Fractional-N Frequency Synthesizer Design at the Transfer Function Level Using a Direct Closed Loop Realization Algorithm

Charlotte Y. Lau
Microsystems Technology Laboratory, MIT
lotty@mit.edu

Michael H. Perrott
Microsystems Technology Laboratory, MIT
http://www-mtl.mit.edu/~perrott

ABSTRACT

A new methodology for designing fractional-N frequency synthesizers and other phase locked loop (PLL) circuits is presented. The approach achieves direct realization of the desired closed loop PLL transfer function given a set of user-specified parameters and automatically calculates the corresponding open loop PLL parameters. The algorithm also accomodates nonidealities such as parasitic poles and zeros. The entire methodology has been implemented in a GUI-based software package, which is used to verify the approach through comparison of the calculated and simulated dynamic and noise performance of a third order Σ-Δ fractional-N frequency synthesizer.

Categories and Subject Descriptors

I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Algorithms

Keywords

fractional-N,frequency,synthesizer,sigma,delta,PLL,design

1. INTRODUCTION

Phase-locked loops (PLLs) are employed in a wide variety of communication circuits including frequency synthesizers, modulators and demodulators, and clock recovery circuits. Of recent interest are fractional-N frequency synthesizers that employ Σ-Δ modulation to achieve fine frequency resolution, fast switching speed, and good noise performance [8]. As shown in Figure 1, the system is similar to a classical integer-N synthesizer in that it is composed of a phase-frequency detector (PFD), charge pump, voltage controlled oscillator (VCO), and a frequency divider.

In contrast to integer-N synthesizers, the fractional-N approach achieves fine frequency resolution by dithering the frequency divider between integer values to achieve fractional divide ratios. As shown in Figure 2, this dithering action creates quantization noise that is added to other noise sources such as charge pump and VCO noise. Inclusion of this extra noise source complicates design efforts compared to that required for more traditional PLL implementations.

DAC 2003, June 2–6, 2003, Anaheim, California, USA.

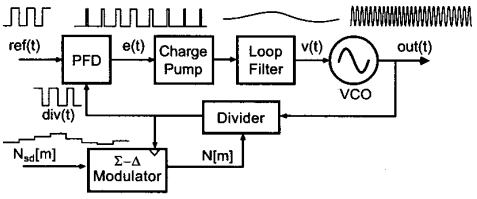

Figure 1: Σ-Δ fractional-N synthesizer and associated signals.

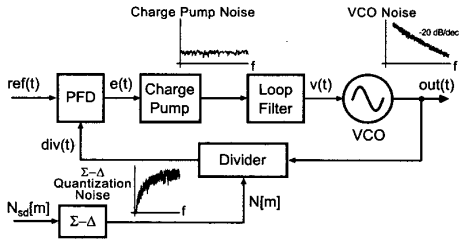

Figure 2: Spectral densities of PLL noise sources.

A general model for fractional-N frequency synthesizers allowing both dynamic and noise analysis of their behavior has recently been proposed in [6]. This model, shown in Figure 3, consists of a frequency domain description of each synthesizer component. As discussed in [6], the closed loop behavior of the synthesizer is completely parameterized by the function $G(f)$, which is related to the open loop transfer function, $A(f)$, of the PLL feedback loop as

$$G(f) = \frac{A(f)}{1 + A(f)}, \text{ where } A(f) = \frac{\alpha I_{cp} H(f) K_v}{N_{nom} 2\pi j f}. \quad (1)$$

As an example, the closed loop transfer function from the reference phase, $\Phi_{ref}[k]$, to the output phase of the synthesizer, $\Phi_{out}(t)$, is equal to $N_{nom} T G(f)$.

Given the above model, the key to designing a fractional-N frequency synthesizer is to appropriately realize its $G(f)$ function according to user specifications, such as desired

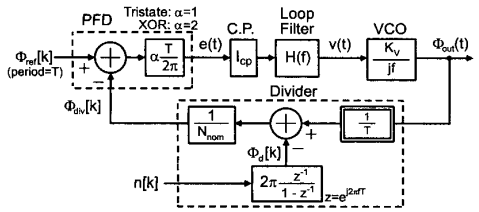

Figure 3: Frequency domain model of Σ-Δ fractional-N synthesizer.

