A Discrete-Time Model for the Design of Type-II PLLs With Passive Sampled Loop Filters

Kevin J. Wang, *Member, IEEE*, and Ian Galton, *Senior Member, IEEE*

***Abstract*—Type-II charge-pump (CP) phase-locked loop (PLLs) are used extensively in electronic systems for frequency synthesis. Recently, a passive sampled loop filter (SLF) has been shown to offer major benefits over the conventional continuous-time loop filter traditionally used in such PLLs. These benefits include greatly enhanced reference spur suppression, elimination of CP pulse-position modulation nonlinearity, and, in the case of phase noise cancelling fractional-N PLLs, improved phase noise cancellation. The main disadvantage of the SLF to date has been the lack of a linear time-invariant (LTI) model with which to perform the system-level design of SLF-based PLLs. Without such a model, designers are forced to rely on trial and error iteration supported by lengthy transient simulations. This paper presents an accurate LTI model of SLF-based type-II PLLs that eliminates this disadvantage.**

***Index Terms*—Frequency synthesis, phase-locked loop (PLL), PLL linearized model, sampled loop filter (SLF).**

I. Introduction

INTEGER-N and fractional-N phase-locked loops (PLLs) are used extensively in electronic systems to synthesize higher frequency signals from lower frequency references. The majority of these PLLs are charge-pump (CP)-based type-II PLLs [1].

Recently, sampled loop filters (SLFs) have been shown to offer advantages over continuous-time loop filters (CLFs) in PLLs. SLFs can greatly reduce reference spurs in both integer-N and fractional-N PLLs [2], [3]. They eliminate CP pulse-position modulation distortion in fractional-N PLLs [4], [5], and they improve phase noise cancellation in phase noise cancelling fractional-N PLLs [5], [6]. Moreover, SLFs eliminate the large reference spur that would otherwise arise as a side effect of the CP offset current method for reducing fractional spurs in fractional-N PLLs [3], [7].

Several different types of SLFs for PLLs have been published. In [4], an active SLF is implemented by preceding a CLF with an op-amp-based sample-and-hold circuit. In [2], a passive switched-capacitor SLF is implemented for a type-I PLL. In [3], a passive SLF is implemented with the addition of a transistor switch within an otherwise conventional CLF.

The SLF presented in [3] offers a major benefit over the other SLFs: It is the only published passive SLF applicable to type-II PLLs. The sampling operation involves only a single switch, so it consumes very little power and circuit area beyond those of a comparable CLF. Its applicability to type-II PLLs is important because such PLLs are by far the most widely used PLLs at present. Furthermore, the SLF has been demonstrated in a fractional-N PLL with record-setting reference and fractional spur performance.

The main drawback to date of the SLF presented in [3] has been the lack of a linear time-invariant (LTI) model with which to perform the system-level design of PLLs based on the SLF. Without such a model, designers are forced to rely on trial and error iteration and lengthy transient simulations as their primary design tools.

Despite its implementation simplicity, the SLF presented in [3] is more difficult to analyze than the other published SLFs because it behaves as a time-varying continuous-time filter. Therefore, it cannot be well approximated as a continuous-time LTI system. Nevertheless, as proven in this paper, PLLs based on the SLF can be modeled accurately as discrete-time LTI systems. This paper derives such an LTI model and demonstrates how it enables the system-level design of PLLs without the need to resort to computer simulation. Hence, the results of this paper eliminate the drawback described earlier.

The model yields equations which accurately predict the transfer functions, bandwidth, and phase margin (PM) of the PLL in terms of its component values. While the equations are not simple, they each have closed form. They can be implemented easily in a tool such as Matlab and used to rapidly generate results that heretofore required lengthy transient simulations. The PLL design process is inherently iterative, so not having to simulate the PLL at each iteration step significantly speeds up the design process.

This paper is organized such that all the information required to use the model to design PLLs is presented separately from the derivation of the model. This allows readers to use the model prior to understanding its derivation. The information required to use the model is presented in Sections II–III and Appendix A, and the detailed mathematical derivation of the model is presented in Section IV and Appendix B.

II. Overview of the SLF PLL

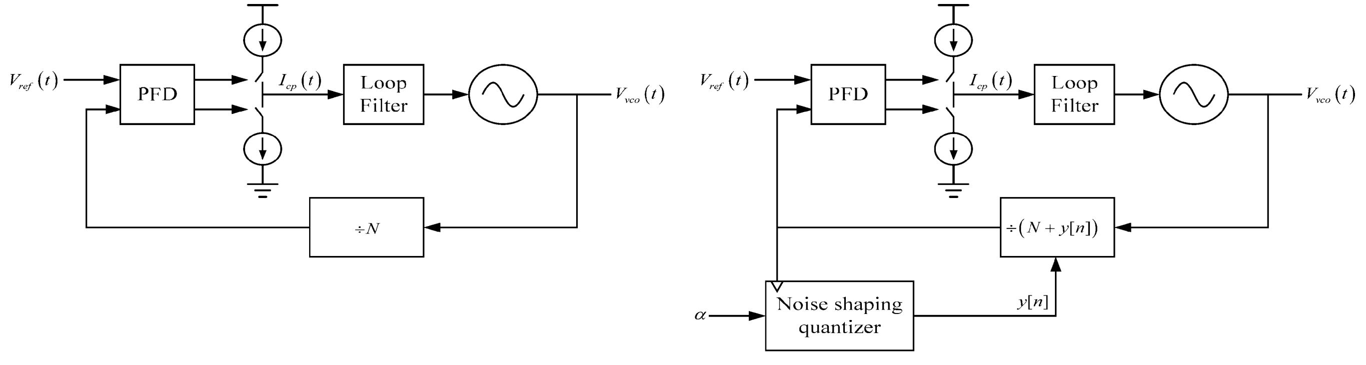
The block diagram of a typical CP-based integer-N PLL is shown in Fig. 1(a) [1]. Its purpose is to generate a spectrally pure periodic output signal with a frequency of Nf_{ref}, where N is a positive integer and f_{ref} is the frequency of the reference signal $V_{\text{ref}}(t)$. It consists of a phase-frequency detector (PFD),

Manuscript received April 18, 2010; revised June 10, 2010; accepted June 30, 2010. Date of publication November 11, 2010; date of current version January 28, 2011. This work was supported by the National Science Foundation under Award 0914748. This paper was recommended by Associate Editor H. Luong.

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

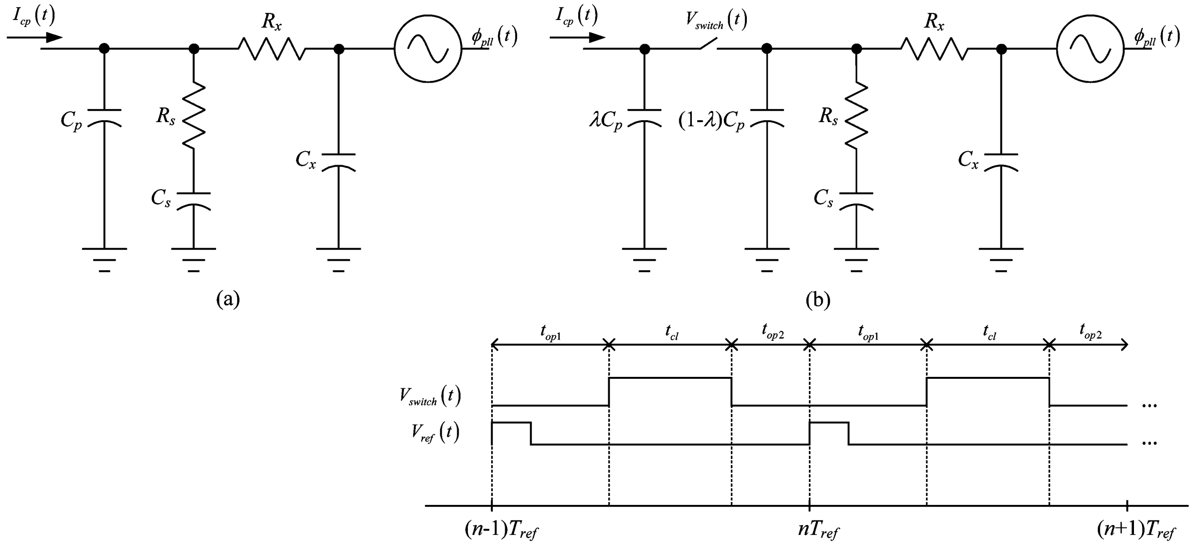
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2010.2072130



(a) (b)

Fig. 1. Block diagram of a typical (a) integer-N PLL and (b) fractional-N PLL.



(a) (b)

(c)

Fig. 2. Circuit diagram of (a) a CLF with the VCO, (b) an SLF with VCO, and (c) the timing of $V_{\mathrm{switch}}(t)$.

a CP, a low-pass loop filter (LF), a voltage-controlled oscillator (VCO), and a digital divider.

The divider output is a two-level signal in which the nth and $(n+1)$th rising edges, for $n = 0, 1, 2, \ldots$, are separated by N periods of the VCO output. The PFD compares the positive going edges of the reference signal to those of the divider's output signal and causes the CP to drive the LF with current pulses whose widths are proportional to the phase difference between the two signals. The pulses are low-pass filtered by the LF, and the resulting waveform drives the VCO.

