. As the noise components are essentially sampled at a rate of f_{ref}, doubling f_{ref} for a given PLL bandwidth reduces the contribution to the PLL's output phase noise of all white and first-order highpass shaped noise components by 3 dB and 9 dB, respectively [7].

Unfortunately, practical FDs, including that shown in Fig. 1(c), introduce deterministic jitter when the duty cycle of $v_{\mathrm{crystal}}(t)$ is not exactly 50%, and conventional PLLs are highly sensitive to such jitter. If the crystal oscillator's duty cycle were exactly 50%, the time of the nth rising edge of $v_{\mathrm{ref}}(t)$ could be written as $t_n = nT_{\mathrm{ref}} - j_{\mathrm{ref}}[n]$, where $j_{\mathrm{ref}}[n]$ is the reference signal's jitter [31]. As illustrated in Fig. 1(c), when the duty-cycle of $v_{\mathrm{crystal}}(t)$ deviates from 50%, the t_n values are further displaced relative to their ideal values by an alternating error sequence, i.e.,

$$t_n = nT_{\mathrm{ref}} - j_{\mathrm{ref}}[n] + d_e[n], \tag{1}$$

where

$$d_e[n] = \Delta T(-1)^n \text{ with } \Delta T = \left(\frac{D}{100} - 0.5\right)T_{\mathrm{ref}}, \tag{2}$$

and D is the crystal oscillator's duty cycle in percent. In practical crystal oscillators, D typically deviates from its ideal value of 50% by anywhere between 5 and 10 percentage points across PVT variations [8], [9], [10], [11], [12].

In a conventional PLL, such duty cycle errors deteriorate the performance of the PEDC circuitry and introduce a large $f_{\mathrm{ref}}/2$ spur at the PLL's output. As $p[n]$ is a quantized measure of the time-difference between the rising edges of $v_{\mathrm{ref}}(t)$ and

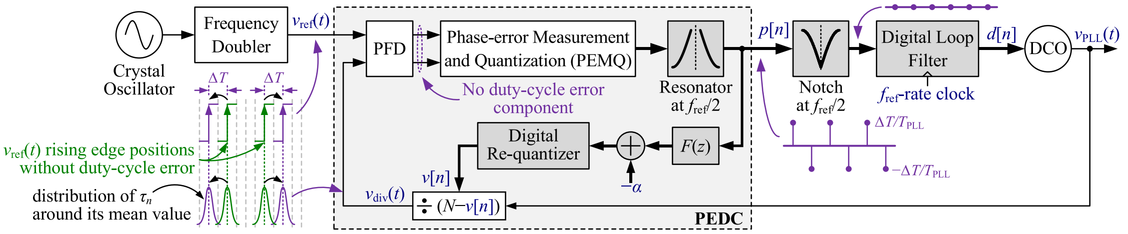
Crystal Oscillator
Frequency Doubler
$v_{\text{ref}}(t)$
ΔT
ΔT
$v_{\text{ref}}(t)$ rising edge positions without duty-cycle error
distribution of τ_n around its mean value
PFD
Phase-error Measurement and Quantization (PEMQ)
No duty-cycle error component
Resonator at $f_{\text{ref}}/2$
Digital Re-quantizer
$F(z)$
$-\alpha$
$v[n]$
$v_{\text{div}}(t)$
$\div\,(N-v[n])$
PEDC
$p[n]$
Notch at $f_{\text{ref}}/2$
$\Delta T/T_{\text{PLL}}$
$-\Delta T/T_{\text{PLL}}$
Digital Loop Filter
f_{ref}-rate clock
$d[n]$
DCO
$v_{\text{PLL}}(t)$

Fig. 2. High-level block diagram of a generic fractional-N digital PLL with the proposed reference frequency-doubling scheme.

$v_{\text{div}}(t)$, it contains a term proportional to $d_e[n]$ which causes the above-mentioned $f_{\text{ref}}/2$ spur. The $f_{\text{ref}}/2$ spur could be perfectly removed via a digital notch filter, but this would not eliminate the presence of $d_e[n]$ at the PEMQ's input. For typical values of D, $d_e[n]$ is much larger than $j_{\text{ref}}[n]$ in (1), so the need to accommodate $d_e[n]$ drastically increases the dynamic range, linearity, and noise performance required of the PEMQ.

For example, suppose a 76.8 MHz crystal oscillator with 5% duty-cycle error is used in a PLL to generate a 10 GHz output waveform, and $v[n]$ is generated by a second-order delta-sigma modulator. In the absence of duty-cycle error, the PFD's output nominal span is $2T_{\text{PLL}}$ [32]. It can be deduced from (1) and (2) that a 5% duty-cycle error increases the PFD output span by a factor of 4.25. In a TDC-based PLL, such as presented in [23], this would require two additional bits of TDC dynamic range which would typically quadruple the TDC's power consumption. An increase in the TDC's dynamic range would also degrade the TDC's linearity and, hence, the PLL's spurious tone performance. In a bang-bang PLL, such as presented in [24], the alternating $d_e[n]$ error would push the bang-bang phase-detector far away from its optimal operating point and the PLL may even fail to lock as the bang-bang phase detector's effective gain would be very small [11], [33]. In a charge-pump (CP) FDC-based PLL, such as presented in [21], the increase in the PFD output span would increase the thermal and flicker CP noise contributions by approximately 6.3 and 12.6 dB, respectively. For a 100 fs rms jitter PLL design with a CP jitter contribution of 50 fs, the increase in the CP's white component alone would increase the PLL's jitter by 35%.

To enable high-performance PLLs with sub-100 fs rms jitter, spurious tones at $f_{\text{ref}}/2$ and its multiples below -80 dBc, and practical PEMQ performance requirements, either a scheme to reduce D or a scheme to cancel $d_e[n]$ prior to the PEMQ must be employed.

III. Proposed RFD Technique Qualitative Description

Fig. 2 shows a top-level block diagram of the proposed scheme. It is a modified version of the system of Fig. 1 in which the PEDC includes an $f_{\text{ref}}/2$ resonator in its forward path, there is an $f_{\text{ref}}/2$ notch filter between the PEDC and the DLF, and the MMD control word, $v[n]$, is generated by adding $-\alpha$ to a digitally filtered version of $p[n]$ and re-quantizing the result. The transfer function, $F(z)$, of the filter applied to $p[n]$, and the transfer function of the $f_{\text{ref}}/2$ resonator are designed such that $p[n]$ is a measure of the PLL's phase-error as in conventional digital PLL architectures, and such that the PEDC is stable.

The proposed technique can be understood qualitatively as follows. The resonator has an infinite gain at $f_{\text{ref}}/2$, so any $f_{\text{ref}}/2$ spur at its input would cause its output to grow without bound. However, as the system is stable by design, the output of the resonator must be bounded, so its input must not contain an $f_{\text{ref}}/2$ tone. The PEMQ circuitry does not introduce a zero at $f_{\text{ref}}/2$ by design, so it follows that the PFD's output must also be free of any $f_{\text{ref}}/2$ spur. Consequently, the system must settle such that the times of the MMD rising edges, defined as τ_n for $n = 0, 1, 2, \ldots$, contain a component that is exactly equal to the duty-cycle error sequence, $d_e[n]$, in the reference path, which implies that $d_e[n]$ is perfectly canceled at the PFD's output.

For the MMD output edges to contain a component equal to $d_e[n]$, $p[n]$ must contain a term proportional to $d_e[n]$. The $f_{\text{ref}}/2$ notch filter following the PEDC removes this term, thereby preventing it from causing an $f_{\text{ref}}/2$ spur in the PLL's output waveform. The effects of the notch filter on the PLL's noise performance and loop dynamics are negligible because the PLL's loop bandwidth is generally much smaller than f_{ref} [32].

As proven in the next section, the proposed technique is free of convergence bias and, in contrast to the techniques presented in [11], [12], [18], [19], and [20], is not subject to the fundamental LMS loop speed-accuracy tradeoff. Furthermore, the PFD outputs only depend on the rising edges of $v_{\text{ref}}(t)$, so jitter on the falling edges of $v_{\text{ref}}(t)$, which arises primarily from noise introduced by the FD's inverter-based delay chain (Fig. 1(c)), does not degrade the PLL's phase noise. Therefore, in contrast to the techniques presented in [13], [14], [15], [16], and [17], there is no additional noise or power consumption penalty associated with adding delay lines in the reference signal path.

If $d_e[n]$ were measured in the analog domain or if the divider or reference edges were shifted in the analog domain by controlling a delay line, the technique would be subject to inaccuracies from nonideal analog circuit behavior. Instead, the proposed technique avoids such inaccuracies by canceling $d_e[n]$ precisely with a digital-domain feedback path through the MMD via $v[n]$. In principle, the technique can be applied to any digital PLL of the form shown in Fig. 1 by adding such a feedback path through the MMD input, $v[n]$. However, FDC-PLLs already contain a feedback path through the MMD to which the technique can be added, so applying the technique to an FDC-PLL requires fewer modifications than applying it to other types of PLLs [21], [27], [29].

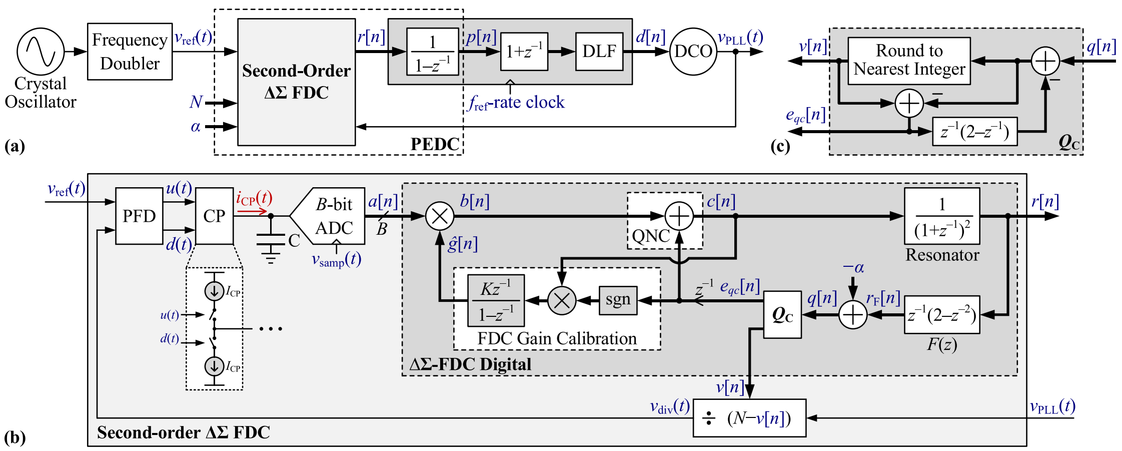

Fig. 3. Block diagram of the proposed FDC-PLL: (a) PLL top-level block diagram, (b) second-order $\Delta\Sigma$-FDC block diagram, and (c) coarse-quantizer, Q_C, implementation details.

IV. Proposed FDC-PLL Architecture

A. System Description

Fig. 3(a) shows a top-level block diagram of the proposed FDC-PLL architecture. It has the form of the generic digital PLL in Fig. 1(a), but the PEDC is implemented as a cascade of a second-order $\Delta\Sigma$-FDC and an accumulator, as in [21], [27], and [29], and a $1+z^{-1}$ block precedes the DLF. As explained shortly, the $\Delta\Sigma$-FDC is also modified relative to prior $\Delta\Sigma$-FDCs to incorporate the RFD technique described qualitatively in Section III.