PLL bandwidth, order, etc. However, as shown in Equation 1, the setting of $G(f)$ depends on the open loop transfer function, $A(f)$. This situation is no different than encountered when designing other PLL circuits—the classical methodology [1, 2, 7, 3] is to choose appropriate open loop poles/zeros and then properly set the open loop gain to achieve a desired phase or gain margin specification to insure closed loop stability. However, the classical design procedure assumes second order closed loop dynamics, and therefore proves inadequate for designing closed loop PLL dynamics with third order rolloff. Further, the procedure is clumsy when trying to accommodate parasitic poles that are close in frequency to the desired closed loop bandwidth. Both of these design conditions are encountered in the design of fractional-N frequency synthesizers since they improve suppression of quantization noise produced by dithering.

To overcome the shortcomings of the classical design approach, we propose a new design methodology in which $G(f)$ is *directly* designed according to user specifications rather than *indirectly* inferred from the design of open loop transfer function, $A(f)$. The algorithm determines $G(f)$ in closed form and automatically calculates the corresponding values of open loop parameters. In contrast to the closed loop design method proposed in [4], our new approach addresses multiple topologies and also incorporates the impact of parasitics. The entire procedure is implemented within a GUI-based Matlab tool, hence dramatically simplifying the process of designing fractional-N frequency synthesizers as well as other PLL circuits, and allowing the designer to immediately assess the overall dynamic and noise performance.

An outline of the paper is as follows. Section 2 summarizes the classical open loop design approach and its limitations, motivating the proposed closed loop approach. An overview of the new approach is presented in Section 3. Since phase/gain margins are not used to dictate the design procedure, specification of the closed loop response adopted for the new approach is explained in Section 4. The method for calculating the open loop transfer function is described in Sections 5 and 6, followed by a design example and verification in Section 7.

2. BACKGROUND

The classical approach for designing a PLL circuit is to first choose a topology for its open loop transfer function, $A(f)$, and then appropriately set the gain and pole/zero locations of $A(f)$ to achieve a desired phase/gain margin. The phase margin method is illustrated in Figure 4 with an example system. Here the topology of $A(f)$ is chosen to be an integrator cascaded with three poles at f_{p1}, f_{p2} and f_{p3}. Its Bode diagram, shown on the left hand side of the figure, is examined to assess closed loop stability based on the open loop phase at unity gain crossover. The right portion of the figure reveals that the closed loop poles shift into the right-half S-plane if the phase margin is less than 0 degrees. Figure 5 verifies that the corresponding closed loop response becomes unstable under this condition, as evidenced by high levels of peaking in its frequency response and growing ringing in its step response. In practice, the open loop gain and/or open loop pole/zero locations would be iteratively adjusted to achieve adequate phase margin.

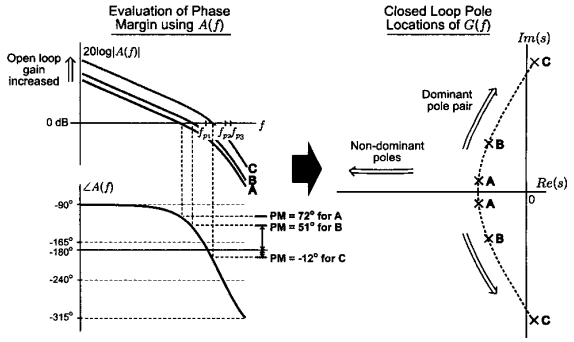

Figure 4: Bode plots of $A(f)$ and corresponding poles of $G(f)$.

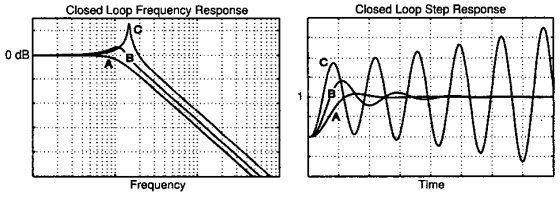

Figure 5: Frequency and step responses of $G(f)$ corresponding to Figure 4.