Fig. 2(a) shows a CLF, and Fig. 2(b) shows the SLF addressed in this paper. The SLF differs from the CLF only in that it includes a switch which splits C_p into λC_p and $(1-\lambda)C_p$, where $0 < \lambda < 1$. For example, in [3], $\lambda = 0.5$. The switch is opened and closed once per reference period such that when the PLL is locked, λC_p and $(1-\lambda)C_p$ are disconnected whenever $I_{\mathrm{cp}}(t) \neq 0$. As explained and experimentally demonstrated in [3], this significantly reduces the reference spur compared to the conventional LF.

The switch is controlled by the two-level signal $V_{\mathrm{switch}}(t)$; it is closed when $V_{\mathrm{switch}}(t)$ is high and open when $V_{\mathrm{switch}}(t)$ is low. A typical waveform for $V_{\mathrm{switch}}(t)$ is shown in Fig. 2(c). The nth reference period is defined as the time interval between the nth and $(n+1)$th rising edges of the reference signal. In the case of a noise-free reference signal, these edges occur at times nT_{ref} and $(n+1)T_{\mathrm{ref}}$, respectively, where $T_{\mathrm{ref}} = 1/f_{\mathrm{ref}}$. As shown in Fig. 2(c), during each reference period, the switch is first open for a duration of t_{op1}, then closed for a duration of t_{cl}, and then open for a duration of t_{op2}, where t_{op1}, t_{cl}, and t_{op2} are constants chosen by the designer. Together with the LF components, these constants define the behavior of the SLF.

As described in Section III and suggested by the model equations in Appendix A, decreasing t_{cl} has the effect of decreasing the PM of the PLL, whereas the values of t_{op1} and t_{op2} for any given value of t_{cl} have little effect on the dynamics of the PLL. Therefore, t_{op1}, t_{cl}, and t_{op2} should be chosen such that t_{cl} is as large as possible subject to the requirement that the switch be open whenever $I_{\mathrm{cp}}(t) \neq 0$ once the PLL is locked.

The block diagram of a typical CP-based fractional-N PLL is shown in Fig. 1(b) [1]. Its purpose is to generate a spectrally pure periodic output signal with a frequency of $(N+\alpha)f_{\mathrm{ref}}$, where N is again a positive integer and α is a fractional value between zero and one. The fractional-N PLL differs from the integer-N PLL only in that the nth and $(n+1)$th rising edges of the divider output, for $n = 0, 1, 2, \ldots$, are separated by $N + y[n]$ periods of the VCO output, where $y[n]$ is the integer-valued

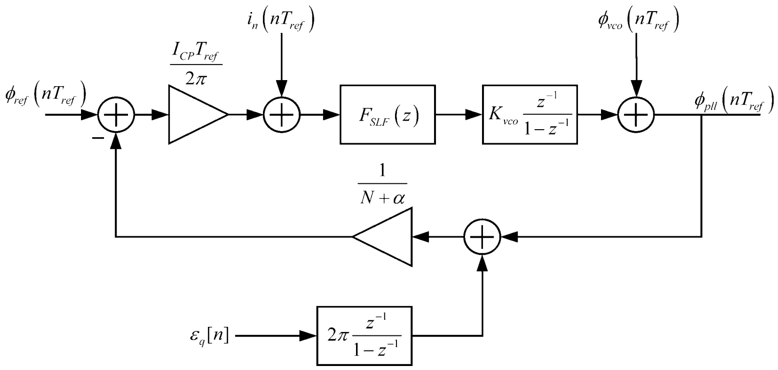


Fig. 3. Single-rate discrete-time linearized model of an SLF-PLL with noise sources.

output sequence from a noise-shaping quantizer with input α. Typically, the noise-shaping quantizer is a digital delta–sigma modulator, but other types of quantizers such as a successive requantizer can also be used [3].

III. Description and Application of the PLL Model

This section describes the proposed model of the PLLs shown in Fig. 1 with the SLF of Fig. 2(b) and explains how the model can be used to analyze and design such PLLs. The mathematical derivations that underlie the models are referred to Section IV and Appendix B.

A. Model Description

The phase of the fractional-N PLL's output signal at time t can be written as

$$2\pi(N+\alpha)f_{\text{ref}}t+\phi_{\text{pll}}(t) \tag{1}$$

where $\phi_{\text{pll}}(t)$ represents the PLL's phase error, i.e., the difference between the actual phase and ideal phase of the PLL output signal at time t.

The purpose of a PLL model is to provide a simple means of evaluating $\phi_{\text{pll}}(t)$ in terms of the PLL's design parameters and error signals such as circuit noise, assuming that the PLL is already locked. PLLs are neither linear nor time invariant, but when locked, they can be approximated as LTI systems. For example, the most commonly used model for PLLs with conventional LFs is a continuous-time LTI system that accurately models the locked behavior of such PLLs [8], [9], [12]. Discrete-time LTI models have also been developed for such PLLs [8], [10], [11].

The model presented in this section is a discrete-time LTI system applicable to the SLF-PLL. As described in the next section, the sampling operation of the SLF would result in a time-varying continuous-time model which would be difficult to analyze, and this problem is avoided by using a discrete-time model.

Two versions of the model are presented: a single-rate version and a multirate version. The single-rate version provides samples of $\phi_{\text{pll}}(t)$ at a sample rate of f_{ref}. The multirate version provides samples of $\phi_{\text{pll}}(t)$ at a sample rate of Lf_{ref}, where $L\geq 2$ is a positive integer.

The two versions of the model are identical in terms of how they represent the PLL's feedback behavior, but the latter performs interpolation to obtain extra $L-1$ output samples per reference period. When $\phi_{\text{pll}}(t)$ has most of its power concentrated at frequencies with magnitudes less than $f_{\text{ref}}/2$, the single-rate version of the model is sufficient. The multirate version, although more complicated than the single-rate version, is useful in cases where $\phi_{\text{pll}}(t)$ has enough power at frequencies with magnitudes above $f_{\text{ref}}/2$ in that it is necessary to sample $\phi_{\text{pll}}(t)$ at a higher sample rate than f_{ref}.

The single-rate version of the model is shown in Fig. 3, where I_{CP} is the magnitude of current pulses sourced and sunk by the CP, $i_n(nT_{\text{ref}})$ is CP noise sampled at nT_{ref}, $\phi_{\text{ref}}(nT_{\text{ref}})$ is the reference signal's phase noise sampled at nT_{ref}, $\phi_{\text{vco}}(nT_{\text{ref}})$ is the open-loop VCO phase noise sampled at nT_{ref}, $\varepsilon_q[n]=y[n]-\alpha$ is the quantization noise from the noise-shaping quantizer

$$F_{\text{SLF}}(z)=\frac{(1-\gamma_1z^{-1})(1-\gamma_2z^{-1})(1-\gamma_3z^{-1})}{(1-z^{-1})(1-\beta_2z^{-1})(1-\beta_3z^{-1})} \tag{2}$$

and β_i, γ_j, and K are constants. Appendix A provides equations that yield the values of β_i, γ_j, and K given the LF design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. The model, as shown in Fig. 3, applies to the fractional-N PLL, but when modified to have $\alpha=0$ and $\varepsilon_q[n]=0$, it also applies to the integer-N PLL.

The PLL's locked behavior can be analyzed by applying well-known LTI system techniques to the model in Fig. 3. Specifically, the model indicates that the loop gain is

$$T(z)=-\frac{I_{\text{CP}}T_{\text{ref}}K_{\text{vco}}}{2\pi(N+\alpha)}F_{\text{SLF}}(z)\frac{z^{-1}}{1-z^{-1}}. \tag{3}$$

Therefore, the PLL's PM is

$$PM=\sphericalangle T\left(e^{j\omega_uT_{\text{ref}}}\right) \tag{4}$$

where ω_u is the unity-gain frequency of $T(e^{j\omega}T_{\text{ref}})$ and the loop bandwidth (LBW) of the PLL is approximately equal to ω_u. The noise transfer functions from $\phi_{\text{ref}}(nT_{\text{ref}})$, $\phi_{\text{vco}}(nT_{\text{ref}})$, $\varepsilon_q[n]$, and $i_n(nT_{\text{ref}})$ to $\phi_{\text{pll}}(nT_{\text{ref}})$, respectively, are

$$\frac{\phi_{\text{pll}}}{\phi_{\text{ref}}}(z)=(N+\alpha)\frac{T(z)}{1+T(z)} \tag{5}$$

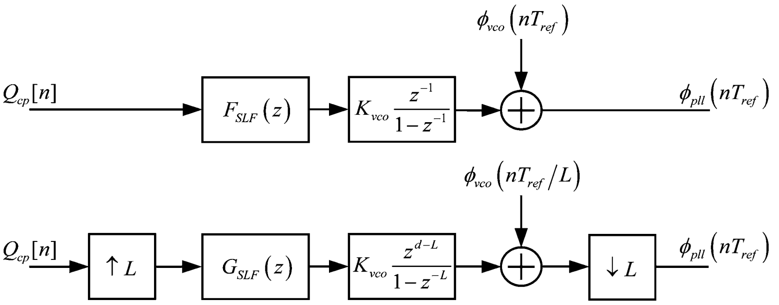


Fig. 4. Model of the SLF and VCO for the (a) single rate and (b) multirate cases.