As proven in Section IV-B and Appendix A, the $\Delta\Sigma$-FDC's output, $r[n]$, is a measure of the PLL's frequency error plus a component proportional to $d_e[n]-d_e[n-1]$. The accumulator following the $\Delta\Sigma$-FDC performs frequency-to-phase conversion, so its output, $p[n]$, contains terms proportional to the PLL's phase-error and $d_e[n]$. The subsequent $1+z^{-1}$ block prior to the DLF plays the role of the notch filter in Fig. 2 as it has a zero at $f_{\mathrm{ref}}/2$. The output of the DLF, $d[n]$, is latched into the DCO on each rising edge of $v_{\mathrm{ref}}(t)$ such that the DCO's instantaneous frequency during each time interval $t_n \le t < t_{n+1}$ is

$$f_{\mathrm{PLL}}(t) = f_c + K_{\mathrm{DCO}} d[n-1] + \psi_{\mathrm{DCO}}(t), \tag{3}$$

where f_c is the nominal center frequency of the DCO in Hz, K_{DCO} is the DCO gain in Hz, and $\psi_{\mathrm{DCO}}(t)$ is the DCO's instantaneous frequency error in Hz [21].

The proposed $\Delta\Sigma$-FDC shown in Fig. 3(b) is an extension of those presented in [21] and [22] that includes the proposed RFD technique. It also includes a modified gain calibration technique, as explained in Section IV-D, and achieves reduced PFD and ADC spans after locking relative to prior $\Delta\Sigma$-FDCs, as explained in Section IV-E. The $\Delta\Sigma$-FDC consists of an MMD, a PFD, a CP, an ADC, and a $\Delta\Sigma$-FDC digital block. The $1/(1+z^{-1})^2$ and $z^{-1}(2-z^{-2})$ transfer functions in the $\Delta\Sigma$-FDC's digital block play the roles of the $f_{\mathrm{ref}}/2$ resonator and $F(z)$ shown in Fig. 2, respectively.

The PFD and CP are comparable to those in analog PLLs [32]. Ideally, during the nth reference period, the PFD causes the CP to output a current pulse with a width of $|\tau_n - t_n|$, and nominal amplitudes of I_{CP} when $t_n < \tau_n$ and $-I_{\mathrm{CP}}$ when $\tau_n < t_n$, where τ_n as the time of the nth rising edge of $v_{\mathrm{div}}(t)$ and, as mentioned previously, t_n is the time of the nth rising edge of $v_{\mathrm{ref}}(t)$. The CP current is integrated by the capacitor, C, so each CP output pulse ideally changes the voltage across the capacitor by $I_{\mathrm{CP}}(\tau_n - t_n)/C$ volts.

As in prior $\Delta\Sigma$-FDCs, the integer-valued MMD control sequence, $v[n]$, is generated as part of the feedback loop within the $\Delta\Sigma$-FDC. It is a quantized version of $r_{\mathrm{F}}[n]$–α, where $r_{\mathrm{F}}[n]$ is the result of filtering the output of the $\Delta\Sigma$-FDC by $F(z)$ as shown in Fig. 3(b). The quantization is performed by the block labeled Q_{C}, which is an implementation of the second-order digital $\Delta\Sigma$ modulator shown in Fig. 3(c).

The $\Delta\Sigma$-FDC's B-bit ADC samples the CP output voltage at the rising edges of $v_{\mathrm{samp}}(t)$, which is a delayed version of $v_{\mathrm{ref}}(t)$. The ADC's output, $a[n]$, is interpreted as a fixed-point two's complement number with B–F and F integer and fractional bits, respectively. Each integer step of the ADC output corresponds to an ADC input step of Δ volts, so the ADC output sequence is interpreted as having a minimum step-size of $2^{-F}\Delta$ and an integer step-size of Δ.

The ADC's output is multiplied by the FDC's gain calibration loop output, $\hat{g}[n]$, and the sequence $e_{qc}[n$–$1]$ is added to the result to cancel quantization error introduced by Q_{C} that would otherwise degrade the PLL's phase noise. As explained in Section IV-E, an additional benefit of this quantization-error cancellation (QNC) technique is that it reduces the PFD and ADC spans compared to those in [21] when the PLL is locked.

The multiplication by $\hat{g}[n]$ corrects for gain error incurred in the $\Delta\Sigma$-FDC's forward path such as can result from deviations of I_{CP}, C, and Δ from their nominal values. As explained in [21], $\Delta\Sigma$-FDCs are not generally sensitive to such gain errors in terms of their input-output transfer functions. However, for a low-jitter PLL, the gain error must be low enough for QNC to sufficiently suppress $e_{qc}[n]$. For instance, in the PLL design example presented in Section V, the gain error must be 1% or less for the leaked component of $e_{qc}[n]$ at the PLL's output to be at least 10 dB lower than any other noise source so as to negligibly degrade the PLL's phase noise.

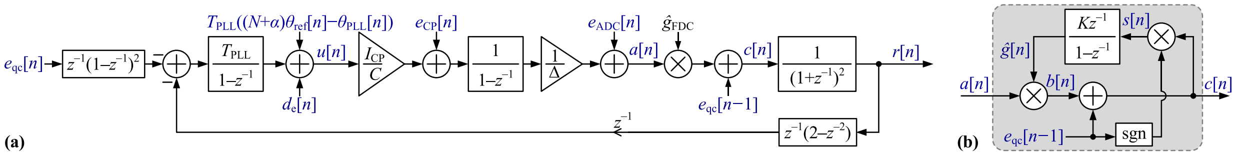

Fig. 4. (a) Linearized model of the proposed $\Delta\Sigma$-FDC where $\hat{g}[n]$ is approximated as a constant, $\hat{g}_{\mathrm{FDC}}$, and (b) FDC gain calibration behavioral model.

The FDC Gain Calibration block together with the $\hat{g}[n]$ multiplier and QNC adder implement a sign-LMS-like loop with loop gain K, reference sequence $e_{qc}[n]$, and error sequence $c[n]$. The FDC gain calibration technique can be qualitatively understood as follows. If $\hat{g}[n]$ is larger or smaller than its ideal value, then $e_{qc}[n]$ is not perfectly canceled at the QNC adder and $c[n]$ contains a term proportional to $-e_{qc}[n-1]$ or $e_{qc}[n-1]$, respectively. The FDC Gain Calibration block multiplies $c[n]$ by the sign of $e_{qc}[n-1]$ and accumulates the result, so the term proportional to $-e_{qc}[n-1]$ or $e_{qc}[n-1]$ in $c[n]$ respectively decreases or increases the accumulator output by $|e_{qc}[n-1]|$. As all other terms in $c[n]$ have zero-mean, this causes $\hat{g}[n]$ to be reduced or increased until $\hat{g}[n]$ reaches its ideal value aside from zero-mean fluctuations caused by noise. As with other LMS-like loops, reducing the magnitude of the loop gain, K, reduces the noise fluctuations at the expense of convergence rate.

B. $\Delta\Sigma$-FDC Linearized Model

As proven in Appendix A, the proposed $\Delta\Sigma$-FDC has a linearized model as shown in Fig. 4(a) for the case where the FDC gain calibration loop has converged such that $\hat{g}[n]$ can be approximated as a constant value, $\hat{g}_{\mathrm{FDC}}$. In Fig. 4(a), $\theta_{\mathrm{ref}}[n]$ is the reference phase noise, in cycles, at time t_n, $\theta_{\mathrm{PLL}}[n]$ is the PLL's phase noise, in cycles, at time τ_n, and $e_{\mathrm{CP}}[n]$ and $e_{\mathrm{ADC}}[n]$ represent error introduced by the CP and ADC, respectively.

The nominal values of I_{CP}, C, and Δ are chosen to satisfy $T_{\mathrm{PLL}}I_{\mathrm{CP}}/\mathrm{C}\Delta = 1$, because, as can be deduced from Fig. 4(a), this with $\hat{g}_{\mathrm{FDC}} = 1$ causes the contribution of $e_{\mathrm{ADC}}[n]$ to $r[n]$ to have the desired second-order highpass spectral shape, and causes the adder with output $c[n]$ to perfectly cancel $e_{qc}[n–1]$ in $r[n]$. In practice, however, I_{CP}, C, and Δ, deviate from their nominal values, so, as explained in Section IV-D, the output of the FDC's gain calibration loop converges such that the $\Delta\Sigma$-FDC's forward path gain is unity, i.e.,

$$T_{\mathrm{PLL}}\frac{I_{\mathrm{CP}}}{C}\frac{1}{\Delta}\hat{g}_{\mathrm{FDC}} = 1. \tag{4}$$

Fig. 4(a) and (4) imply that the $\Delta\Sigma$-FDC's output is

$$\begin{aligned} r[n] = &-(\theta_{\mathrm{PLL}}[n]-\theta_{\mathrm{PLL}}[n-1]) + e_{\mathrm{FDC}}[n] - e_{\mathrm{FDC}}[n-1] \\ &+ f_{\mathrm{PLL}}\left(d_e[n]-d_e[n-1]\right), \end{aligned} \tag{5}$$

where

$$\begin{aligned} e_{\mathrm{FDC}}[n] = (N+\alpha)\theta_{\mathrm{ref}}[n] + \frac{\hat{g}_{\mathrm{FDC}}}{\Delta}e_{\mathrm{CP}}[n] \\ + \hat{g}_{\mathrm{FDC}}\left(e_{\mathrm{ADC}}[n]-e_{\mathrm{ADC}}[n-1]\right) \end{aligned} \tag{6}$$

represents error introduced by the reference signal and the $\Delta\Sigma$-FDC. As shown in Fig. 3(a), $r[n]$ is accumulated to generate $p[n]$, so (5) implies

$$p[n] = -\theta_{\mathrm{PLL}}[n] + e_{\mathrm{FDC}}[n] + f_{\mathrm{PLL}}d_e[n]. \tag{7}$$

As in prior CP-based FDC-PLLs, $p[n]$ contains a component proportional to the PLL's phase error, $-\theta_{\mathrm{PLL}}[n]$, and a first-order highpass shaped component corresponding to the ADC's error. The error introduced by the CP and reference oscillator circuitry (including the FD) are unshaped, and the $d_e[n]$ component in $p[n]$ is canceled prior to the DLF by the 1 $+ z^{-1}$ block in Fig. 3.

It follows from Fig. 4(a) that the transfer function from $d_e[n]$ to $u[n]$, where $u[n]$ is defined as $\tau_n - t_n$, is

$$(1-z^{-1})^2(1+z^{-1})^2. \tag{8}$$

Hence, the transfer function from $d_e[n]$ to $u[n]$ has a pair of zeros at $z = -1$, so $d_e[n]$, which is proportional to $(-1)^n$, does not appear at the PFD's output, which proves the corresponding result presented and explained qualitatively in Section III.

Although the FDC gain calibration loop provides the benefit outlined in Section IV-A, it is not necessary for the RFD technique to function properly. If the $\Delta\Sigma$-FDC's loop gain is left uncalibrated such that

$$T_{\mathrm{PLL}}\frac{I_{\mathrm{CP}}}{C}\frac{1}{\Delta}\hat{g}_{\mathrm{FDC}} = 1 + g, \tag{9}$$

where g is the gain-error, it follows from Fig. 4(a) and (4) that the transfer function from $d_e[n]$ to $u[n]$ becomes:

$$\frac{(1-z^{-1})^2(1+z^{-1})^2}{1+2gz^{-2}-gz^{-4}}. \tag{10}$$

Therefore, the transfer function has a pair of zeros at $z = -1$ like (8), so $d_e[n]$ does not appear at the PFD's output even when the $\Delta\Sigma$-FDC has a gain error.