The classical approach has become the design methodology of choice for PLL circuits due to the relative simplicity of examining and adjusting the open loop PLL response, $A(f)$, rather than its closed loop counterpart, $G(f)$. However, the approach is heuristic in nature and based on the assumption that the closed loop system can be well approximated as an ideal second order system. As mentioned in the introduction, this assumption may not hold true when designing fractional-N frequency synthesizers. Further, the approach, focused primarily on system stability, only provides coarse control in the actual realization of $G(f)$. Finally, the requirement for manual iteration causes the procedure to be somewhat tedious for experienced PLL designers and daunting for novice designers.

3. PROPOSED APPROACH

The proposed closed loop approach achieves the design of the closed loop response, $G(f)$, directly from user specifications. The difference between this approach and the classical open loop design approach is illustrated in Figure 6.

As shown by the dotted path, the classical approach focuses on designing the open loop response, $A(f)$, according to the phase or gain margin criterion. The closed loop response, $G(f)$, is thereby indirectly designed to be stable. However, the new approach takes a direct design path from the performance specification to $G(f)$, and then determines the required $A(f)$ to achieve that $G(f)$.

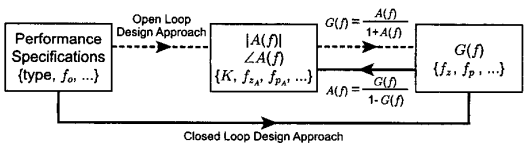

Figure 6: Comparison between the classical open loop approach and the new closed loop approach.

The key to realizing the proposed approach is to recognize that the pole/zero locations of $G(f)$ in the S-plane ultimately govern the desired closed loop behavior. Therefore, a design procedure that achieves accurate placement of the closed loop poles and zeros will yield a precisely set closed loop response. Stability robustness is inherently achieved in this approach by properly positioning the poles in the left-half S-plane.

Two issues must be addressed to achieve the proposed design approach. First, since the design procedure is no longer based on phase/gain margin criterion applied to $A(f)$, a method of mapping user specifications to the desired closed loop function, $G(f)$, must be developed. This method is discussed in Section 4. Second, after $G(f)$ is specified, the open loop response, $A(f)$, must be appropriately chosen to implement the desired $G(f)$. A complicating factor is that, to be practical, the procedure must accommodate the fact that $A(f)$ may contain parasitic poles/zeros that are somewhat out of the control of the designer. Each of these issues are addressed in Sections 5 and 6.

4. SPECIFICATION OF CLOSED LOOP RESPONSE

We will assume that $G(f)$ is specified by four parameters: bandwidth, order, shape and type. Qualitative definitions for these parameters are provided in this section, followed by quantitative descriptions of their impact on $G(f)$. As we proceed, we will alter the choice of independent frequency variable between f, w, and s according to which variable offers the simplest parameterization in the given context. In all cases, it will be assumed that each of these variables are related as

$$s = jw = j2\pi f.$$

4.1 Qualitative Description

The first two parameters are illustrated in Figure 7. Bandwidth, which is denoted as f_o, is defined in an asymptotic manner as illustrated in the figure. Specifically, it is measured by extrapolating the slope of the rolloff that occurs after the dominant poles but before the high frequency parasitic poles and zeros. Order, which is denoted as n, is also defined by the rolloff characteristic as shown — n equals the number of dominant closed loop poles in $G(f)$. This definition is preferred over specifying n as the number of independent state variables in the system since the rolloff characteristic is of key importance for achieving good PLL noise performance.

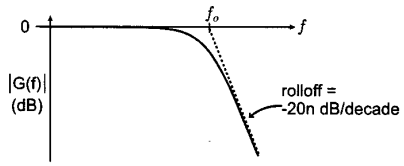

Figure 7: Definition of order and bandwidth for $G(f)$.

Shape describes the particular form of the closed loop transfer function and step response for a given specification of order. Typical choices of shape specification include Butterworth, Bessel, Chebyshev and elliptic. In practice, we will only consider the dominant closed loop poles when striving for a given shape specification since non-dominant poles have only a second order effect on the closed loop response.