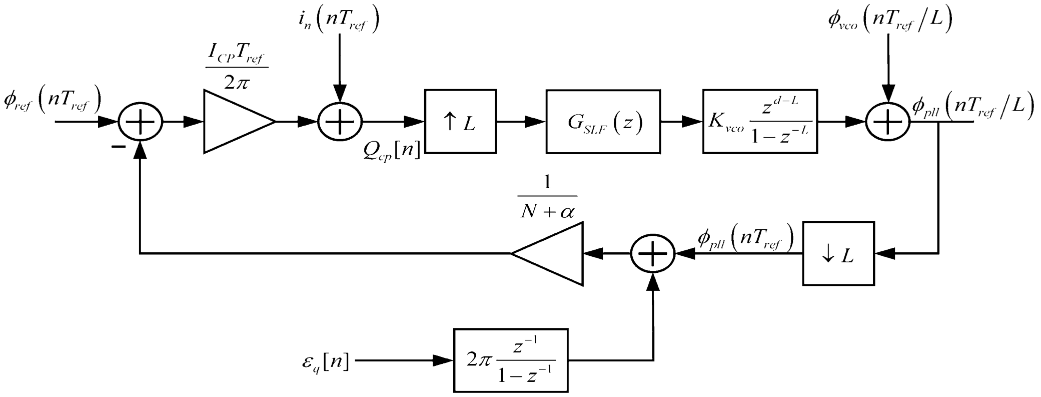


Fig. 5. Multirate discrete-time linearized model of an SLF-PLL with noise sources.

$$\frac{\phi\text{pll}}{\phi_{\text{vco}}}(z) = \frac{1}{1+T(z)} \tag{6}$$

$$\frac{\phi\text{pll}}{\varepsilon_q}(z) = -2\pi\frac{z^{-1}}{1-z^{-1}}\frac{T(z)}{1+T(z)} \tag{7}$$

$$\frac{\phi\text{pll}}{i_{\text{cp}}}(z) = \frac{2\pi(N+\alpha)}{I_{\text{CP}}T_{\text{ref}}}\frac{T(z)}{1+T(z)}. \tag{8}$$

The multirate version of the PLL model differs from the single-rate version shown in Fig. 3 only in its representation of the SLF and VCO. The components of the single-rate model that represent the SLF and VCO are shown separately in Fig. 4(a). The multirate version is obtained by removing these components in the single-rate model of Fig. 3 and replacing them with the components shown in Fig. 4(b). The resulting multirate model is shown in Fig. 5.

Therefore, the SLF and VCO in the multirate model are represented by the components shown in Fig. 4(b): an L-fold upsampler, a discrete-time filter with sample rate Lf_{ref} and transfer function

$$K_{\text{vco}}G_{\text{SLF}}(z)\frac{z^{d-L}}{1-z^{L}} \tag{9}$$

the addition of the VCO phase noise sampled at a rate of Lf_{ref}, and an L-fold downsampler. The integer d is defined as

$$d = \left\lfloor \frac{t_{\text{cl}}+t_{\text{op2}}}{T_{\text{ref}}}L \right\rfloor \tag{10}$$

where $\lfloor y \rfloor$ is the largest integer less than or equal to y. The output of the L-fold upsampler is given by

$$R_{\text{cp}}[n] = \begin{cases} Q_{\text{cp}}[n/L], & \text{if } n \text{ is an integer multiple of } L \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

and the L-fold downsampler discards all but every Lth sample of $\phi_{\text{pll}}(nT_{\text{ref}}/L)$ to obtain $\phi_{\text{pll}}(nT_{\text{ref}})$. The transfer function $G_{\text{SLF}}(z)$ has the form

$$G_{\text{SLF}}(z) = \frac{1}{L}\sum_{i=0}^{L-1} z^{-i}F_{\text{SLF},i}(z^L) \tag{12}$$

where each $F_{\text{SLF},i}(z)$ has the same three poles as (2) and can have either two or three zeros. Appendix A provides equations that yield the full transfer function of each $F_{\text{SLF},i}(z)$ given L and the LF design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}.

B. Analysis Example

The parameters that specify the system-level design of an SLF-PLL are $N, f_{\text{ref}}, I_{\text{CP}}, K_{\text{vco}}$, and the LF design values, i.e., the values of C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. Both versions of the model described in Section III-A describe the locked behavior of the PLL in terms of these parameters. An example is presented hereinafter for the case of an SLF-PLL with $N = 200$, $f_{\text{ref}} = 10$ MHz, $I_{\text{CP}} = 2$ mA, $K_{\text{vco}} = 2\pi \cdot 120 \cdot 10^6$ rad/(V $\cdot$ s), $C_p = 2.53$ pF, $C_s = 328$ pF, $R_s = 5408\ \Omega$, $C_x = 795$ fF, $R_x = 20$ kΩ, $\lambda = 0.5$, $t_{\text{op1}} = 50$ ns, $t_{\text{op2}} = 10$ ns, and $t_{\text{cl}} = 40$ ns.

To apply the single-rate version of the model, it is first necessary to calculate $F_{\text{SLF}}(z)$, and to apply the multirate version of the model, it is first necessary to calculate $G_{\text{SLF}}(z)$. Appendix A provides the equations required to calculate

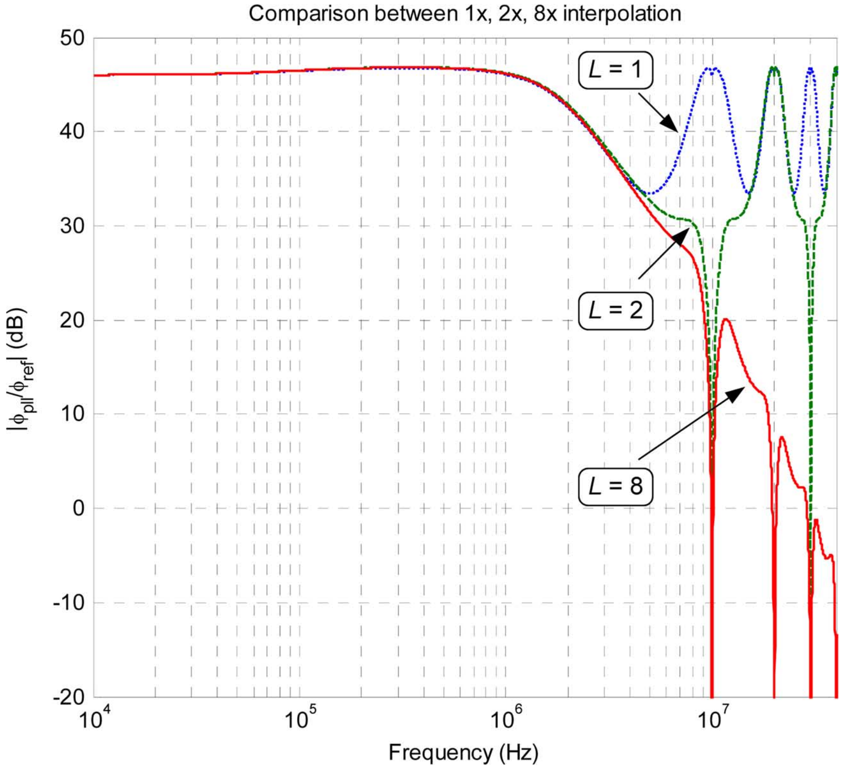


Fig. 6. Comparison between the transfer function for the single-rate $(L = 1)$ case and the transfer functions for the multirate $(L = 2, L = 8)$ case.

$F_{\mathrm{SLF}}(z)$ and $G_{\mathrm{SLF}}(z)$ starting from the LF design values. Executing Steps 1)–6) in Appendix A with the LF design values listed earlier yields

$$F_{\mathrm{SLF}}(z) = 225\frac{(1.06 - z^{-1})(-4.74 - z^{-1})(-67749 - z^{-1})}{(1 - z^{-1})(22.9 - z^{-1})(775 - z^{-1})}. \quad (13)$$

In addition, executing Steps 7) and 8) in Appendix A for $L = 2$ yields

$$F_{\mathrm{SLF},0}(z) = F_{\mathrm{SLF}}(z) \quad (14)$$

where $F_{\mathrm{SLF}}(z)$ is given by (13) and

$$F_{\mathrm{SLF},1}(z) = -637335\frac{(1.06 - z^{-1})(-139 - z^{-1})}{(1 - z^{-1})(22.9 - z^{-1})(775 - z^{-1})}. \quad (15)$$

Substituting (14) and (15) into (12) yields $G_{\mathrm{SLF}}(z)$ for $L = 2$. The same procedure can be used to obtain $G_{\mathrm{SLF}}(z)$ for any positive integer L.

These $F_{\mathrm{SLF}}(z)$ and $G_{\mathrm{SLF}}(z)$ functions can be used in the versions of the model shown in Figs. 3 and 5, respectively, to analyze the locked behavior of the SLF-PLL. As described earlier, the model implies that the loop gain of the PLL is given by (3). Substituting (13) into (3) and solving for the unity-gain frequency indicate that the LBW of the PLL is 1 MHz, and it follows from (4) that the PM of the PLL is 60°. Figs. 6–8 show various additional aspects of the behavior of the SLF-PLL as predicted by the two versions of the model.