The gain error does introduce poles, though, which slightly increases the initial $d_e[n]$ settling time. After a cold-start, the PLL's dynamics are nonlinear so the linearized model of Fig. 4(a) does not apply and the $\Delta\Sigma$-FDC's gain error has little effect. Once all analog nodes and digital registers stop clipping and remain within their linear operating regions, the $\Delta\Sigma$-FDC linearized model becomes applicable and (10) can be used to evaluate the settling of the $d_e[n]$ component of $u[n]$. Without loss of generality, suppose the $\Delta\Sigma$-FDC linearized model becomes applicable at $n = 0$. Convolving $d_e[n]$ with the inverse z-transform of (10) shows that the component of $u[n]$ corresponding to $d_e[n]$ is

$$h_{de}[n] = \begin{cases} \dfrac{\Delta T}{p_1-p_2}\left\{(p_1-1)\,p_1^{n/2}-(p_2-1)\,p_2^{n/2}\right\}, & \text{if } n \text{ is even,} \\ -h_{de}[n-1], & \text{otherwise,} \end{cases} \tag{11}$$

for $n = 0, 1, 2, \ldots,$ and

$$p_{1,2} = -g \pm \sqrt{g^2+g}. \tag{12}$$

For gain errors bounded in magnitude by 20% (i.e., $|g| < 0.2$), (11) implies that $h_{de}[n]$ decays with time and its magnitude becomes equivalent to a duty-cycle error less than 0.005% in at most 22 reference cycles. For the PLL parameters used in the design presented in Section V, this level corresponds to an $f_{\text{ref}}/2$ spur less than -80 dBc and adds less than 0.5% to the CP and ADC nominal spans. Even for a gain error of 30%, which is far larger than would be expected in practice, a residual duty-cycle error of 0.005% is reached in 102 reference cycles. Provided $-1 < g < 1/3$, which corresponds to an uncalibrated $\Delta\Sigma$-FDC loop gain error between -100% and 33%, a range far more than expected in practice, (11) and (12) imply that $h_{de}[n]$ converges to zero as $n \to \infty$, independent of other PLL parameters.

C. PLL Linearized Phase Noise Model

When the PLL is locked, its ideal output frequency is $(N+\alpha) f_{\text{ref}}$, so it follows from (3) that the DCO input sequence can be written as $d[n] = [(N+\alpha) f_{\text{ref}} - f_{\text{c}}]/K_{\text{DCO}} + f[n]$, where $f[n]$ is the component of $d[n]$ arising from noise. This with (3) implies that during each time interval $t_n \le t < t_{n+1}$, the PLL's instantaneous frequency error in Hz can be expressed as

$$\psi_{\text{PLL}}(t) = K_{\text{DCO}} f[n-1] + \psi_{\text{DCO}}(t). \tag{13}$$

Phase is the integral of frequency, so integrating (13) from t_0 to t, for $t_n \le t < t_{n+1}$ with $n \ge 0$ gives

$$\begin{aligned}\theta_{\text{PLL}}(t) &= \theta_{\text{DCO}}(t) + K_{\text{DCO}}\,(t - t_n)\, f[n-1] \\ &\quad + K_{\text{DCO}} \sum_{k=-1}^{n-2} (t_{k+2} - t_{k+1})\, f[k],\end{aligned} \tag{14}$$

where it has been assumed without loss of generality that $\theta_{\text{PLL}}(t_0) = 0$ and $\theta_{\text{DCO}}(t_0) = 0$. Typical reference oscillators have high spectral purity, so (1) implies

$$t_{k+2} - t_{k+1} \cong T_{\text{ref}} + \Delta T(-1)^n. \tag{15}$$

Substituting this into (14) gives

$$\begin{aligned}\theta_{\text{PLL}}(t) &\cong \theta_{\text{DCO}}(t) + K_{\text{DCO}}\,(t - t_n)\, f[n-1] \\ &\quad + \theta_{\text{loop}}[n] + K_{\text{DCO}} \Delta T \sum_{k=-1}^{n-2} (-1)^k f[k],\end{aligned} \tag{16}$$

where

$$\theta_{\text{loop}}[n] = K_{\text{DCO}} T_{\text{ref}} \sum_{k=-1}^{n-2} f[k]. \tag{17}$$

The bandwidth of a PLL is generally much smaller than f_{ref} and $|\tau_n - t_n|$ is less than a few DCO periods, so

$$\theta_{\text{PLL}}(\tau_n) \cong \theta_{\text{PLL}}(t_n). \tag{18}$$

Therefore, (16) implies

$$\theta_{\text{PLL}}[n] \cong \theta_{\text{DCO}}[n] + \theta_{\text{loop}}[n] + K_{\text{DCO}} \Delta T \sum_{k=-1}^{n-2} (-1)^k f[k], \tag{19}$$

where $\theta_{\text{DCO}}[n] = \theta_{\text{DCO}}(t_n)$ and $\theta_{\text{PLL}}[n] = \theta_{\text{PLL}}(t_n)$.

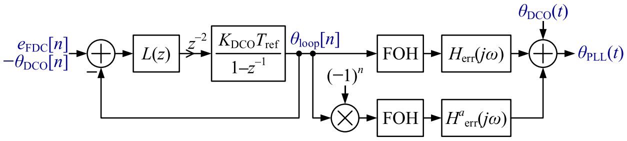
$e_{\text{FDC}}[n]$
$-\theta_{\text{DCO}}[n]$
$-$
$L(z)$
z^{-2}
$\frac{K_{\text{DCO}}T_{\text{ref}}}{1-z^{-1}}$
$\theta_{\text{loop}}[n]$
$(-1)^n$
FOH
$H_{\text{err}}(j\omega)$
FOH
$H^a_{\text{err}}(j\omega)$
$\theta_{\text{DCO}}(t)$
$\theta_{\text{PLL}}(t)$

Fig. 5. PLL LTV phase noise model.

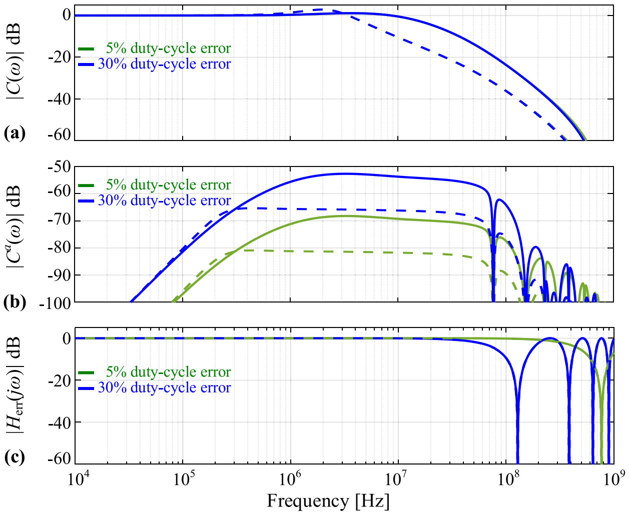

Fig. 6. Magnitude responses of (a) C(ω), (b) $C^a(\omega)$, and (c) $H_{\text{err}}(j\omega)$. Dashed (solid) lines correspond a PLL bandwidth of 280 kHz (1.3 MHz). The 5% and 30% duty-cycle error curves coincide in (a).

Equations (16) and (19) are linear equations, but they are not time-invariant when the crystal oscillator has a non-50% duty cycle because of the $(-1)^k f[n]$ terms. As shown in Appendix B, they give rise to the linear time-varying (LTV) PLL phase noise model shown in Fig. 5, where $L(z)$ is the transfer function from $p[n]$ to $d[n]$ in Fig. 3(a), each FOH block is a first-order hold (FOH) interpolator, and the expressions for $H_{\text{err}}(j\omega)$ and $H^a_{\text{err}}(j\omega)$ are given in Appendix B. If the duty cycle of the crystal oscillator were exactly 50%, then $H_{\text{err}}(j\omega)$ and $H^a_{\text{err}}(j\omega)$ would equal 1 and 0, respectively, and the phase noise model would reduce to the LTI phase noise model presented in [21] despite the modifications of the PLL presented in this paper.

The output of each FOH interpolator is a continuous-time waveform given by

$$\sum_{n=0}^{\infty} s[n] h_{\text{tri}}\,(t - nT_{\text{ref}} - t_0), \tag{20}$$

where

$$h_{\text{tri}}(t) = \begin{cases} 1 - |t|/T_{\text{ref}}, & \text{if } |t| < T_{\text{ref}}, \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

and $s[n]$ is a dummy variable which represents the FOH interpolator's input sequence. The continuous-time Fourier transform (CTFT) of (20) is

$$T_{\text{ref}} H\,(\omega)\, S\left(e^{j\omega T_{\text{ref}}}\right), \text{where } H(\omega) = \left[\frac{\sin\left(\omega T_{\text{ref}}/2\right)}{\omega T_{\text{ref}}/2}\right]^2, \tag{22}$$

and $S(e^{j\omega T\text{ref}})$ is the discrete-time Fourier transform (DTFT) of $s[n]$ [34]. Therefore, the CTFT of the sum of $H_{\text{err}}(j\omega)$ and

TABLE I

CONTRIBUTION OF DIFFERENT NOISE SOURCES[†] TO THE PLL'S OUTPUT

Noise Source	Contribution to the PSD of $\theta_{\mathrm{PLL}}(t)$ in dBc/Hz
Reference Source	$4\pi^2\left\|(N+\alpha)G(e^{j\omega T_{\mathrm{ref}}})H(\omega)\right\|^2 S_{\mathrm{ref}}(\omega)$
CP	$4\pi^2 T_{\mathrm{ref}}\left\|(\hat{g}_{\mathrm{FDC}}/\Delta)G(e^{j\omega T_{\mathrm{ref}}})H(\omega)\right\|^2 S_{\mathrm{CP}}(e^{j\omega T_{\mathrm{ref}}})$
ADC	$4\pi^2(T_{\mathrm{ref}}/3)\left\|2^{-F}\hat{g}_{\mathrm{FDC}}\sin(\omega T_{\mathrm{ref}}/2)G(e^{j\omega T_{\mathrm{ref}}})H(\omega)\right\|^2$
DCO	$4\pi^2\left\|1-G(e^{j\omega T_{\mathrm{ref}}})H(\omega)\right\|^2 S_{\mathrm{DCO}}(\omega)$

[†] $S_{\mathrm{ref}}(\omega)$ and $S_{\mathrm{DCO}}(\omega)$ are the two-sided phase-noise PSDs of the reference and DCO, respectively, in cycles squared per Hz, and $S_{\mathrm{CP}}(e^{j\omega T_{\mathrm{ref}}})$ is the two-sided discrete-time PSD of $e_{\mathrm{CP}}[n]$ in dBV/Hz.