Type is defined as the number of integrators in the open loop transfer function. Practical PLL implementations are usually type I or II. Since the VCO includes an integrator, the additional integrator in a type II PLL is realized in the loop filter. As two integrators contribute a -180^o phase shift, a type II loop filter must include a zero, f_z, to allow stable closed loop implementation of $G(f)$. The corresponding PLL loop filter is described as a lead/lag filter. The type II implementation introduces an additional factor that must be included in the closed loop design specification, namely the ratio f_z/f_o. The impact of this ratio setting will be explained shortly.

4.2 Quantitative Description

Table 1 displays $G(s)$ parameterizations as a function of type and order. The type and order values are constrained to the values shown since few PLL designs will ever seek higher values due to the prohibitive analog complexity that would be required in such cases. In any case, the table reveals that $G(s)$ is parameterized by a set of variables that includes w_{co}, w_{c1}, and Q. As shown in Table 2, the value of these variables are set by the values chosen for bandwidth, order, and filter shape. Table 2 displays the relationship between these sets of parameters when the shape is chosen to be either Butterworth or Bessel. If a Chebyshev or elliptic filter shape is desired, standard Matlab routines are applied to solve for the appropriate parameter values.

Table 1 shows that an extra pole and zero are present for type II systems, namely w_{cp} and w_z. This pole/zero pair occurs due to the fact that the open loop transfer function has two integrators and a stabilizing zero, as readily seen through application of the root locus technique [3]. As shown in Table 1, the ratio w_{cp}/w_z is completely determined by the values chosen for shape, bandwidth, and the ratio w_z/w_o. Since w_z/w_o can be freely chosen independent of the shape and bandwidth parameters, we will assume that it is the key parameter for setting a desired w_{cp}/w_z ratio. Practical designs set the value of w_z/w_o to be in the range of 1/10 to 1/3.

Table 1: Allowed $G(s)$ parameterizations for system with no parasitics

Type	Order	$G(s)$	w_{cp}/w_z
I	1	$\frac{1}{1+\frac{s}{w_{co}}}$	N/A
	2	$\frac{1}{1+\frac{s}{w_{co}Q}+\left(\frac{s}{w_{co}}\right)^2}$	N/A
	3	$\frac{1}{\left(1+\frac{s}{w_{c1}}\right)\left(1+\frac{s}{w_{co}Q}+\left(\frac{s}{w_{co}}\right)^2\right)}$	N/A
II	1	$\frac{\left(1+\frac{s}{w_z}\right)}{\left(1+\frac{s}{w_{cp}}\right)}\frac{1}{\left(1+\frac{s}{w_{co}}\right)}$	$\frac{1}{1-\frac{w_z}{w_{co}}}$
	2	$\frac{\left(1+\frac{s}{w_z}\right)}{\left(1+\frac{s}{w_{cp}}\right)}\frac{1}{\left(1+\frac{s}{w_{co}Q}+\left(\frac{s}{w_{co}}\right)^2\right)}$	$\frac{1}{1-\frac{w_z}{w_{co}Q}}$
	3	$\frac{\left(1+\frac{s}{w_z}\right)}{\left(1+\frac{s}{w_{cp}}\right)}\frac{1}{\left(1+\frac{s}{w_{c1}}\right)\left(1+\frac{s}{w_{co}Q}+\left(\frac{s}{w_{co}}\right)^2\right)}$	$\frac{1}{1-\frac{w_z}{w_{c1}}-\frac{w_z}{w_{co}Q}}$

Table 2: Filter parameters for Butterworth and Bessel responses

	Butterworth			Bessel		
Order	w_{c1}	w_{co}	Q	w_{c1}	w_{co}	Q
2	N/A	w_o	0.707	N/A	w_o	0.577
3	w_o	w_o	1	$(0.9416)w_o$	$(1.0305)w_o$	0.691