Fig. 6 shows the phase noise transfer function from the reference signal input, i.e., the squared magnitude of (5) in decibels with $z = \exp(j2\pi f/f_{\mathrm{ref}})$, as predicted by the single-rate version of the model $(L = 1)$ and the multirate version of the model for $L = 2$ and $L = 8$. As expected, there is little deviation among the predicted transfer functions for frequencies below $f_{\mathrm{ref}}/4$, and each transfer function is periodic with a period of Lf_{ref}. In general, the larger the value of L, the higher the maximum frequency at which the transfer function predicted by the model accurately represents that of the actual SLF-PLL. The PLL bandwidth is relatively wide in this example, so the transfer function is not highly attenuated at $f_{\mathrm{ref}}/4$. In such cases, the multirate version of the model provides useful information.

Fig. 7 shows plots of the squared magnitudes of (5)–(7) in decibels with $z = \exp(j2\pi f/f_{\mathrm{ref}})$ as predicted by the model with $L = 8$ and the corresponding transfer functions as predicted by computer simulation. The plots suggest that model agrees well with the simulation. The one exception is that the transfer functions corresponding to (6) deviate somewhat at low frequencies, but this has been traced to limitations of the simulator.

Fig. 8 shows a time-domain plot of the simulated VCO output phase corresponding to a reference signal phase step, and the corresponding sample values predicted by the model. As expected, the sample values predicted by the model fall precisely on the simulated curve.

C. Synthesis Problem

As shown earlier, the proposed model allows for straightforward analysis of an SLF-PLL given the PLL design parameters, i.e., given N, f_{ref}, I_{CP}, K_{vco}, C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. However, designers are often faced with the

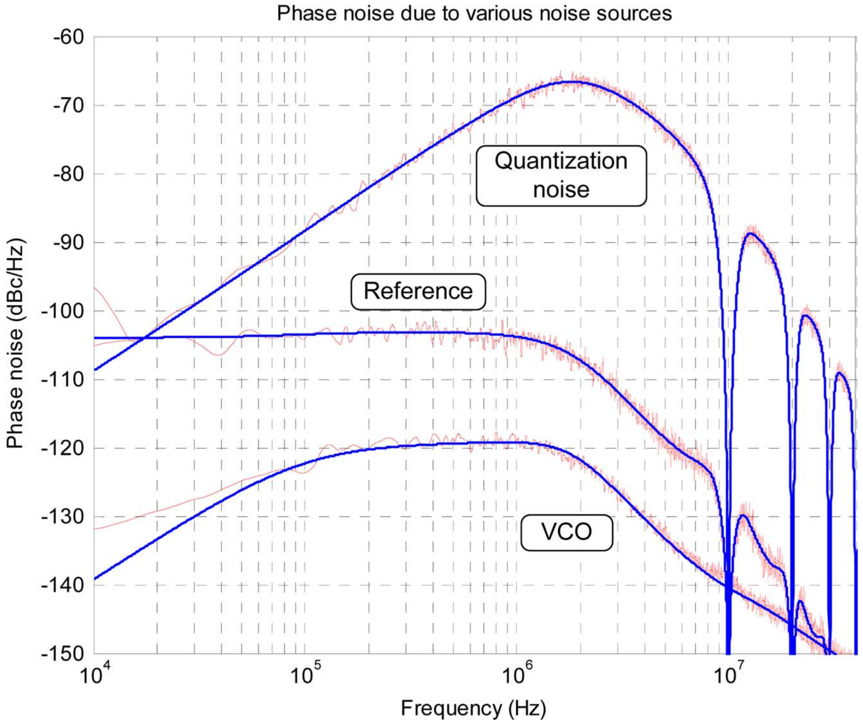


Fig. 7. Comparison between the model and simulated results for various noise sources for a fractional-N PLL $(L = 8)$.

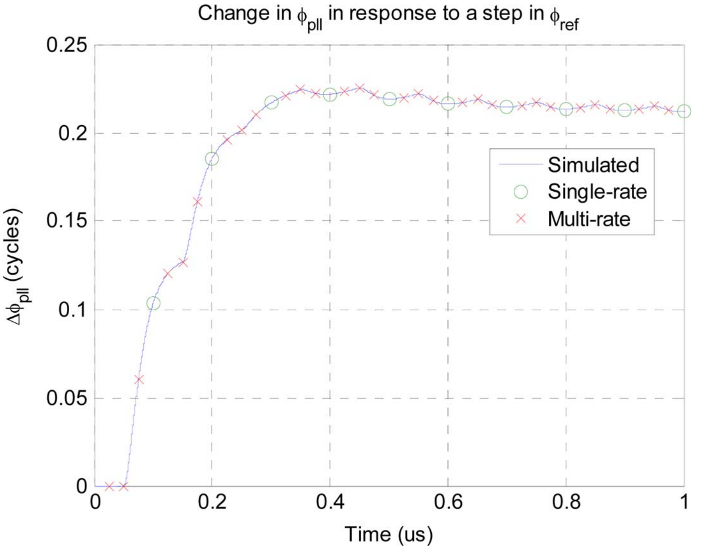


Fig. 8. Simulated VCO output phase of the SLF-PLL corresponding to a reference signal phase step, and the corresponding sample values predicted by the model.

synthesis problem of choosing the SLF component values, i.e., C_p, C_s, C_x, R_s, and R_x, such that the PLL has a desired LBW and PM. Typically, N, f_{ref}, I_{CP}, K_{vco}, λ, t_{op1}, t_{cl}, and t_{op2} are known prior to choosing the SLF component values because they depend on circuit-level considerations and application requirements.

The model equations could be solved numerically to provide the SLF component values in terms of the other PLL design parameters and the desired LBW and PM, but it is simpler to use the following iterative approach. The first step is to choose the LF component values for a conventional CLF-PLL that approximately achieves the desired LBW and PM. Approximate equations that provide the values of C_p, C_s, and R_s for a conventional CLF-PLL in the absence of C_x and R_x are well known [12]. Typically, designers use these equations to find C_p, C_s, and R_s and then choose C_x and R_x such that the extra pole they introduce has a high-enough frequency in that it negligibly affects the LBW and PM. The second step is to iteratively adjust the values of C_p, C_s, and R_s to compensate for the sampling operation in the SLF using the proposed SLF-PLL model to guide the iteration process.

As observed in [4], the sampling operation in an SLF decreases the PM of the PLL by approximately the product of the LBW and the duration over which the switch is open each reference period, i.e.,

$$PM_{\text{SLF}} = PM_{\text{CLF}} - \omega_{\text{LBW}}(t_{\text{op1}} + t_{\text{op2}}) \tag{16}$$

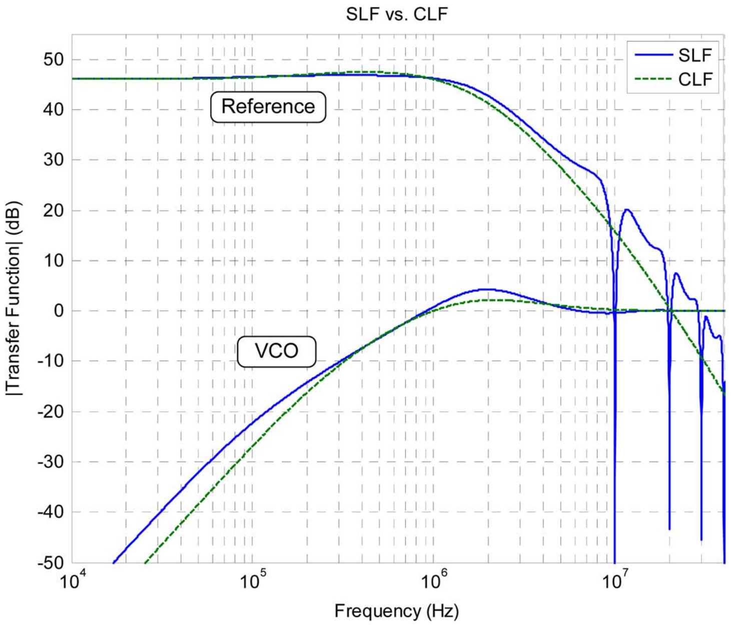


Fig. 9. Reference and VCO phase noise transfer functions of the original CLF-PLL and the SLF-PLL obtained via the synthesis procedure.

where ω_{LBW} is the LBW. This loss in PM can be addressed by increasing the ratio C_s/C_p. Increasing C_s moves one of the filter zeros to a lower frequency, but C_s typically is large, so moving the zero significantly requires a significant increase in the circuit area. Decreasing C_p moves one of the filter poles to a higher frequency, but this has the disadvantage of reducing the high-frequency attenuation of the loop. Experimentally, a combination of these two adjustments yields the best tradeoff between area and high-frequency attenuation. The approach is to iteratively adjust C_s and C_p and, at each iteration step, use the proposed SLF-PLL model to evaluate whether further adjustment of C_s and C_p is necessary. A similar iterative process can be used to optimize the choices of C_x and R_x if necessary.

The amount by which the sampling operation affects the behavior of the SLF-PLL depends, to a large extent, on the LBW. If the LBW is sufficiently low, the LF components obtained in the first step mentioned earlier for the CLF-PLL can be used in the SLF-PLL with only a minor degradation of the PM. Nevertheless, in such cases, the proposed SLF-PLL model is useful to verify that no further adjustment is necessary.

D. Synthesis Example

Consider an integer-N SLF-PLL for which $N = 200$, $f_{\text{ref}} = 10$ MHz, $I_{\text{CP}} = 2$ mA, $K_{\text{vco}} = 2\pi \cdot 120 \cdot 10^6$ rad/(V $\cdot$ s), $\lambda = 0.5$, $t_{\text{op1}} = 50$ ns, $t_{\text{cl}} = 40$ ns, and $t_{\text{op2}} = 10$ ns. Suppose that it is desired to choose C_p, C_s, C_x, R_s, and R_x, such that the LBW is 1 MHz and the PM is 60°.