$H_{\mathrm{err}}^a(j\omega)$ outputs in Fig. 5 is

$$X\left(e^{j\omega T_{\mathrm{ref}}}\right)C(\omega)+X\left(e^{j(\omega T_{\mathrm{ref}}-\pi)}\right)C^a(\omega), \tag{23}$$

where $X(z)$ is the z-transform of $x[n]=-\theta_{\mathrm{DCO}}[n]+e_{\mathrm{FDC}}[n]$,

$$C(\omega)=T_{\mathrm{ref}}G\left(e^{j\omega T_{\mathrm{ref}}}\right)H(\omega)H_{\mathrm{err}}(j\omega), \tag{24}$$

$$C^a(\omega)=T_{\mathrm{ref}}G\left(e^{j(\omega T_{\mathrm{ref}}-\pi)}\right)H(\omega)H_{\mathrm{err}}^a(j\omega), \tag{25}$$

$$G(z)=\frac{T(z)}{1+T(z)},\text{ and } T(z)=z^{-2}L(z)\frac{K_{\mathrm{DCO}}T_{\mathrm{ref}}}{1-z^{-1}}. \tag{26}$$

Figures 6(a) and 6(b) show the magnitudes in dB of $C(\omega)$ and $C^a(\omega)$ versus frequency in Hz for the PLL design example presented in Section V with loop bandwidths of 280 kHz and 1.3 MHz. The results imply that the $H_{\mathrm{err}}^a(j\omega)$ path in the phase noise model has little effect for the design example. The reason is that $H_{\mathrm{err}}^a(j\omega)$ is highly attenuated over the bandwidth of its input signal. Furthermore, as illustrated in Fig. 6(c), $H_{\mathrm{err}}(j\omega)\cong 1$ to a high degree of accuracy for the design example up to frequencies well past the PLL's bandwidth. Consequently, the phase noise model is well-approximated for the design example by the system of Fig. 5 with $H_{\mathrm{err}}(j\omega)=1$ and the $H_{\mathrm{err}}^a(j\omega)$ path neglected.

To the extent that the PLL's noise sources can be modeled as uncorrelated zero-mean wide-sense stationary random processes, it follows that the PSD of $\theta_{\mathrm{PLL}}(t)$ is the superposition of the PSDs of the individual sources. Hence, the two-sided PSD of $\theta_{\mathrm{PLL}}(t)$ is:

$$\begin{aligned}S_{\theta_{\mathrm{PLL}}}(f)=\ &S_{\theta_{\mathrm{PLL}}}(f)\big|_{\mathrm{ref}}+S_{\theta_{\mathrm{PLL}}}(f)\big|_{\mathrm{CP}}\\&+S_{\theta_{\mathrm{PLL}}}(f)\big|_{\mathrm{ADC}}+S_{\theta_{\mathrm{PLL}}}(f)\big|_{\mathrm{DCO}},\end{aligned} \tag{27}$$

where the terms on the right hand side of (27) are the two-sided PSDs of the reference source, CP, ADC, and DCO noise contributions to the PLL's output. Following the same reasoning in [21], Table I summarizes the contribution of the different sources above for the PLL phase noise model with $H_{\mathrm{err}}(j\omega)=1$ and $H_{\mathrm{err}}^a(j\omega)=0$.

D. ΔΣ-FDC Gain Calibration Details

Fig. 4(b) shows the portion of the ΔΣ-FDC's behavioral model connecting the ADC's output, $a[n]$, and the resonator's input, $c[n]$, with the FDC gain calibration loop details added. As explained in Sections IV-A and IV-B, the objective of the FDC gain calibration loop is to cause $\hat{g}[n]$ to converge to a constant, $\hat{g}_{\mathrm{FDC}}$, aside from zero-mean error, that satisfies (4). Therefore, (4) implies that $\hat{g}[n]$ can be written as

$$\hat{g}[n]=\frac{1}{A}+\varepsilon[n],\text{ with } A=T_{\mathrm{PLL}}\frac{I_{\mathrm{CP}}}{C}\frac{1}{\Delta}, \tag{28}$$

where $\varepsilon[n]$ is the FDC gain calibration error.

It follows from Fig. 4(b) that $\hat{g}[n]=\hat{g}[n-1]+Ks[n-1]$, so (28) implies

$$\varepsilon[n]=\varepsilon[n-1]+Ks[n-1]. \tag{29}$$

This and the linearity of the expectation operator implies

$$\bar{\varepsilon}[n]=\bar{\varepsilon}[n-1]+K\bar{s}[n-1], \tag{30}$$

where $\bar{s}[n]$ and $\bar{\varepsilon}[n]$ are the expectations of $s[n]$ and $\varepsilon[n]$, respectively. To the extent that the gain-error does not deteriorate the self-dithering property of delta-sigma modulators to a great extent, $e_{qc}[n]$ is well-approximated as independent of all other random variables and uniformly distributed between –1/2 and 1/2. A nearly identical analysis to that presented in [35] shows that

$$\bar{s}[n]=-A\bar{\varepsilon}[n]\cdot E\left\{\left|e_{qc}[n-1]\right|\right\}. \tag{31}$$

The uniform distribution of $e_{qc}[n]$ between –1/2 and 1/2 implies that $|e_{qc}[n]|$ is uniformly distributed between 0 and 1/2, so it follows that

$$\bar{\varepsilon}[n]=\bar{\varepsilon}[n-1]\left(1-\frac{AK}{4}\right). \tag{32}$$

Recursively substituting (32) in itself yields

$$\bar{\varepsilon}[n]\cong\varepsilon[0]\left(1-\frac{AK}{4}\right)^n, \tag{33}$$

Hence, $\bar{\varepsilon}[n]$ converges to zero, so $\hat{g}[n]$ converges to $\hat{g}_{\mathrm{FDC}}$ aside from zero-mean error, as $n\to\infty$, provided $0<K<4/A$.

As with any LMS-like loop, the FDC gain calibration technique is subject to a convergence speed versus accuracy trade-off; increasing K increases the convergence rate, but it also increases the power of $\varepsilon[n]$. However, as implied by the signal processing operations shown in Fig. 3, the contribution of $\varepsilon[n]$ to the PEDC output, $p[n]$, is $\varepsilon[n]-\varepsilon[n-1]$, so the FDC gain calibration error is subjected to first-order highpass shaping. This reduces its contribution to the PLL phase noise, thereby relaxing the convergence rate versus accuracy trade-off relative to the FDC gain calibration technique presented in [35].

E. Additional ΔΣ-FDC Properties

In prior CP-based ΔΣ-FDCs, such as those presented in [21] and [22], the $v[n]$ input to the MMD is $2y[n]-y[n-1]$, where $y[n]$ denotes the integer portion of the output of either the ADC or, if FDC gain calibration is implemented, the FDC gain calibration multiplier following the ADC. Ideally, the time at which the CP output pulse terminates during the nth reference period is the larger of t_n and τ_n, but, in practice, the CP takes time to settle and the ADC then takes time to perform a conversion, so $y[n]$ is not available until well after the start of the reference period. Once $y[n]$ is available, $v[n]$ must be computed and loaded into the MMD early enough that the time of the MMD's next rising output edge, τ_{n+1}, is $N - v[n]$ DCO periods after that of the MMD's prior rising output edge, τ_n. These timing constraints can be tight, especially for high reference frequencies.

In contrast, the proposed $\Delta\Sigma$-FDC (Fig. 3(b)) has an extra reference period delay between the ADC output and $v[n]$ relative to prior CP-based $\Delta\Sigma$-FDCs because of the delay through $F(z)$. Although the QNC adder, resonator, and $F(z)$ block represent more digital operations than are needed to just compute $2y[n]-y[n-1]$ in prior $\Delta\Sigma$-FDCs, they can be performed in a small fraction of a reference period in typical CMOS technology, so the timing constraints of the proposed $\Delta\Sigma$-FDC are significantly more relaxed than those of prior CP-based $\Delta\Sigma$-FDCs.

However, the extra feedback delay through $F(z)$ causes the proposed $\Delta\Sigma$-FDC to have a smaller maximum input frequency range than that of prior CP-based $\Delta\Sigma$-FDCs with the same ADC. This is because the deviation of the average DCO frequency per reference period relative to its ideal value of $(N + \alpha)f_{\text{ref}}$ affects the ADC output through a transfer function of $(1+z^{-1})^2/f_{\text{ref}}$ in the proposed $\Delta\Sigma$-FDC but of just $1/f_{\text{ref}}$ in prior CP-based FDCs. The issue mainly affects the PLL during the locking process because, once the PLL's output frequency has converged to $(N+\alpha)f_{\text{ref}}$, the ADC output sequence is dominated by coarse quantization error from Q_{C}, the span of which is not affected by the extra delay in $F(z)$. Therefore, the required ADC span is higher while the PLL locks than it is once the PLL finishes locking. In the design example presented in the next section, this issue is addressed via a SAR ADC that provides 7 bits of resolution while the PLL locks and then reduces its resolution to 6 bits to save power once the PLL has locked.

In contrast, three other modifications of the proposed $\Delta\Sigma$-FDC—performing QNC within the $\Delta\Sigma$-FDC, and subtracting α and performing the Q_{C} quantization within the $\Delta\Sigma$-FDC's feedback path—act to relax the required ADC input range compared to prior CP-based $\Delta\Sigma$-FDCs. They do so by reducing the contribution of coarse quantization error, $e_{qc}[n]$, to the ADC's input, which makes the biggest difference after the PLL locks when the ADC span is dominated by $e_{qc}[n]$. Fig. 3(c) implies that $|e_{qc}[n]| \leq 1/2$, and Fig. 4(a) implies that the transfer function from $e_{qc}[n]$ to the ADC's input is nominally z^{-1}, so the ADC in the proposed $\Delta\Sigma$-FDC requires only one integer ADC step to accommodate coarse quantization error. In contrast, each of the $\Delta\Sigma$-FDC's presented in [21] and [22] require three integer ADC steps to accommodate coarse quantization error. For example, the $\Delta\Sigma$-FDCs in [21] and [22] each require a total input range of 4 integer ADC steps to achieve a frequency acquisition range of f_{ref}, whereas the proposed $\Delta\Sigma$-FDC requires total input ranges of 5 and 1.25 integer ADC steps before and after the PLL locks, respectively.

Similar reasoning shows that the modifications also reduce the average duration of the CP output pulses relative to prior CP-based $\Delta\Sigma$-FDCs. In the proposed $\Delta\Sigma$-FDC, the average CP pulse duration is reduced by a factor of 2.6 relative to prior CP-based $\Delta\Sigma$-FDCs. This corresponds to a reduction in the CP's thermal and flicker noise contributions to the PLL's phase noise by 4.1 dB and 8.2 dB, respectively [32]. The thermal noise reduction alone reduces the CP's contribution to the PLL's jitter by 37% for a given PLL bandwidth and output frequency.

TABLE II
PLL DESIGN PARAMETERS USED FOR THE BEHAVIORAL SIMULATIONS

Design Parameters		Value	
Crystal Oscillator	Frequency, f_{crystal}	76.8 MHz	
	Phase noise[(1)]	−160 and −163 dBc/Hz	
Frequency Doubler	Added phase noise	Negligible	
	T_{DL}	3.25 ns	
Charge Pump	Nominal current, I_{CP}	1 mA	
	Nominal capacitance, C	1 pF	
	Noise[(2)]	−138 and −148 dBV/Hz	
ADC	Number of bits (B, F)	7, 5	
	Coarse step-size, Δ	100 mV	
DCO	DCO gain, K_{DCO}	150 kHz	
	Phase noise[(3)]	−122, −120, and −150 dBc/Hz	
DLF	Proportional gain, K_P	5	20
	Integral gain, K_I	0.0390625	0.15625
	IIR pole, λ	0.75	0.75
FDC Digital	FDC gain calibration, K	0.015625	
PLL Settings	Integer multiplier, N	65	
	Fractional multiplier, α	0.0007848739624	
	Output frequency, f_{PLL}	9.984 GHz	
	Loop bandwidth	280 kHz	1.3 MHz
	RMS Jitter	113 fs	75 fs

[1] $1/f$ and white phase noise components at 10 kHz offset.
[2] $1/f$ and white discrete-time PSD of $e_{\text{CP}}[n]$ at 10 kHz offset.
[3] $1/f^3$, $1/f^2$ and white phase noise components at 1 MHz offset.