5. SPECIFICATION OF IDEAL OPEN LOOP RESPONSE

Given that $G(s)$ is chosen according to Tables 1 and 2 for a given type, order, bandwidth, and shape specification, the next step is to determine its corresponding $A(s)$ according to Equation 1. Under the assumption that there is no pole/zero cancellation, Equation 1 represents a one-to-one mapping between $G(s)$ and $A(s)$, implying that there is a unique $A(s)$ for a given choice of $G(s)$. The key equation is derived from Equation 1 as

$$A(s) = \frac{G(s)}{1 - G(s)}. \tag{2}$$

For the constrained set of order and type values for $G(s)$ considered in Table 1, the corresponding $A(s)$ functions satisfying Equation 2 can be derived as closed form analytical expressions using algebra. Table 3 displays these analytical expressions for $A(s)$, and shows the relationship between the open loop parameters, K, w_p and Q, and the corresponding closed loop parameters displayed in Table 1. As a result, design of the open loop dynamics amounts to a simple table lookup procedure.

6. INCORPORATION OF PARASITIC POLES AND ZEROS

The lookup table design procedure encapsulated by Tables 1, 2, and 3 assumes that the open loop PLL transfer function, $A(s)$, can be realized free of parasitic poles and zeros. However, in practice, this assumption is unrealistic since various PLL components, such as the VCO and loop filter, invariably contain parasitic poles and/or zeros whose values are poorly controlled by the designer. As shown in Figure 8, these open loop parasitics will shift the dominant closed loop pole locations away from the values targeted by the lookup table design procedure. The manner in which these open loop parasitics shift the dominant closed loop poles is nonlinear, as seen by inspection of Equation 2.

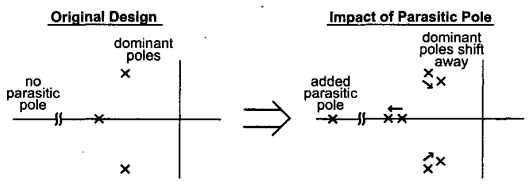

Figure 8: Impact of an open loop parasitic pole on the dominant closed loop pole locations.

At first glance, the existence of parasitics may seem to invalidate the lookup table design procedure. However, under the assumption that such parasitics occur at frequencies reasonably higher than the closed loop bandwidth of the PLL, their inclusion in the open loop transfer function will not *dramatically* shift the position of the dominant closed loop poles. Therefore, rather than abandoning the lookup table approach, we can leverage it to obtain a good starting point for a nonlinear optimzation algorithm that seeks the proper parameter values of $A(s)$ to achieve the desired closed loop dominant pole locations specified in Table 2.

Figure 9 illustrates the fundamental idea behind the nonlinear optimization algorithm that we now propose to achieve the desired closed loop dominant pole placement. First, the ideal open loop transfer function, $A(s)$, is calculated from the table lookup procedure so that it achieves the desired closed loop pole locations. Estimated values of parasitic poles and zeros, which are obtained from SPICE simulations or measurements of PLL components, are then added to $A(s)$ and the resulting dominant pole locations are calculated using a root solver in Matlab—this step is illustrated in the top left portion of Figure 9. Each open loop parameter specified in Table 3 is then perturbed by a small amount and the resulting change in the dominant closed loop pole locations is observed, as illustrated in the remaining portion of the figure. These observed perturbations are then used to construct a matrix relationship which is used to calculate the required shift in open loop parameter values for returning the dominant closed loop pole to their desired locations.

The above matrix relationship is formulated as

$$\Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{h},$$

where $\Delta\mathbf{h}$ is a vector corresponding to a change in the open loop parameters, $\Delta\mathbf{y}$ is a vector corresponding to the resulting change in the dominant closed loop pole locations, and $\mathbf{C}$ is a matrix representing a linearized relationship between the two vectors. For the third order example illustrated in Figure 9, with closed loop dominant poles at P_{real}, $P_{complex}$ and $P^*_{complex}$, we have