The first step of the procedure described earlier is to choose C_p, C_s, C_x, R_s, and R_x for a corresponding conventional CLF-PLL. Applying the equations in [12] with $C_x = 795$ fF, $R_x = 20$ kΩ, an LBW of 1.015 MHz, and a PM of 67° yields

$$C_p = 6\,\text{pF} \quad C_s = 139\,\text{pF} \quad R_s = 5543\,\Omega. \tag{17}$$

Note that the LBW and PM have both been increased relative to the target values of 1 MHz and 60°, respectively, to approximately account for the effects of C_x and R_x which are neglected by the equations in [12].

If the values in (17) are used without modification in the SLF-PLL, the resulting LBW and PM are 960 kHz and 44°, respectively. Iteratively adjusting C_p and C_s and, to a lesser extent, R_s, as described earlier, indicates that the SLF-PLL achieves the target LBW and PM with

$$C_p = 2.53\,\text{pF} \quad C_s = 328\,\text{pF} \quad R_s = 5408\,\Omega. \tag{18}$$

Fig. 9 shows the phase noise transfer functions from the reference signal and the VCO for both the CLF-PLL and the SLF-PLL in the aforementioned design example. Although the corresponding transfer functions of the two PLLs are similar, some differences are evident. The difference between the transfer functions from the reference signal occurs because C_p in (18) is less than half of C_p in (17). The difference between the transfer functions from the VCO occurs because $C_p + C_s$ in (18) is greater than that of (17). These differences are exaggerated because of the high LBW in this example. A lower LBW would result in less significant differences between the two sets of curves.

Fig. 10 shows a comparison of SLF-PLLs and CLF-PLLs using the same LF components. Two cases are examined: a low-LBW (200 kHz) design and a high-LBW (1 MHz) design. In each case, $f_{\text{ref}} = 10$ MHz, and for the SLF-PLL, $\{t_{\text{op1}}, t_{\text{op2}}, t_{\text{cl}}\} = \{50\text{ ns}, 10\text{ ns}, 40\text{ ns}\}$. The results demonstrate that it is reasonable to use the component values derived for a CLF-PLL in an SLF-PLL when $\omega_{\text{LBW}}(t_{\text{op1}} + t_{\text{op2}})$ is small.

IV. Derivation of the PLL Model

A. Background Results

Once the PLL is locked, the output of the VCO can be modeled as $v_{\text{vco}}(t) = A(t)\sin(\omega_{\text{pll}}t + \phi_{\text{pll}}(t))$, where $A(t)$ is some

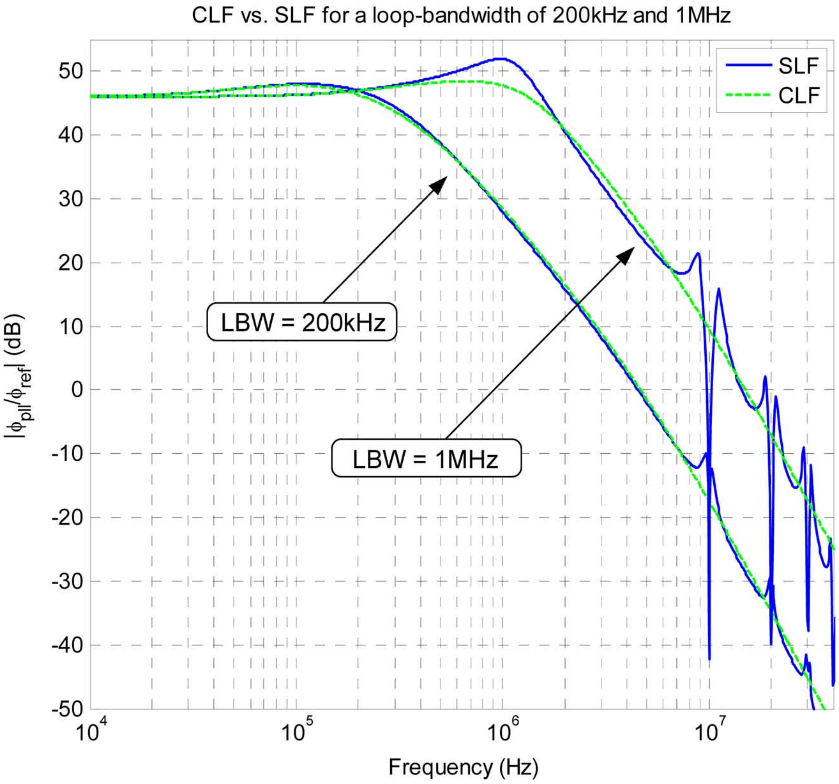


Fig. 10. Reference phase noise transfer functions of the CLF-PLL and SLF-PLL for low and high LBWs.

nonzero positive waveform and $\omega_{\text{pll}} = 2\pi(N+\alpha)f_{\text{ref}}$ is the ideal output frequency of the PLL. The PLL's total phase noise is given by

$$\phi_{\text{pll}}(t) = \phi_{\text{ctrl}}(t) + \phi_{\text{vco}}(t) \tag{19}$$

where $\phi_{\text{ctrl}}(t)$ is the phase noise caused by deviations of the VCO control voltage from its mean value and $\phi_{\text{vco}}(t)$ is the open-loop VCO phase noise, i.e., the phase noise that would remain if the VCO control voltage were held constant. Therefore

$$\phi_{\text{ctrl}}(t) = K_{\text{vco}} \int_0^t (v_{\text{ctrl}}(\tau) - \bar{v}_{\text{ctrl}})d\tau \tag{20}$$

where K_{vco} is the VCO gain and is the voltage for which the free-running frequency of the VCO would be exactly ω_{pll} in the absence of $\phi_{\text{vco}}(t)$.

Suppose that the PLL is already locked at time $t = 0$. Let τ_n be the time of the nth rising edge of the reference signal, and let t_n be the corresponding rising edge of the divider output for $n = 0, 1, 2, \ldots$. As shown in [8], the net charge delivered to or removed from the LF by the CP during the nth reference period is

$$\begin{aligned} Q_{\text{cp}}[n] = &\frac{I_{\text{CP}}T_{\text{ref}}}{2\pi} \\ &\times \left[\frac{2\pi\sum_{k=0}^{n}\varepsilon_q[k] - \phi_{\text{ctrl}}(t_n) - \phi_{\text{vco}}(t_n)}{N+\alpha}\right. \\ &\left. + Q_{\text{ref}}(\tau_n)\right] \end{aligned} \tag{21}$$

where I_{CP} is the magnitude of current pulses sourced and sunk by the CP, $\varepsilon_q[k] = y[k] - \alpha$ is the quantization noise from the noise-shaping quantizer, and $\phi_{\text{ref}}(t)$ is the phase noise of the reference signal.

B. Derivation of the Single-Rate Version of the Model

The state of the SLF and $\phi_{\text{ctrl}}(t)$ at time t can be represented together as a vector $\mathbf{x}(t)$ given by

$$\mathbf{x}(t) = [q_T(t)\ q_s(t)\ q_x(t)\ \phi_{\text{ctrl}}(t)]^{\text{T}} \tag{22}$$

where $q_T(t)$ is the total charge on all the LF capacitors, $q_s(t)$ is the charge on C_s, and $q_x(t)$ is the charge on C_x, all at time t. Let $\mathbf{x}[n]$, for $n = 0, 1, 2, \ldots$, be a sampled version of $\mathbf{x}(t)$, defined as

$$\mathbf{x}[n] = \mathbf{x}(nT_{\text{ref}} + t_{\text{op1}}). \tag{23}$$

As proven in the next section

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}Q_{\text{cp}}[n] \tag{24}$$

$$\phi_{\text{ctrl}}(nT_{\text{ref}}) = \mathbf{C}\mathbf{x}[n-1] \tag{25}$$

where $\mathbf{B} = [1\,0\,0\,0]^{\text{T}}$, and $\mathbf{A}$ and $\mathbf{C}$ are a 4×4 matrix and a 1×4 vector, respectively. The elements of $\mathbf{A}$ and $\mathbf{C}$ are fixed numbers that depend only on the LF component values, t_{op1}, t_{cl}, and t_{op2}.

In a practical PLL, $\phi_{\text{ctrl}}(t)$ has a bandwidth that is less than a tenth of the reference frequency and $t_n \approx nT_{\text{ref}}$ for $n = 0, 1, 2, \ldots$, so it follows that

$$\phi_{\text{ctrl}}(t_n) \approx \phi_{\text{ctrl}}(nT_{\text{ref}}) \tag{26}$$

to a good approximation [8]. Consequently, (24) and (25) provide an expression for $\phi_{\text{ctrl}}(t_n)$ in terms of $Q_{\text{cp}}[n]$.

Equations (24) and (25) are called state-space equations [13]. They indicate that $\phi_{\text{ctrl}}(nT_{\text{ref}})$ is the output of an LTI discrete-time system with input $Q_{\text{cp}}[n]$. Appendix A provides equations with which to obtain $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, and well-known techniques are available to calculate the transfer function once $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are known. For example, the built-in Matlab command, i.e., $\text{ss2zp}(\mathbf{A}, \mathbf{B}, \mathbf{C}, 0, 1)$, can be used. The result is a four-pole three-zero function which can be written as

$$K_{\text{vco}} F_{\text{SLF}}(z) \frac{z^{-1}}{1 - z^{-1}} \tag{27}$$

where $F_{\text{SLF}}(z)$ has the form given by (2).