V. PLL DESIGN EXAMPLE

The design example of the proposed PLL presented in this section has $f_{\text{crystal}} = 76.8$ MHz, $f_{\text{PLL}} = 9.984$ GHz, and an RMS output jitter of 75 fs. Table II presents the relevant design parameters and noise contributions.[2] The noise contributions and other nonideal circuit behavior, such as CP nonlinearity and component mismatches, were determined via Cadence Spectre simulations of FD, CP, and DCO circuits implemented in the Global Foundries 22-nm CMOS 22FDX process. Parameters that describe the nonideal circuit behavior were extracted from the transistor-level simulations and back-annotated into a custom, C-language, event-driven, bit-exact, behavioral simulator along the lines of those described in [27], [30], [35], and [38]. The events modeled by the behavioral simulator are the rising and falling edges of the crystal oscillator, FD, MMD, PFD and DCO.

To capture the CP's nonlinear behavior, the CP's transistor-level simulated output voltages versus the expected range of PFD output pulse-widths in increments of 1 ps were back-annotated into a look-up table (LUT), which the behavioral simulator uses to calculate each CP output voltage via piecewise linear interpolation between adjacent LUT points. The transistor-level simulation testbench included realistic models for the supply network (including the supply source impedance, routing traces, bond-wires, and decoupling capac-

[2] Table II does not include PFD or MMD circuit noise contributions as they are negligible relative to those of the PLL's other noise sources by design.

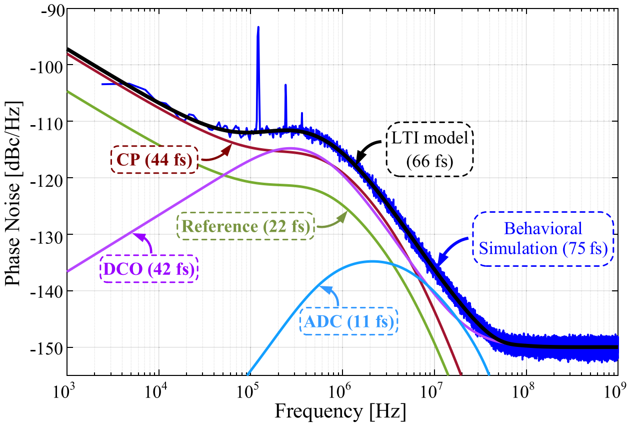

Fig. 7. PLL phase noise power spectra with 8% crystal oscillator duty-cycle error and RFD technique enabled.

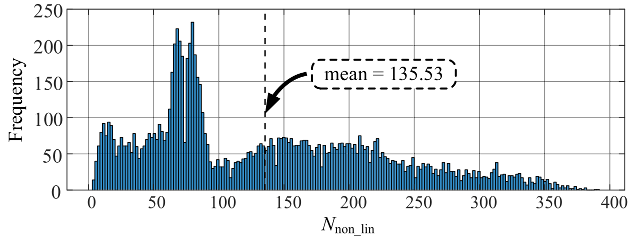

Fig. 8. Histogram of $N_{\text{non-lin}}$ for 10000 PLL runs with random initial conditions.

TABLE III

DUTY-CYCLE ERROR CANCELLATION TECHNIQUES COMPARISON

	[11]	[12]	[16]	[18]	This Work
Architecture	Sampling PLL	Sampling PLL	MDLL	MDLL	FDC-PLL
Estimation Technique	LMS based	LMS based	LMS based	LMS based	Resonator Based ΔΣ-FDC
Cancellation Path	MMD	MMD	Reference	Reference	MMD
Duty-cycle Error	6.00 %	7.00%	0.25%	0.50%	−10% to 10%
Convergence Time (1)	2000	3800	6000	2500	412(2)

(1) In number of reference cycles.

(2) Worst-case convergence time across 10000 PLL runs with random duty-cycle error and PLL initial conditions.

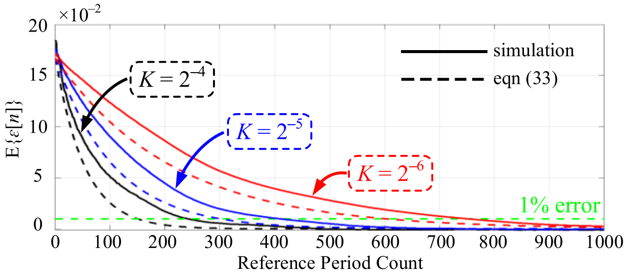

Fig. 9. FDC gain calibration error sequence, $E\{\varepsilon[n]\}$.

itors) to capture the effect of the PFD transitions on the power supply shared with the CP.

The ΔΣ-FDC's CP circuit uses a current-steering topology with cascode nMOS and pMOS current sources and incorporates an offset-current linearization technique [32], [36]. Its pMOS current source plays the role of the offset current, thereby drawing a fixed amount of charge from the supply each reference cycle, which reduces potential coupling through the supply network. Fast CP settling, enabled by the CP's current steering topology, allowed for the use of a relatively short 400 ps offset current pulse which helped reduce the CP noise. Simulations predict that 400 ps pulse width is sufficient for supply ripples to not significantly degrade the linearity of the PFD/CP combination.

Simulations suggest that the CP's average leakage current over a reference cycle is approximately −85 nA. However, the PLL is not sensitive to CP leakage current in general. As explained in Section II, the crystal oscillator's duty cycle error causes the time between the reference edges to alternate between two values. Therefore, CP leakage current can be well modeled as causing the voltage sampled by the ADC to contain an error sequence which takes on one of two different values depending on whether the reference period index is even or odd. As this error has the same effect on the PLL as an alternating sequence injected at the PFD/CP interface, it is canceled by the proposed RFD technique in the same manner that the crystal oscillator duty-cycle error is canceled.

The ΔΣ-FDC's ADC is a 7-bit asynchronous SAR ADC with 2 integer bits and 5 factional bits. The corresponding frequency acquisition range of the ΔΣ-FDC is 30 MHz, which transistor-level simulations suggest is more than sufficient to cover temperature and flicker-noise induced DCO frequency drifts. The standard deviation of the unit capacitor random mismatch is set to two percent in the behavioral model, based on Monte Carlo simulations in Spectre. After locking, the input range of the ADC is such that only 1 integer bit is required as explained in Section IV-E, so the ADC resolution is reduced to 6 bits to save power. Behavioral simulations suggest that a comparator metastability rate of 0.01% is sufficient to not significantly degrade the PLL's performance, which is not difficult to satisfy in practice [37].

All digital operations performed by the behavioral simulator are bit-exact. The bus widths of α, $e_{qc}[n]$, $r[n]$, $v[n]$, $p[n]$, and $d[n]$ are 18, 18, 20, 7, 19, and 16 bits respectively. The FDC gain calibration accumulator has a bus width of 25 bits, which is truncated to 15-bits to generate $\hat{g}[n]$. The DLF consists of a conventional proportional-integral stage, and one single-pole IIR stage [19], [29], [38]. The transfer function from $p[n]$ to $d[n]$ is given by:

$$L(z) = \left(1 + z^{-1}\right)\left(K_P + K_I \frac{z^{-1}}{1 - z^{-1}}\right)\left(\frac{1 - \lambda}{1 - \lambda z^{-1}}\right), \quad (34)$$

where K_P and K_I are the proportional and integral path gains, respectively, λ is the pole of the IIR stage, and the $1+z^{-1}$ factor represents the $f_{\text{ref}}/2$ notch filter.

Fig. 7 shows the PLL's various phase noise spectra for a scenario where the crystal oscillator's duty-cycle error was set to 8% and the proposed RFD technique enabled. The fractional spurs at integer multiples of 120.56 kHz are from CP nonlinearity. Their total power is just under −56 dBc and their presence increases the jitter from 66 fs to 75 fs. Additional

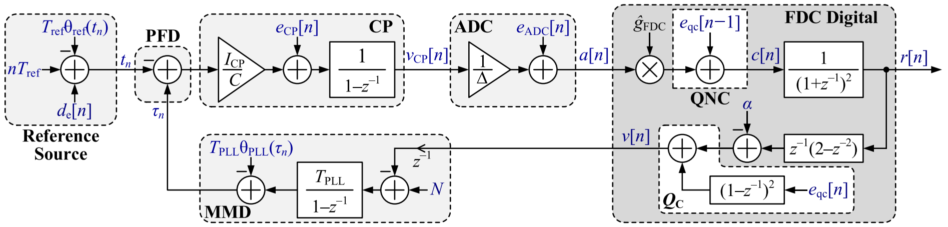

Fig. 10. Linearized behavioral model of the proposed $\Delta\Sigma$-FDC where $\hat{g}[n] = \hat{g}_{\text{FDC}}$ is approximated as constant.

simulations run by the authors indicate that in the absence of the FD, the PLL's total jitter would have been 90 fs.

The convergence time of the proposed RFD technique, in reference cycles, is defined as $N_{\text{conv}} = N_{\text{non-lin}} + N_{\text{lin}}$. The first term, $N_{\text{non-lin}}$, is the number of reference cycles after the DCO's coarse frequency is set to an initial value between $(N+\alpha)f_{\text{ref}} - 15\cdot 10^6$ Hz and $(N+\alpha)f_{\text{ref}} + 15\cdot 10^6$ Hz that are required for the PLL to settle to the point where the $\Delta\Sigma$-FDC's linearized model (Fig 4) holds. The second term, N_{lin}, is the number of reference cycles required for $h_{de}[n]$ in (11) to decay to a magnitude which corresponds to a duty-cycle error less than 0.005%. Plotting (11) shows that N_{lin} increases with the $\Delta\Sigma$-FDC gain-error and is equal to 22 for the worst-case $\Delta\Sigma$-FDC gain error of 20%. For simplicity, the worst-case value of $N_{\text{lin}} = 22$ is assumed in the following.

Fig. 8 shows a histogram of $N_{\text{non-lin}}$ for 10,000 PLL runs. For each run, the PLL was initialized to have a random crystal oscillator initial phase, a crystal-oscillator duty-cycle between 40% and 60%, an uncalibrated $\Delta\Sigma$-FDC gain-error between -20% and 20%, and an initial DCO frequency error between -15 MHz and 15 MHz. Fig. 8 and $N_{\text{lin}} = 22$ imply that the maximum and average convergence times are 412 and 157 reference cycles, respectively.

Table III compares the worst-case convergence time of the design example to the published convergence times of the published prior. It shows that the design example has a significantly lower convergence time than the published prior art even with higher crystal oscillator duty cycle errors and lower jitter. The closest competitor is [11], but its reported convergence time is not directly comparable to that of the design example. As described in [11], behavioral simulation results indicate that the LMS duty-cycle calibration loop converges to a duty cycle error of 6% in 2000 reference cycles, after which the LMS loop bandwidth is reduced followed by an unspecified additional convergence time to prevent the LMS loop noise from degrading the PLL's phase noise. As it was not specified in [11], the corresponding value in Table III does not include the extra required convergence time. Therefore, the convergence time of the design example is pessimistically at least 5× faster than that the published prior art, but the authors estimate that it is at least 10× faster than that of the prior art with comparable initial duty-cycle errors, comparable PLL jitter, and negligible added LMS loop noise.