$$\Delta\mathbf{y} = \begin{bmatrix} Re\{\Delta P_{complex}\} \\ Im\{\Delta P_{complex}\} \\ \Delta P_{real} \end{bmatrix}, \quad \Delta\mathbf{h} = \begin{bmatrix} \Delta K \\ \Delta f_p \\ \Delta Q_p \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} Re\left\{\frac{\Delta P_{complex}}{\Delta K}\right\} & Re\left\{\frac{\Delta P_{complex}}{\Delta f_p}\right\} & Re\left\{\frac{\Delta P_{complex}}{\Delta Q_p}\right\} \\ Im\left\{\frac{\Delta P_{complex}}{\Delta K}\right\} & Im\left\{\frac{\Delta P_{complex}}{\Delta f_p}\right\} & Im\left\{\frac{\Delta P_{complex}}{\Delta Q_p}\right\} \\ \frac{\Delta P_{real}}{\Delta K} & \frac{\Delta P_{real}}{\Delta f_p} & \frac{\Delta P_{real}}{\Delta Q_p} \end{bmatrix},$$

Table 3: Formulae for calculating $A(s)$ parameters

Type	Order	$A(s)$	K	w_p	Q_p
I	1	$\frac{K}{s}$	w_{co}	N/A	N/A
	2	$\frac{K}{s\left(1+\frac{s}{w_p}\right)}$	$w_{co}Q$	$\frac{w_{co}}{Q}$	N/A
	3	$\frac{K}{s\left(1+\frac{s}{w_pQ_p}+\left(\frac{s}{w_p}\right)^2\right)}$	$\frac{w_{co}Q}{Q\left(\frac{w_{co}}{w_{c1}}\right)+1}$	$w_{co}\sqrt{\frac{w_{c1}}{K}}$	$\frac{w_pQ}{w_{co}+Qw_{c1}}$
II	1	$\frac{K\left(1+\frac{s}{w_z}\right)}{s^2}$	$w_{co}w_{cp}$	N/A	N/A
	2	$\frac{K\left(1+\frac{s}{w_z}\right)}{s^2\left(1+\frac{s}{w_p}\right)}$	$\frac{w_{co}Q}{\frac{Q}{w_{co}}+\frac{1}{w_{cp}}}$	$w_{co}\left(\frac{1}{Q}+\frac{w_{cp}}{w_{co}}\right)$	N/A
	3	$\frac{K\left(1+\frac{s}{w_z}\right)}{s^2\left(1+\frac{s}{w_pQ_p}+\left(\frac{s}{w_p}\right)^2\right)}$	$\frac{w_{co}Q}{\text{DEN}^*}$	$w_{co}\sqrt{\frac{w_{c1}w_{cp}}{K}}$	$\frac{w_pQ}{w_{co}+Q\left(w_{cp}+w_{c1}\right)}$
		$\text{DEN}^* = Q\left(\frac{w_{co}}{w_{c1}w_{cp}}+\frac{1}{w_{co}}\right)+\frac{1}{w_{c1}}+\frac{1}{w_{cp}}$			

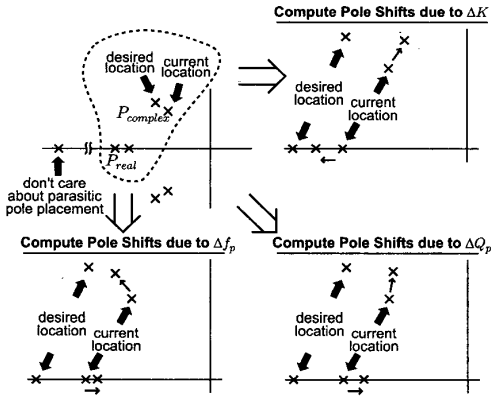

Figure 9: Construction of C matrix for calculating open loop parameter adjustment.

where

$$\begin{aligned}\Delta P_{complex} &= P_{complex\,desired} - P_{complex\,current},\\ \Delta P_{real} &= P_{real\,desired} - P_{real\,current}.\end{aligned}$$

Given the above relationships, we calculate the required change in the open loop parameters to return the dominant closed loop poles back to their desired locations as

$$\Delta \mathbf{h} = \mathbf{C}^{-1}\Delta \mathbf{y}.$$

Since the above matrix formulation is based on a linearization of the nonlinear relationship between the open and closed loop parameters, the process described above is conducted iteratively until the dominant closed loop poles are within a chosen distance from their desired locations. Application of the algorithm has revealed that less than 20 iterations are typically required to return the closed loop poles to within 0.1% of their desired locations.