The single-rate model shown in Fig. 4 follows directly from and is a graphical representation of (19), (21), (26), and (27). The aforementioned derivation applies to fractional-N PLLs. However, by setting $\alpha = 0$ and $\varepsilon_q[n] = 0$, it also applies to integer-N PLLs.

The only approximation made in the model's derivation is (26). The standard model for conventional PLLs also relies on this approximation. However, in contrast to the model presented in this paper, the standard model for conventional PLLs relies on several additional approximations.

C. Derivation of the Single-Rate State-Space Equations

Without loss of generality, $v_{\text{ctrl}}(t)$ can be taken to have zero mean so that (20) reduces to

$$\phi_{\text{ctrl}}(t) = K_{\text{vco}} \int_0^t v_{\text{ctrl}}(\tau) d\tau. \tag{28}$$

This simplifies the notation of the following derivation, yet it can be verified that it does not change the results of the derivation.

The SLF is a time-varying circuit, but during any time interval over which the switch either remains closed or remains open, it reduces to an LTI system. Furthermore, it follows from (28) that $\phi_{\text{ctrl}}(t)$ is an LTI function of the LF output.

Suppose that the switch is closed for the time interval from t_0 to $t_0 + \Delta t$. As described in Section II, $I_{\text{cp}}(t)$ is zero when the switch is closed, so the total charge in the SLF remains unchanged during this time interval, i.e.,

$$q_T(t_0 + \Delta t) = q_T(t_0). \tag{29}$$

Given that the system is linear and time invariant over the interval, well-known linear systems theory results can be invoked (see, e.g., Appendix B) to write the other elements of (22) at time $t_0 + \Delta t$ in terms of their values at time t_0 as

$$\begin{aligned} q_s(t_0 + \Delta t) &= r_{2,1}(\Delta t) q_T(t_0) + r_{2,2}(\Delta t) q_s(t_0) \\ &\quad + r_{2,3}(\Delta t) q_x(t_0) \qquad (30) \\ q_x(t_0 + \Delta t) &= r_{3,1}(\Delta t) q_T(t_0) + r_{3,2}(\Delta t) q_s(t_0) \\ &\quad + r_{3,3}(\Delta t) q_x(t_0) \qquad (31) \\ \phi_{\text{ctrl}}(t_0 + \Delta t) &= r_{3,1}(\Delta t) q_T(t_0) + r_{3,2}(\Delta t) q_s(t_0) \\ &\quad + r_{3,3}(\Delta t) q_x(t_0) + \phi_{\text{ctrl}}(t_0) \qquad (32) \end{aligned}$$

where each $r_{i,j}(t)$ is an LTI system impulse response. For instance, $r_{3,2}(t)$ is the charge on C_x as a function of t in response to a Dirac delta function current impulse, $I(t) = \delta(t)$, injected across the terminals of C_s for the case in which the charge on each capacitor is zero for $t < 0$. In particular, the time-invariance property of (28) and the SLF over the time interval imply that the $r_{i,j}(\Delta t)$ factors in (30)–(32) depend only on the duration of the interval but not on the start time t_0 of the interval.

Equations (29)–(32) can be written more compactly as

$$\mathbf{x}(t_0 + \Delta t) = \mathbf{H_{cl}}(\Delta t) \mathbf{x}(t_0), \tag{33}$$

where $\mathbf{x}(t)$ is given by (22) and

$$\mathbf{H_{cl}}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ r_{2,1}(t) & r_{2,2}(t) & r_{2,3}(t) & 0 \\ r_{3,1}(t) & r_{3,2}(t) & r_{3,3}(t) & 0 \\ r_{4,1}(t) & r_{4,2}(t) & r_{4,3}(t) & 1 \end{bmatrix}. \tag{34}$$

Now, suppose that the switch is open for the time interval from t_1 to $t_1 + \Delta\tau$. As described in Section II, $I_{\text{cp}}(t)$ is not necessarily zero when the switch is open, so

$$q_T(t_1 + \Delta\tau) = q_T(t_1) + \int_{t_1}^{t_1+\Delta\tau} I_{\text{cp}}(t) dt. \tag{35}$$

However, it follows from Fig. 2(b) that, when the switch is open, $I_{\text{cp}}(t)$ does not affect the other elements of (22). Therefore, equations for the other elements of (22) that apply to the case in which the switch is open can be obtained by exactly the same reasoning that led to (30)–(32). These equations, along with (35), can be written as

$$\mathbf{x}(t_1 + \Delta\tau) = \mathbf{H_{op}}(\Delta\tau) \mathbf{x}(t_1) + \mathbf{B} \int_{t_1}^{t_1+\Delta\tau} I_{\text{cp}}(t) dt \tag{36}$$

where $\mathbf{B} = [1\ 0\ 0\ 0]^{\mathrm{T}}$

$$\mathbf{H_{op}}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ s_{2,1}(t) & s_{2,2}(t) & s_{2,3}(t) & 0 \\ s_{3,1}(t) & s_{3,2}(t) & s_{3,3}(t) & 0 \\ s_{4,1}(t) & s_{4,2}(t) & s_{4,3}(t) & 1 \end{bmatrix} \tag{37}$$

and each $s_{i,j}(t)$ is an LTI system impulse response. Specifically, the expression for each $s_{i,j}(t)$ is identical to that of the corresponding $r_{i,j}(t)$ except with C_p replaced by $(1 - \lambda)C_p$.

These results can be combined to prove (24) and (25). It follows from Fig. 2(c), (23), (33), and (36) that

$$\begin{aligned} \mathbf{x}(nT_{ref} + t_{\text{op1}} + t_{\text{cl}}) &= \mathbf{H_{cl}}(t_{\text{cl}}) \mathbf{x}[n] \qquad (38) \\ \mathbf{x}[n+1] &= \mathbf{H_{op}}(t_{\text{op1}} + t_{\text{op2}}) \mathbf{x}(nT_{\text{ref}} + t_{\text{op1}} + t_{\text{cl}}) \\ &\quad + \mathbf{B} Q_{\text{cp}}[n]. \qquad (39) \end{aligned}$$

Substituting (38) into (39) yields (24) with

$$\mathbf{A} \triangleq \mathbf{H_{op}}(t_{\text{op1}} + t_{\text{op2}}) \mathbf{H_{cl}}(t_{\text{cl}}). \tag{40}$$

Similar reasoning leads to (25) with

$$\mathbf{C} \triangleq [0\ 0\ 0\ 1] \mathbf{H_{op}}(t_{\text{op2}}) \mathbf{H_{cl}}(t_{\text{cl}}). \tag{41}$$

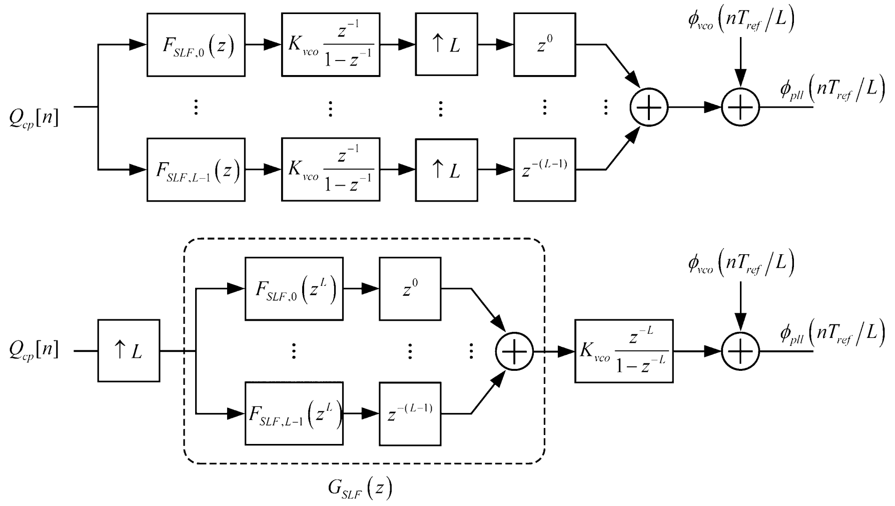


Fig. 11. Model for the multirate SLF and transformation to $G_{\mathrm{SLF}}(z)$.

D. Extension to the Multirate Version of the Model

Nearly identical reasoning to that presented earlier which led to (25) and (41) also implies that

$$\phi_{\mathrm{ctrl}}((n-1)T_{\mathrm{ref}}+t_{\mathrm{op1}}+\Delta t)=[0\,0\,0\,1]\mathbf{D}(\Delta t)\mathbf{x}[n-1] \tag{42}$$

for Δt in the range $0<\Delta t\le T_{\mathrm{ref}}$, where

$$\mathbf{D}(\Delta t)=\begin{cases}\mathbf{H_{cl}}(\Delta t), & \text{if } 0<\Delta t\le t_{\mathrm{cl}}\\ \mathbf{H_{op}}(\Delta t-t_{\mathrm{cl}})\mathbf{H_{cl}}(t_{\mathrm{cl}}), & \text{if } t_{\mathrm{cl}}<\Delta t\le T_{\mathrm{ref}}.\end{cases} \tag{43}$$

Therefore, (42) and (43) can be used to obtain any sample of $\phi_{\mathrm{ctrl}}(t)$ in the range $(n-1)T_{\mathrm{ref}}+t_{\mathrm{op1}}<t\le nT_{\mathrm{ref}}+t_{\mathrm{op1}}$.