To evaluate the convergence speed of the FDC gain calibration loop and compare it with that predicted by (33), $\bar{\varepsilon}[n]$ was simulated by averaging $\varepsilon[n]$ over 1000 PLL simulation runs. As the derivation which led to (33) assumes that the PLL is locked, the FDC gain calibration loop was enabled after the PLL locked to provide a meaningful comparison. Fig. 9 shows the simulated and calculated values of $\bar{\varepsilon}[n]$ for an initial $\Delta\Sigma$-FDC gain error of 20% and different values of the LMS loop gain, K. It shows that the simulated and calculated values of $\bar{\varepsilon}[n]$ are within 2% of each other and decay to within 1% of the ideal gain, $1/A$, in less than 750 reference cycles for the value of $K = 2^{-6}$ used in the design example. The 1% error threshold ensures that the leaked $e_{qc}[n]$ component contribution to the PLL's phase noise is 10 dB less than all other noise sources. Behavioral simulations performed by the authors show that similar convergence results are achieved when the FDC gain calibration loop is enabled before the PLL loop starts locking.

VI. Conclusion

An entirely digital technique has been presented which makes a PLL immune to alternating reference edge timing errors, thereby enabling the use of a reference frequency doubler to reduce the PLL's jitter without the need for duty-cycle error calibration. The technique is free of convergence bias and is not subject to the speed-accuracy tradeoff inherent to prior-art duty-cycle error calibration solutions. A $\Delta\Sigma$-FDC based PLL design example enabled by the technique has been presented with at least 10× faster convergence than that of comparable prior art solutions. The presented $\Delta\Sigma$-FDC also includes new techniques which significantly reduce its PFD and ADC spans after locking, and which can be applied to prior $\Delta\Sigma$-FDCs independently from the duty-cycle error immunity technique. A detailed theoretical analysis of the proposed PLL has been presented, including the development of a linear time-varying phase noise model, the results of which closely match simulation results.

Appendix A
$\Delta\Sigma$-FDC Linearized Model Derivation

As explained in Section IV-A, the CP current is integrated by the capacitor, C, and changes the voltage across the capacitor by $I_{\text{CP}}(\tau_n - t_n)/C$ volts during the nth reference period. Hence, the CP output voltage is

$$v_{\text{CP}}[n] = v_{\text{CP}}[n-1] + \frac{I_{\text{CP}}}{C}(\tau_n - t_n) + e_{\text{CP}}[n], \quad (35)$$

where τ_n and t_n are the times of the nth rising edges of $v_{\text{div}}(t)$ and $v_{\text{ref}}(t)$, respectively, and $e_{\text{CP}}[n]$ is the noise and distortion added by the CP. As proven in [21],

$$\tau_n = \tau_0 - T_{\text{PLL}}\theta_{\text{PLL}}[n] + T_{\text{PLL}}\sum_{k=1}^{n}(N - v[k-1]), \quad (36)$$

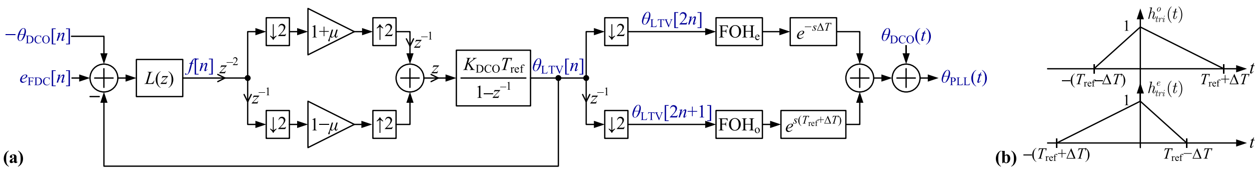

Fig. 11. (a) PLL's LTV model and (b) the FOH$_e$ and FOH$_o$ interpolation functions, $h^e_{\text{tri}}(t)$ and $h^o_{\text{tri}}(t)$.

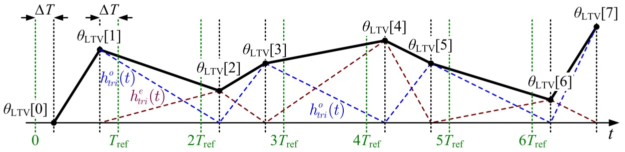

Fig. 12. Non-uniform linear interpolation between the $\theta_{\text{LTV}}[n]$ samples.

and it follows from Fig. 3(b) and Fig. 3(c) that

$$v[n] = -\alpha - r_F[n] + e_{qc}[n] - 2e_{qc}[n-1] + e_{qc}[n-2]. \tag{37}$$

The reference jitter can be written as $j_{\text{ref}}[n] = T_{\text{ref}}\theta_{\text{ref}}(t_n)$, where $\theta_{\text{ref}}(t)$ is the reference phase noise in cycles [31]. Consequently, (1) can be re-written as

$$t_n = nT_{\text{ref}} - T_{\text{ref}}\theta_{\text{ref}}(t_n) + d_e[n]. \tag{38}$$

The ADC samples and quantizes the CP output voltage at the rising edges of $v_{\text{samp}}(t)$. The time of the rising edge of $v_{\text{samp}}(t)$ during the nth reference period is greater than both τ_n and t_n and less than both τ_{n+1} and t_{n+1}, and each integer step of the ADC output corresponds to an ADC input step of Δ volts, so the ADC output can be written as

$$a[n] = \frac{1}{\Delta}v_{\text{CP}}[n] + e_{\text{ADC}}[n], \tag{39}$$

where $e_{\text{ADC}}[n]$ is the noise and distortion added by the ADC.

The $\Delta\Sigma$-FDC linearized model shown in Fig. 10 follows from (35) through (39) and Fig. 3(b) as follows. The shaded blocks in Fig. 10 labeled MMD, reference source, and ADC graphically implement (36), (38), and (39), respectively, and those labeled PFD and CP together graphically implement (35). The τ_0 term in (35) is not shown explicitly in the CP block of Fig. 10 because it can be interpreted as just contributing an initial condition of $\tau_0 I_{\text{CP}}/C$ to the CP block's accumulator, so it does not affect the $\Delta\Sigma$-FDC linearized model's transfer functions. The forward path blocks and the $z^{-1}(2-z^{-2})$ block within the FDC digital block in Fig. 10 are those shown in Fig. 3(b) where $\hat{g}_{\text{FDC}}$ is the value to which $\hat{g}[n]$ converges, and the remaining blocks within the FDC digital block in Fig. 10 graphically implement (37).

As $T_{\text{ref}} = T_{\text{PLL}}(N+\alpha)$, the portion of the output of the MMD accumulator in Fig. 10 corresponding to the $(N+\alpha)T_{\text{ref}}$ component of its input is nT_{ref}. This term cancels the nT_{ref} term introduced by the reference source at the PFD's differencer. Eliminating nT_{ref}, N, and α and rearranging the MMD and Q_{C} portions of Fig. 10 results in the linearized model shown in Fig. 4(a).

The $\Delta\Sigma$-FDC linearized model in Fig. 4(a) and Fig. 10 are valid provided that the ADC does not overload once the PLL locks. A nearly identical analysis to that presented in [21] shows that if the ADC's no-overload range is large enough to accommodate the various noise sources within the system, which is guaranteed by design, and the ADC does not overload at time indices n_0, n_0-1, n_0-2, n_0-3, and n_0-4, for some integer n_0, then the ADC will not overload for all $n \geq n_0$.

It can be verified from Fig. 4(a) that the system is stable (i.e., has all its poles inside the unit-circle in the z-domain) provided that $-1 < g < 1/3$, with g as defined in (9). This corresponds to an uncalibrated $\Delta\Sigma$-FDC loop gain error between -100% and 33%, which is a far larger range than would be expected in practice.

Appendix B
PLL Phase Noise LTV Model Derivation

Substituting (17) into (16) gives

$$\theta_{\text{PLL}}(t) = \theta_{\text{DCO}}(t) + K_{\text{DCO}}(t-t_n)f[n-1] + \theta_{\text{LTV}}[n] \tag{40}$$

for $t_n \leq t < t_{n+1}$, where

$$\theta_{\text{LTV}}[n] = K_{\text{DCO}}T_{\text{ref}}\sum_{k=-1}^{n-2}\left(1+(-1)^k\mu\right)f[k], \tag{41}$$

and $\mu = 2\Delta T/T_{\text{ref}}$. Equation (40) with $t = t_n$ reduces to $\theta_{\text{PLL}}[n] = \theta_{\text{DCO}}[n] + \theta_{\text{LTV}}[n]$. This with (7) implies that $p[n]$ in Fig. 3(a) can be written as $p[n] = -\theta_{\text{DCO}}[n] + e_{\text{FDC}}[n] - \theta_{\text{LTV}}[n]$. As $L(z)$ is the transfer function from $p[n]$ to $d[n]$, the component of $d[n]$ corresponding to noise is $f[n]$, and $p[n]$ is a noise sequence, it follows that $L(z)$ is the transfer function from $p[n]$ to $f[n]$. The portion of Fig. 11(a) between the $f[n]$ node and the node labeled $\theta_{\text{LTV}}[n]$ is a graphical representation of (41). It down-samples $f[n]$ into a stream of even-indexed samples scaled by $1+\mu$ and a stream of odd-index samples scaled by $1-\mu$, combines the streams via up-sampling and time-shift operations, and accumulates and scales the result. Together, these observations prove that the output of the feedback loop in Fig. 11(a) is indeed $\theta_{\text{LTV}}[n]$.

The second and third terms on the right side of (40) represent a linear interpolation operation between t_n and t_{n+1}, where, for each n, t_n is given by (1) with $d_e[n]$ given by (2). In typical PLLs, the reference source jitter is low enough that its effect on the interpolation operation is negligible [21]. However, for typical levels of duty cycle error, the effect of $d_e[n]$ on the interpolation process is not necessarily negligible.

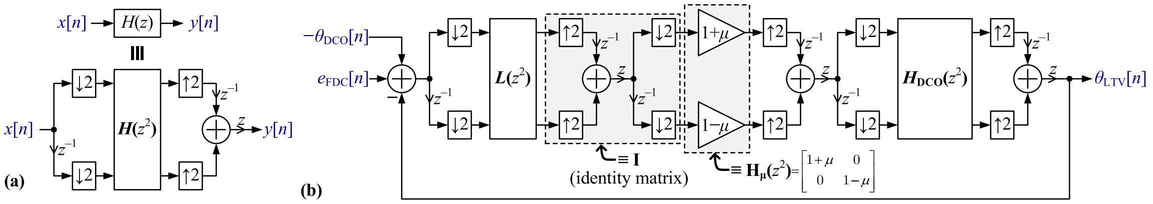

Fig. 13. (a) Block digital filtering technique, and (b) block digital filtering applied to the PLL's LTV phase noise model of Fig. 13(a).

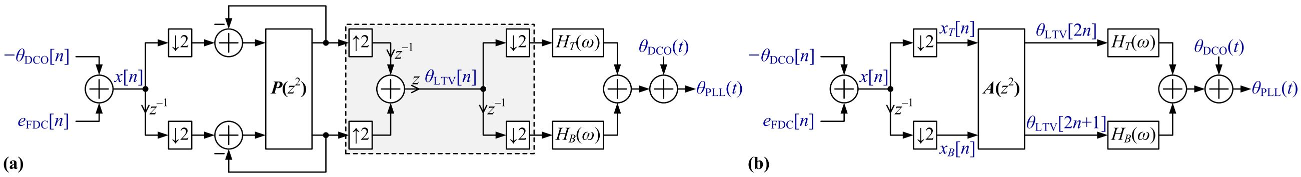

Fig. 14. (a) PLL LTV phase noise model after applying the block digital filtering technique, and (b) simplified representation.

As $d_e[n] = \Delta T(-1)^n$, it follows that

$$\theta_{\text{PLL}}(t) = \theta_{\text{DCO}}(t) + \sum_{n=0}^{\infty} \theta_{\text{LTV}}[2n] h_{\text{tri}}^{e}(t - nT_{\text{ref}} - \Delta T) + \sum_{n=0}^{\infty} \theta_{\text{LTV}}[2n+1] h_{\text{tri}}^{o}(t - (n+1)T_{\text{ref}} + \Delta T), \tag{42}$$

for all $t > 0$, where $h_{\text{tri}}^{e}(t)$ and $h_{\text{tri}}^{o}(t)$ are as shown in Fig. 11(b). The contributions of the two summations in (42) are illustrated in Fig. 12, from which it can be seen that the samples of $\theta_{\text{LTV}}[n]$ are first-order-hold interpolated between times $nT_{\text{ref}} + \Delta T$ and $(n+1)T_{\text{ref}} - \Delta T$ when n is even and between times $nT_{\text{ref}} - \Delta T$ and $(n+1)T_{\text{ref}} + \Delta T$ when n is odd. The portion of Fig. 11(a) between the $\theta_{\text{LTV}}[n]$ node and the output is a graphical implementation of (42), wherein the FOH$_{\text{e}}$ and FOH$_{\text{o}}$ interpolators respectively implement (20) with $s[n]$ replaced by the even-index samples of $\theta_{\text{LTV}}[n]$ and $h_{\text{tri}}(t)$ replaced by $h_{\text{tri}}^{e}(t)$ and with $s[n]$ replaced by the odd-index samples of $\theta_{\text{LTV}}[n]$ and $h_{\text{tri}}(t)$ replaced by $h_{\text{tri}}^{o}(t)$ ($h_{\text{tri}}^{e}(t)$ and $h_{\text{tri}}^{o}(t)$ are shown in Fig. 11(b)).

The analysis presented above proves that Fig. 11(a) represents a valid phase noise model of the PLL. The remainder of this appendix shows that the system of Fig. 5 is equivalent to that of Fig. 11(a).

The CTFT of the first summation in (42) can be evaluated as the product of the DTFT of $\theta_{\text{LTV}}[2n]$ and

$$H_T(\omega) = T_{\text{ref}} H(\omega) H_\mu(\omega) e^{-j\omega\Delta T}, \tag{43}$$

where $H(\omega)$ is given by (22) and

$$H_\mu(\omega) = \frac{1 + e^{j\omega T_{\text{ref}}\mu}\left((\mu-1)\cos(\omega T_{\text{ref}}) - \mu e^{-j\omega T_{\text{ref}}}\right)}{2(1-\mu^2)\sin^2(\omega T_{\text{ref}}/2)}. \tag{44}$$

Similarly, the CTFT of the second summation in (42) can be evaluated as the product of the DTFT of $\theta_{\text{LTV}}[2n+1]$ and

$$H_B(\omega) = T_{\text{ref}} H(\omega) H_\mu^*(\omega) e^{-j\omega(T_{\text{ref}} - \Delta T)}, \tag{45}$$

where $H_\mu^*(\omega)$ is the complex-conjugate of $H_\mu(\omega)$. Therefore, the FOH$_{\text{e}}$ interpolator followed by the $-\Delta T$ time shift and the FOH$_{\text{o}}$ interpolator followed by the $T_{\text{ref}} + \Delta T$ time shift in Fig. 11(a) represent multiplication in the frequency domain by $H_T(\omega)$ and $H_B(\omega)$, respectively.

The remainder of the proof utilizes a multi-rate system technique called block digital filtering [39]. Specifically, as illustrated in Fig. 13(a), any LTI transfer function, $H(z)$, with f_{ref}-rate input sequence, $x[n]$, can be parallelized and processed at a rate of $f_{\text{ref}}/2$ by a matrix transfer function,

$$\mathbf{H}(z^2) = \begin{bmatrix} H_0(z^2) & H_1(z^2) \\ z^{-2}H_1(z^2) & H_0(z^2) \end{bmatrix}, \tag{46}$$

called the blocked version of $H(z)$, where $H_0(z^2)$ and $H_1(z^2)$ are Type-I poly-phase components of $H(z)$ which satisfy

$$H(z) = H_0(z^2) + z^{-1}H_1(z^2). \tag{47}$$

For example, the DCO transfer function, $K_{\text{DCO}}T_{\text{ref}}/(1\text{–}z^{-1})$, can be represented as in Fig. 13(a) with $\mathbf{H}(z^2)$ replaced by

$$\mathbf{H_{DCO}}(z^2) = \frac{K_{\text{DCO}}T_{\text{ref}}}{1 - z^{-2}} \begin{bmatrix} 1 & 1 \\ z^{-2} & 1 \end{bmatrix}. \tag{48}$$

Applying the block digital-filtering technique to the portion to the left of the $\theta_{\text{LTV}}[n]$ node in Fig. 11(a) results in the block diagram in Fig. 13(b), where $\mathbf{L}(z^2)$ and $\mathbf{H_{DCO}}(z^2)$ are the blocked versions of $z^{-2}L(z)$ and $K_{\text{DCO}}T_{\text{ref}}/(1\text{–}z^{-1})$, respectively. As indicated in Fig. 13(b), the 1+μ and 1–μ multipliers can be implemented as $\mathbf{H}_{\boldsymbol{\mu}}(z^2)$, where $\mathbf{H}_{\boldsymbol{\mu}}(z^2)$ is specified in the figure, and the shaded cascade of up-sampling and down-sampling operations can be implemented as the identity matrix, $\mathbf{I}$. Two other cascades of up-sampling and down-sampling operations, each of which can also be implemented as the identity matrix, occur in Fig. 13(b): one between $\mathbf{H}_{\boldsymbol{\mu}}(z^2)$ and $\mathbf{H_{DCO}}(z^2)$ and the other with the up-sampling operations to the right of $\mathbf{H_{DCO}}(z^2)$ and the down-sampling operations to the left of $\mathbf{L}(z^2)$.

Applying these observations leads to the system shown in Fig. 14(a), where $\mathbf{P}(z) = [p_{ij}(z)]$ is the matrix product of $\mathbf{H_{DCO}}(z)$, $\mathbf{H}_{\boldsymbol{\mu}}(z)$, and $\mathbf{L}(z)$. Therefore,

$$\mathbf{P}(z) = p(z) \begin{bmatrix} 1+\mu & 1-\mu \\ z^{-1}(1+\mu) & 1-\mu \end{bmatrix} \begin{bmatrix} a(z) & b(z) \\ z^{-1}b(z) & a(z) \end{bmatrix}, \tag{49}$$

where

$$a(z) = \left[1 + z^{-1}g(z)\right] + \lambda z^{-1}\left[1 + g(z)\right], \tag{50}$$

$$b(z) = \lambda \left[1 + z^{-1} g(z)\right] + \left[1 + g(z)\right], \quad (51)$$

$$p(z) = K_I \frac{z^{-1}}{g(z)} \cdot \frac{K_{\mathrm{DCO}} T_{\mathrm{ref}}}{(1 - z^{-1})^2} \cdot \left(\frac{1-\lambda}{1-\lambda^2 z^{-1}}\right), \quad (52)$$

and

$$g(z) = \frac{K_I}{K_P} \cdot \frac{1}{1 + \left(\frac{K_I}{K_P} - 1\right) z^{-1}}. \quad (53)$$

It follows from the expressions for $a(z)$, $b(z)$, and $p(z)$ above and tedious algebra that $T(z)$ in (26) can be written as

$$T(z) = p(z^2)(1 + z^{-1}) \left[a(z^2) + z^{-1} b(z^2)\right]. \quad (54)$$

As the up-sampling and down-sampling operations in the shaded box in Fig. 14(a) can be implemented as an identity matrix, Fig. 14(a) can be redrawn as shown in Fig. 14(b), where $\mathbf{A}(z) = [a_{ij}(z)]$ is given by

$$\mathbf{A}(z) = \frac{1}{D(z)} \begin{bmatrix} 1 + p_{22}(z) & -p_{12}(z) \\ -p_{21}(z) & 1 + p_{11}(z) \end{bmatrix} \mathbf{P}(z), \quad (55)$$

and

$$D(z) = \left[1 + p_{11}(z)\right]\left[1 + p_{22}(z)\right] - p_{12}(z) p_{21}(z). \quad (56)$$

Substituting the elements of $\mathbf{P}(z)$ implied by (49) through (53) into (56) and applying (54) gives

$$D(z^2) = [1 + T(z)]\,[1 + T(-z)] - \mu^2 T(z) T(-z). \quad (57)$$

The results presented above show that the system of Fig. 11(a) is equivalent to that of Fig. 14(b).

The DTFTs of $x_T[n]$ and $x_B[n]$ in Fig. 14(b) are

$$X_T\left(e^{2j\omega T_{\mathrm{ref}}}\right) = \frac{1}{2}\left(X\left(e^{j\omega T_{\mathrm{ref}}}\right) + X\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right)\right), \quad (58)$$

and

$$X_B\left(e^{2j\omega T_{\mathrm{ref}}}\right) = \frac{1}{2} e^{-j\omega T_{\mathrm{ref}}}\left(X\left(e^{j\omega T_{\mathrm{ref}}}\right) - X\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right)\right), \quad (59)$$

respectively [39]. It follows from (58) and (59) and the operations shown in Fig. 14(b) with expressions for the elements of $\mathbf{A}(z)$ given by (49) through (57) with $z = e^{j\omega T \mathrm{ref}}$ that the CTFT of the sum of the outputs of $H_T(\omega)$ and $H_B(\omega)$ can be written as (23) with

$$C(\omega) = A_T\left(e^{2j\omega T_{\mathrm{ref}}}\right) H_T(\omega) + A_B\left(e^{2j\omega T_{\mathrm{ref}}}\right) H_B(\omega), \quad (60)$$

and

$$C^{\mathrm{a}}(\omega) = A_T\left(e^{j(2\omega T_{\mathrm{ref}} - \pi)}\right) H_T(\omega) + A_B\left(e^{j(2\omega T_{\mathrm{ref}} - \pi)}\right) H_B(\omega), \quad (61)$$

where

$$A_T\left(e^{2j\omega T_{\mathrm{ref}}}\right) = \frac{1}{2} G\left(e^{j\omega T_{\mathrm{ref}}}\right)\left[E\left(e^{j\omega T_{\mathrm{ref}}}\right) + F\left(e^{j\omega T_{\mathrm{ref}}}\right)\right], \quad (62)$$

$$A_B\left(e^{2j\omega T_{\mathrm{ref}}}\right) = \frac{e^{-j\omega T_{\mathrm{ref}}}}{2} G\left(e^{j\omega T_{\mathrm{ref}}}\right)\left[E\left(e^{j\omega T_{\mathrm{ref}}}\right) - F\left(e^{j\omega T_{\mathrm{ref}}}\right)\right], \quad (63)$$

$G\left(e^{j\omega T_{\mathrm{ref}}}\right)$ is given by (26) with $z = e^{j\omega T_{\mathrm{ref}}}$,

$$E\left(e^{j\omega T_{\mathrm{ref}}}\right) = \frac{1 + T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right) - \mu^2 T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right)}{1 + T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right) - \mu^2 \frac{T\left(e^{j\omega T_{\mathrm{ref}}}\right) T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right)}{1 + T\left(e^{j\omega T_{\mathrm{ref}}}\right)}}, \quad (64)$$

and

$$F\left(e^{j\omega T_{\mathrm{ref}}}\right) = \frac{(1 - e^{-j\omega T_{\mathrm{ref}}})\mu/(1 + e^{-j\omega T_{\mathrm{ref}}})}{1 + T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right) - \mu^2 \frac{T\left(e^{j\omega T_{\mathrm{ref}}}\right) T\left(e^{j(\omega T_{\mathrm{ref}} - \pi)}\right)}{1 + T\left(e^{j\omega T_{\mathrm{ref}}}\right)}}. \quad (65)$$