In particular

$$\phi_{\mathrm{ctrl}}\left(\left(n+\frac{i-d}{L}\right)T_{\mathrm{ref}}\right)=\mathbf{C}_i\mathbf{x}[n-1] \tag{44}$$

for $i=0,2,\ldots,L-1$, where

$$\mathbf{C}_i=\mathbf{D}\left(t_{\mathrm{cl}}+t_{\mathrm{op2}}+\frac{i-d}{L}T_{\mathrm{ref}}\right) \tag{45}$$

and d is given by (10). For each value of i, (44) defines an LTI filter with input $Q_{\mathrm{cp}}[n]$ and output samples given by (44) for $n=0,1,2,\ldots$. The transfer function of the ith of such filter has the form

$$K_{\mathrm{vco}}F_{\mathrm{SLF},i}(z)\frac{z^{-1}}{1-z^{-1}} \tag{46}$$

where $F_{\mathrm{SLF},i}(z)$ is obtained in the same way that $F_{\mathrm{SLF}}(z)$ is obtained from $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, as described in Section III-B, except with $\mathbf{C}$ replaced by $\mathbf{C}_i$. Note, in particular, that $\mathbf{C}_d=\mathbf{C}$ by definition, so $F_{\mathrm{SLF},d}(z)=F_{\mathrm{SLF}}(z)$.

It follows that the SLF and VCO can be modeled as shown in Fig. 11(a). With the noble identity for upsampling, this can be redrawn as shown in Fig. 11(b) which is equivalent to the system shown in Fig. 4(b), with $G_{\mathrm{SLF}}(z)$ as given by (12).

Appendix A

This Appendix describes all the calculations necessary to obtain $F_{\mathrm{SLF}}(z)$ and $G_{\mathrm{SLF}}(z)$ starting from the values of T_{ref}, C_p, C_s, C_x, R_s, R_x, λ, t_{op1}, t_{cl}, and t_{op2}. The calculations are most easily implemented via a computer calculation script executed by a software tool such as Matlab. Therefore, the calculations are listed hereinafter in the form of specific steps that must be executed by such a calculation script. Steps 1)–6) specify the calculation details of $F_{\mathrm{SLF}}(z)$ in (2). Steps 1)–5) followed by Steps 7)–8) specify the calculation details of $G_{\mathrm{SLF}}(z)$ in (12).

1) Define the following functions of the variables C_p, C_s, R_s, C_x, R_x, and Γ:

$$p_{1,\Gamma}=\frac{-b_\Gamma+\sqrt{b_\Gamma^2-4a_\Gamma c_\Gamma}}{2a_\Gamma}$$
$$p_{1,\Gamma}=\frac{-b_\Gamma-\sqrt{b_\Gamma^2-4a_\Gamma c_\Gamma}}{2a_\Gamma}$$

where

$$\begin{aligned}a_\Gamma&=C_p(1-\Gamma)R_sR_xC_sC_x\\ b_\Gamma&=(C_p(1-\Gamma)+C_x)R_sC_s\\ &\quad+(C_p(1-\Gamma)+C_s)R_xC_x\\ c_\Gamma&=C_p(1-\Gamma)+C_s+C_x.\end{aligned}$$

2) Define the following functions of the variables C_u, C_v, R_u, R_v, and Γ and the functions defined in Step 1):

$$\begin{aligned}E_{u,v,\Gamma}&=(1-\Gamma)C_u(p_{1,\Gamma}-p_{2,\Gamma})\\ F_{u,v,\Gamma}&=(1-\Gamma)C_up_{2,\Gamma}(1+p_{1,\Gamma}R_vC_v)\\ G_{u,v,\Gamma}&=-(1-\Gamma)C_up_{1,\Gamma}(1+p_{2,\Gamma}R_vC_v)\\ I_{u,v,\Gamma}&=\Gamma C_u(p_{1,\Gamma}-p_{2,\Gamma})\\ J_{u,v,\Gamma}&=C_up_{2,\Gamma}((1+p_{1,\Gamma}R_vC_v)\\ &\qquad\times(\Gamma+p_{1,\Gamma}R_uC_p(1-\Gamma))+p_{1,\Gamma}R_uC_v)\\ L_{u,v,\Gamma}&=-C_up_{1,\Gamma}((1+p_{2,\Gamma}R_vC_v)\\ &\qquad\times(\Gamma+p_{2,\Gamma}R_uC_p(1-\Gamma))+p_{2,\Gamma}R_uC_v)\end{aligned}$$

$$P_{u,v,\Gamma} = C_u p_{2,\Gamma}(\Gamma(1 + p_{1,\Gamma}R_vC_v) - p_{1,\Gamma}R_vC_v)$$
$$Q_{u,v,\Gamma} = -C_u p_{1,\Gamma}(\Gamma(1 + p_{2,\Gamma}R_vC_v) - p_{2,\Gamma}R_vC_v).$$

3) Define the following 4 × 4 matrix function of the variable t and the functions defined in Steps 1) and 2):

$$\mathbf{H}_\Gamma(t) = K_\Gamma \begin{bmatrix} \frac{1}{K_\Gamma} & 0 & 0 & 0 \\ h_{2,1,\Gamma}(t) & h_{2,2,\Gamma}(t) & h_{2,3,\Gamma}(t) & 0 \\ h_{3,1,\Gamma}(t) & h_{3,2,\Gamma}(t) & h_{3,3,\Gamma}(t) & 0 \\ h_{4,1,\Gamma}(t) & h_{4,2,\Gamma}(t) & h_{4,3,\Gamma}(t) & \frac{1}{K_\Gamma} \end{bmatrix} \tag{47}$$

where

$$K_\Gamma = -\frac{1}{(C_p(1-\gamma) + C_s + C_x(p_{1,\Gamma} - p_{2,\Gamma})}$$
$$h_{2,1,\Gamma}(t) = E_{s,x,\Gamma} + F_{s,x,\Gamma}e^{p_{1,\Gamma}t} + G_{s,x,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{2,2,\Gamma}(t) = I_{s,x,\Gamma} + J_{s,x,\Gamma}e^{p_{1,\Gamma}t} + L_{s,x,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{2,3,\Gamma}(t) = I_{s,x,\Gamma} + P_{s,x,\Gamma}e^{p_{1,\Gamma}t} + Q_{s,x,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,1,\Gamma}(t) = E_{x,s,\Gamma} + F_{x,s,\Gamma}e^{p_{1,\Gamma}t} + G_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,2,\Gamma}(t) = I_{x,s,\Gamma} + P_{x,s,\Gamma}e^{p_{1,\Gamma}t} + Q_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{3,3,\Gamma}(t) = I_{x,s,\Gamma} + J_{x,s,\Gamma}e^{p_{1,\Gamma}t} + L_{x,s,\Gamma}e^{p_{2,\Gamma}t}$$
$$h_{4,1,\Gamma}(t) = \frac{1}{C_x}\left(E_{x,s,\Gamma}t + \frac{F_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{G_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{2,\Gamma}t} - 1\right)\right)$$
$$h_{4,2,\Gamma}(t) = \frac{1}{C_x}\left(I_{x,s,\Gamma}t + \frac{P_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{Q_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{2,\Gamma}t} - 1\right)\right)$$
$$h_{4,3,\Gamma}(t) = \frac{1}{C_x}\left(I_{x,s,\Gamma^t} + \frac{J_{x,s,\Gamma}}{p_{1,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right) + \frac{L_{x,s,\Gamma}}{p_{2,\Gamma}}\left(e^{p_{1,\Gamma}t} - 1\right)\right).$$

Therefore, with the functions defined in Steps 1) and 2) substituted into the functions defined in Step 3), $\mathbf{H}_\Gamma(t)$ is a matrix function of the variables $C_p,\ C_s,\ R_s,\ C_x,\ R_x,\ \Gamma$, and t.

4) Define the following 4 × 4 matrix functions of $C_p,\ C_s,\ R_s,\ C_x,\ R_x$, and t using the matrix function defined in Step 4):

$$\mathbf{H_{op}}(t) = \mathbf{H}_\Gamma(t)|_{\Gamma=\lambda} \text{ and } \mathbf{H_{cl}}(t) = \mathbf{H}_\Gamma(t)|_{\Gamma=0}.$$

5) With numerical values for $C_p,\ C_s,\ C_x,\ R_s,\ R_x,\ \lambda,\ t_{\text{op1}},\ t_{\text{cl}}$, and t_{op2} and the matrix functions from Step 4), calculate the following 4 × 4 matrix of numbers:

$$\mathbf{A} = \mathbf{H_{op}}(t_{\text{op1}} + t_{\text{op2}}\mathbf{H_{cl}}(t_{\text{cl}})$$

and the following 1 × 4 vector of numbers:

$$\mathbf{C}[0\ 0\ 0\ 1]\mathbf{H_{op}}(t_{\text{op2}})\mathbf{H_{cl}}(t_{\text{cl}}).$$

6) Calculate the poles, zeros, and scale factor of $F_{\text{SLF}}(z)$ in (2) from its state-space representation as specified by the matrix $\mathbf{A}$ and vector $\mathbf{C}$ from Step 5) and $\mathbf{B} = [1\ 0\ 0\ 0]^{\mathbf{T}}$. For example, this can be done using the Matlab function

$$\text{ss2zp}(\mathbf{A}, \mathbf{B}, \mathbf{C}, 0, 1).$$

7) With numerical values for $C_p,\ C_s,\ C_x,\ R_s,\ R_x,\ \lambda,\ T_{\text{ref}},\ t_{\text{op2}}$, and t_{cl} and the matrix functions from Step 4), calculate the following 1 × 4 vectors of numbers for $i = 0, 1, \ldots, L-1$:

$$\mathbf{C}_i[0\ 0\ 0]]\mathbf{D}\left(t_{\text{cl}} + t_{\text{op2}} + \frac{i-d}{L}T_{\text{ref}}\right)$$

where

$$\mathbf{D}(\Delta t) = \begin{cases} \mathbf{H_{cl}}(\Delta t), & \text{if } 0 < \Delta t \le t_{\text{cl}} \\ \mathbf{H_{op}}(\Delta t - t_{\text{cl}})\mathbf{H_{cl}}(t_{\text{cl}}), & \text{if } t_{\text{cl}} < \Delta t \le T_{\text{ref}} \end{cases}$$
$$d = \left\lfloor \frac{t_{\text{cl}}t_{\text{op2}}}{T_{\text{ref}}} \right\rfloor.$$

8) For each $i = 0, 1, \ldots, L-1$, calculate the poles, zeros, and scale factor of each function $F_{\text{SLF},i}(z)$ [which has the same form as (2)] from its state-space representation as specified by the matrix $\mathbf{A}$ from Step 5), the vector $\mathbf{C}_i$ from Step 7), and $\mathbf{B} = [1\ 0\ 0\ 0]^{\text{T}}$. For example, this can be done using the Matlab function

$$\text{ss2zp}(\mathbf{A}, \mathbf{B}, \mathbf{C}_i, 0, 1).$$

Substitute the L resulting $F_{\text{SLF},i}(z)$ functions into (12) to obtain $G_{\text{SLF}}(z)$.

Appendix B

This Appendix derives (31) and (32) to find expressions for the third and fourth row elements of $\mathbf{H_{cl}}(t)$ in (34). The derivation of (30) is not presented because it is almost identical to that of (31).

Let C_{p_A} denote the capacitor to the left of the switch, let C_{p_B} denote the capacitor to the immediate right of the switch in Fig. 2(b), and let their respective charges at t_0 be $q_{p_A}(t_0)$ and $q_{p_B}(t_0)$. Then, the charge on $q_x(t_0 + \Delta t)$ can be written as

$$q_x(t_0 + \Delta t) = h_1(\Delta t)q_{p_A}(t_0) + h_2(\Delta t)q_{p_B}(t_0) + h_3(\Delta t)q_s(t_0) + h_4(\Delta t)q_x(t_0) \tag{48}$$

where $h_1(\Delta t)$ is the charge transfer function from capacitor C_{p_A} to capacitor C_x, $h_2(\Delta t)$ is the charge transfer function from capacitor C_{p_B} to capacitor C_x, $h_3(\Delta t)$ is the charge transfer function from capacitor C_s to capacitor C_x, and $h_4(\Delta t)$ is the charge transfer function from capacitor C_x to capacitor C_x, all over a time interval of Δt. These charges can be expressed in terms of the elements of $\mathbf{x}(t_0)$ as

$$q_{p_A}(t_0) = [q_r(t_0) - q_s(t_0) - q_x(t_0)]\lambda$$
$$q_{p_B}(t_0) = [q_r(t_0) - q_s(t_0) - q_x(t_0)](1-\lambda). \tag{49}$$

Substituting (49) into (48) leads to (31) with

$$r_{3,1}(\Delta t) = \lambda h_1(\Delta t) + (1-\lambda)h_2(\Delta t)$$
$$r_{3,2}(\Delta t) = h_3(\Delta t) - \lambda h_1(\Delta t) - (1-\lambda)h_2(\Delta t)$$
$$r_{3,3}(\Delta t) = h_4(\Delta t) - \lambda h_1(\Delta t) - (1-\lambda)h_2(\Delta t). \tag{50}$$

The $h_i(\Delta t)$ functions can be found by computing the inverse Laplace transform of the s-domain charge transfer function from any one of the capacitors to any other and then evaluating the result at $t = \Delta t$. For example, suppose that the switch is closed $(\lambda = 0)$ and consider $h_3(\Delta t)$. In the s-domain, the charge on capacitor C_x due to charge on capacitor C_s is given by

$$\begin{aligned} Q_x(s) &= C_x V_x(x) = C_x \frac{V_x(s)}{I_s(s)} I_s(s) \\ &= \frac{C_x}{(C_p + C_s + C_x)s\left(1 - \frac{s}{p_1}\right)\left(1 - \frac{s}{p_2}\right)} q_s(0) \end{aligned} \tag{51}$$

where V_x and I_x are the s-domain voltage and current associated with capacitor C_x, V_s and I_s are those for capacitor C_s, p_1 and p_2 are the two non-dc poles of $V_x(s)/I_s(s)$, and $q_s(0) = C_s v_s(0)$ represents the initial charge on capacitor C_s.

Taking the inverse Laplace transform of (51) yields

$$\begin{aligned} q_x(t) &= L^{-1}\{Q_x(s)\} \\ &= h_3(t) q_s(0) \end{aligned} \tag{52}$$

where

$$K = \frac{1}{(C_p + C_s + C_x)(p_1 - p_2)} \tag{53}$$

$$h_3(t) = K C_x \left[(p_1 - p_2) + p_2 e^{p_1^t} + p_1 e^{p_2^t}\right]. \tag{54}$$

Repeating this calculation for all the $h_i(\Delta t)$ functions and substituting the results into (50) lead to the third row of (34).

Now, consider the transfer functions associated with the state variable $\phi_{\mathrm{ctrl}}(t)$. The VCO integrates the voltage on capacitor C_x. Thus

$$r_{4,1}(\Delta t) = \frac{1}{C_x} \int_0^{\Delta t} r_{3,i}(\tau) d\tau. \tag{55}$$

REFERENCES

[1] B. Razavi*, Phase-Locking in High-Performance Systems: From Devices to Architectures*. Hoboken, NJ: Wiley, 2003.

[2] B. Zhang, P. Allen, and J. Huard, "A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 855–865, Jun. 2003.

[3] K. 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.

[4] M. Cassia, P. Shah, and E. Bruun, "Analytical model and behavioral simulation approach for a $\Sigma\Delta$ fractional-N synthesizer employing a sample-hold element," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 11, pp. 850–859, Nov. 2003.

[5] S. E. Meninger and M. H. Perrott, "A 1-MHz bandwidth 3.6-GHz 0.18-μm CMOS fractional-N synthesizer utilizing a hybrid PFD/DAC structure for reduced broadband phase noise," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 966–980, Apr. 2006.

[6] L. Liu and B. Li, "Phase noise cancellation for a $\Sigma\Delta$ fractional-N PLL employing a sample-and-hold element," presented at the Asia-Pac. Microw. Conf. (APMC), Suzhou, China, 2005.

[7] E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700-kHz bandwidth $\Sigma\Delta$ fractional synthesizer with spurs compensation and linearization techniques for WCDMA applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1446–1454, Sep. 2004.

[8] F. M. Gardner, "Charge-pump phase-lock loops," *IEEE Trans. Commun.*, vol. COM-28, no. 11, pp. 1849–1858, Nov. 1980.

[9] I. Galton, "Delta-sigma fractional- N phase-locked loops," in *Phase-Locking in High-Performance Systems: From Devices to Architectures*, B. Razavi, Ed. Hoboken, NJ: Wiley, 2003.

[10] J. P. Hein and J. W. Scott, "z-domain model for discrete-time PLL's," *IEEE Trans. Circuits Syst.*, vol. 35, no. 11, pp. 1393–1400, Nov. 1988.

[11] J. A. Crawford*, Frequency Synthesizer Handbook*. Norwood, MA: Artech House, 1994.

[12] J. Craninckx and M. S. Steyaert, "A fully integrated CMOS DCS-1800 frequency synthesizer," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2054–2065, Dec. 1998.

[13] J. G. Proakis and D. K. Manolakis*, Digital Signal Processing: Principles, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 2006.

Kevin J. Wang (S'96–M'99) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995 and the M.S. degree in electrical engineering from the University of California, Berkeley, in 1998. Since 2003, he has been working toward the Ph.D. degree at the University of California, San Diego.

From 1998 to 2003, he was a Member of the Technical Staff at Silicon Wave, San Diego, where he designed mixed-signal circuits for Bluetooth.

Ian Galton (M'92–SM'09) received the B.Sc. degree in electrical engineering from Brown University, Providence, RI, in 1984 and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1989 and 1992, respectively.

Since 1996, he has been a Professor of electrical engineering with 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 the University of California, Irvine, and prior to 1989, he was with Acuson and Mead Data Central. His research involves the invention, analysis, and integrated circuit implementation of critical communication system blocks such as data converters, frequency synthesizers, and clock recovery systems. In addition to his academic research, he regularly consults at several semiconductor companies and teaches industry-oriented short courses on the design of mixed-signal integrated circuits.

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