Therefore, to prove that system of Fig. 5 is equivalent to those of Fig. 14(b) and Fig. 11(a), it is sufficient to derive expressions for $H_{\mathrm{err}}(j\omega)$ and $H_{\mathrm{err}}^{a}(j\omega)$ in Fig. 5 with which $C(\omega)$ and $C^{a}(\omega)$ given by (60) and (61) are equivalently given by (24) and (25). Substituting (44) into (43) and (45) and the results into (60) and (61) shows that (60) and (61) can be written as (24) and (25) with

$$H_{\mathrm{err}}(j\omega) = E(e^{j\omega T_{\mathrm{ref}}}) \cdot \mathrm{Re}\left\{H_\mu(\omega) e^{-j\omega T_{\mathrm{ref}}\mu/2}\right\} + jF(e^{j\omega T_{\mathrm{ref}}}) \cdot \mathrm{Im}\left\{H_\mu(\omega) e^{-j\omega T_{\mathrm{ref}}\mu/2}\right\}, \quad (66)$$

and

$$H_{\mathrm{err}}^{a}(j\omega) = F(e^{j(\omega T_{\mathrm{ref}} - \pi)}) \cdot \mathrm{Re}\left\{H_\mu(\omega) e^{-j\omega T_{\mathrm{ref}}\mu}\right\} + jE(e^{j(\omega T_{\mathrm{ref}} - \pi)}) \cdot \mathrm{Im}\left\{H_\mu(\omega) e^{-j\omega T_{\mathrm{ref}}\mu}\right\}. \quad (67)$$

References

[1] W. Wu, "Low-jitter frequency generation techniques for 5G communication: A tutorial," *IEEE Solid State Circuits Mag.*, vol. 13, no. 4, pp. 44–63, Aug. 2021.

[2] M. Oveisi and P. Heydari, "A study of BER and EVM degradation in digital modulation schemes due to PLL jitter and communication-link noise," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 8, pp. 3402–3415, Aug. 2022.

[3] S. Levantino, "Recent advances in high-performance frequency synthesizer design," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2022, pp. 1–7.

[4] M. Shinagawa, Y. Akazawa, and T. Wakimoto, "Jitter analysis of high-speed sampling systems," *IEEE J. Solid-State Circuits*, vol. 25, no. 1, pp. 220–224, Feb. 1990.

[5] B. Razavi, "Jitter-power trade-offs in PLLs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 4, pp. 1381–1387, Apr. 2021.

[6] B. Razavi, "Lower bounds on power consumption of clock generators for ADCs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[7] P. M. Aziz, H. V. Sorensen, and J. van der Spiegel, "An overview of sigma-delta converters," in *IEEE Signal Process. Maga-zine*, vol. 13, no. 1, pp. 61–84, Jan. 1996.

[8] *Ultra-Performance Oscillator*, document SiT8209, 2017.

[9] *VCXO Ultra-Low Phase Noise Oscillators*, document CVHD-950, 2019.

[10] *Ultra-Low Phase Noise VCXO With Standby Mode*, document CVHD-957, 2019.

[11] W. Wu et al., "A 28-nm 75-fsrms analog fractional-*N* sampling PLL with a highly linear DTC incorporating background DTC gain calibration and reference clock duty cycle correction," *IEEE J. Solid-State Circuits*, vol. 54, no. 5, pp. 1254–1265, May 2019.

[12] W. Wu et al., “A 14-nm ultra-low jitter fractional-N PLL using a DTC range reduction technique and a reconfigurable dual-core VCO,” *IEEE J. Solid-State Circuits*, vol. 56, no. 12, pp. 3756–3767, Dec. 2021.

[13] S. Abdollahi-Alibeik et al., “A 65nm dual-band 3-stream 802.11n MIMO WLAN SoC,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2011, pp. 170–172.

[14] Y.-L. Hsueh et al., “28.2 A 0.29 mm^2 frequency synthesizer in 40nm CMOS with 0.19ps$_{rms}$ jitter and <-100dBc reference spur for 802.11ac,” in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, San Francisco, CA, USA, Feb. 2014, pp. 472–473.

[15] T. Siriburanon et al., “A 28-GHz fractional-N frequency synthesizer with reference and frequency doublers for 5G cellular,” in *Proc. 41st Eur. Solid-State Circuits Conf. (ESSCIRC)*, Graz, Graz, Austria, Sep. 2015, pp. 76–79.

[16] H. Kim, Y. Kim, T. Kim, H.-J. Ko, and S. Cho, “A 2.4-GHz 1.5-mW digital multiplying delay-locked loop using pulsewidth comparator and double injection technique,” *IEEE J. Solid-State Circuits*, vol. 52, no. 11, pp. 2934–2946, Nov. 2017.

[17] H. Yoon et al., “A -31dBc integrated-phase-noise 29GHz fractional-N frequency synthesizer supporting multiple frequency bands for backward-compatible 5G using a frequency doubler and injection-locked frequency multipliers,” in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2018, pp. 366–368.

[18] A. Elkholy, D. Coombs, R. K. Nandwana, A. Elmallah, and P. K. Hanumolu, “A 2.5–5.75-GHz ring-based injection-locked clock multiplier with background-calibrated reference frequency doubler,” *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 2049–2058, Jul. 2019.

[19] C.-W. Yao, L. Lin, B. Nissim, H. Arora, and T. Cho, “A low spur fractional-N digital PLL for 802.11 a/b/g/n/AC with 0.19 ps$_{rms}$ jitter,” in *Proc. Symp. VLSI Circuits-Dig. Tech. Papers*, Jun. 2011, pp. 110–111.

[20] X. Gao et al., “9.4 A 28nm CMOS digital fractional-N PLL with -245.5dB FOM and a frequency tripler for 802.11abgn/AC radio,” in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2015, pp. 1–3.

[21] C. Venerus and I. Galton, “Delta-sigma FDC based fractional-N PLLs,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 5, pp. 1274–1285, May 2013.

[22] C. Venerus and I. Galton, “Quantization noise cancellation for FDC-based fractional-N PLLs,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 12, pp. 1119–1123, Dec. 2015.

[23] C. -M. Hsu, M. Z. Straayer, and M. H. Perrott, “A low-noise wide-BW 3.6-GHz digital $\Delta\Sigma$ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation,” *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec. 2008.

[24] M. Zanuso, S. Levantino, C. Samori, and A. L. Lacaita, “A wideband 3.6 GHz digital $\Delta\Sigma$ fractional-N PLL with phase interpolation divider and digital spur cancellation,” *IEEE J. Solid-State Circuits*, vol. 46, no. 3, pp. 627–638, Mar. 2011.

[25] C.-W. Yao et al., “A 14-nm 0.14-psrms Fractional-N digital PLL with a 0.2-ps resolution ADC-assisted coarse/fine-conversion chopping TDC and TDC nonlinearity calibration,” *IEEE J. Solid-State Circuits*, vol. 52, no. 12, pp. 3446–3457, Dec. 2017.

[26] L. Bertulessi, D. Cherniak, M. Mercandelli, C. Samori, A. L. Lacaita, and S. Levantino, “Novel feed-forward technique for digital bang-bang PLL to achieve fast lock and low phase noise,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 5, pp. 1858–1870, May 2022.

[27] C. Weltin-Wu, E. Familier, and I. Galton, “A linearized model for the design of fractional-N digital PLLs based on dual-mode ring oscillator FDCs,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 8, pp. 2013–2023, Aug. 2015.

[28] M. Talegaonkar et al., “A 5GHz digital fractional-N PLL using a 1-bit delta-sigma frequency-to-digital converter in 65 nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 52, no. 9, pp. 2306–2320, Sep. 2017.

[29] E. Helal, E. Alvarez-Fontecilla, A. I. Eissa, and I. Galton, “A time amplifier assisted frequency-to-digital converter based digital fractional-N PLL,” *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2711–2723, Sep. 2021.

[30] E. Helal, A. I. Eissa, and I. Galton, “DTC linearization via mismatch-noise cancellation for digital fractional-N PLLs,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 12, pp. 4993–5006, Dec. 2022.

[31] I. Galton and C. Weltin-Wu, “Understanding phase error and jitter: Definitions, implications, simulations, and measurement,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 1–19, Jan. 2019.

[32] B. Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[33] N. D. Dalt, “Markov chains-based derivation of the phase detector gain in bang-bang PLLs,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 11, pp. 1195–1199, Nov. 2006.

[34] J. Proakis and D. Manolakis, *Digital Signal Processing*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[35] E. Alvarez-Fontecilla, A. I. Eissa, E. Helal, C. Weltin-Wu, and I. Galton, “Delta-sigma FDC enhancements for FDC-based digital fractional-N PLLs,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 965–974, Mar. 2021.

[36] 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.

[37] A. Yu, D. Bankman, K. Zheng, and B. Murmann, “Understanding metastability in SAR ADCs: Part II: Asynchronous,” *IEEE Solid State-Circuits Mag.*, vol. 11, no. 3, pp. 16–32, May 2019.

[38] R. B. Staszewski and P. T. Balsara, “Phase-domain all-digital phase-locked loop,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 3, pp. 159–163, Mar. 2005.

[39] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

Amr I. Eissa received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from Ain Shams University (ASU), Cairo, Egypt, in 2012 and 2016, respectively. He is currently pursuing the Ph.D. degree with the University of California at San Diego, San Diego, CA, USA. From 2013 to 2016, he was a Teaching and Research Assistant with the Electronics and Communications Engineering Department, ASU. His research interests include the analysis and design of analog/mixed-signal integrated circuits and systems.

Enrique Alvarez-Fontecilla received the B.Sc. and M.Sc. degrees in electrical engineering from Pontificia Universidad Católica de Chile (PUC), Santiago, Chile, in 2011 and 2013, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego, San Diego, CA, USA, in 2021. From 2012 to 2015, he was an Adjunct Assistant Professor with the School of Engineering and the Institute of Philosophy, PUC. He currently works on calibration algorithms for high-speed data converters with Analog Devices Inc., Santiago.

Colin Weltin-Wu (Member, IEEE) received the B.S. degree in mathematics and the M.E. degree in electrical engineering from Massachusetts Institute of Technology and the Ph.D. degree from Columbia University. In 2014, he spent 15 months as a Post-Doctoral Researcher with Ian Galton at UC San Diego researching highly-digital frequency synthesizers. In 2018, he co-founded Aeonsemi, Inc., a fabless semiconductor company specializing in mixed-signal ICs for communication and networking. His research interests include data converters, frequency synthesizers, and computer-aided design.

Ian Galton (Fellow, IEEE) received the B.Sc. degree in electrical engineering from Brown University in 1984, and the M.S. and Ph.D. degrees in electrical engineering from California Institute of Technology in 1989 and 1992, respectively. Since 1996, he has been a Professor of electrical engineering with the University of California at San Diego, where he teaches and conducts research in the field of mixed-signal integrated circuits and systems for communications. His research involves the invention, analysis, and integrated circuit implementation of critical communication system blocks, such as data converters and phase-locked loops.