An observant reader may suggest that this iterative approach has two potential limitations. First, if the parasitics are too close to the dominant poles, they can seriously alter the dominant poles' root locus, making it impossible to return the dominant poles to the desired location by tweaking the open loop parameters. Nonetheless, our observation confirms that as long as the parasitics are a factor of two to four above the closed loop bandwidth, depending on the choices of type and order, the iterative algorithm always converges.

The second potential limitation is that the closed loop parasitic poles and zeros also move during the iterations, but their translations are ignored since the above procedure only considers the dominant poles. However, other than their impact of shifting the dominant poles, these parasitics only have second order influence on the closed loop dynamics. Further, regardless of the tweaking of open loop parameters, the closed loop locations of the parasitic poles/zeros end up being quite close to their open loop counterparts. This fact is seen by observing that

$$G(s) = \frac{A(s)}{1+A(s)} \approx A(s)$$

given $|A(s)| \ll 1$, which holds true at frequencies above the closed loop bandwidth where the open loop parasitics occur.

In summary, the proposed design methodology accurately places the closed loop dominant poles according to user specifications and accommodates the presence of open loop parasitic poles and zeros. The direct incorporation of the parasitics stands in contrast to the classical open loop approach, which simply seeks to insure that parasitics are at least an order of magnitude higher in frequency than the closed loop bandwidth in order to minimize phase margin degradation.

7. RESULTS AND VERIFICATION

We now demonstrate the proposed closed loop design methodology applied to a Σ-Δ fractional-N frequency synthesizer employing a third order Σ-Δ modulator. The example will include design and verification of the closed loop PLL dynamics as well as an estimation of its noise performance. The complete design operation was performed using a GUI-based Matlab tool that implements the proposed design approach. A PLL simulator, described in [5], was used to confirm the results. Both of these tools are available at http://www-mtl.mit.edu/~perrott.

Since the synthesizer uses a third order Σ-Δ modulator, the closed loop transfer function, $G(f)$, is chosen to be third order. Other specifications of $G(f)$ include choosing a type II implementation, with $f_z/f_o = 1/8$, a Butterworth re-

Table 4: Calculated Open loop parameters

Open loop parameters	Without parasitics	With parasitics
K	2.538e+11	2.294e+11
f_p	4.583e+5	4.841e+5
f_z	3.75e+4	3.75e+4
Q_p	0.705	0.7931

sponse, and a bandwidth of 300 kHz. Further, a parasitic pole is purposefully included at 1.2MHz to improve suppression of the quantization noise caused by dithering the divide value.

Using Equation 1 and Table 3, the open loop transfer function is chosen to be

$$A(s) = \frac{\alpha I_{cp} H(s) K_v}{N_{nom} s} = \frac{K\left(1+\frac{s}{w_z}\right)}{s^2\left(1+\frac{s}{w_p Q_p}+\left(\frac{s}{w_p}\right)^2\right)}.$$

Therefore, the corresponding loop filter transfer function is

$$H(s) = \frac{K_{LP}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_p Q_p}+\left(\frac{s}{w_p}\right)^2\right)}, \text{ where } K_{LP} = K\frac{N_{nom}}{\alpha I_{cp} K_v}.$$

The open loop parameter values in the above expression were calculated according to Tables 2 and 3, and are displayed in Table 4 under the "Without parasitics" column. The parasitic pole was then included and the proposed iterative routine adjusted the open loop values to return the dominant poles back to within 0.1% of their original locations. The resulting open loop parameter values are displayed in the same table under the "With Parasitics" column. Note that tedious manual iterations, as would be encountered when seeking adequate phase margin with the classical open loop design procedure, were completely avoided in this procedure. The new algorithm simplifies the design procedure and computes all parameters in Matlab.

The PLL is then simulated with the same configuration and open loop parameters. Figure 10 displays the calculated closed loop step response from the design procedure versus the simulated step response. The close correspondence of the two plots verifies the accuracy of the design calculations.

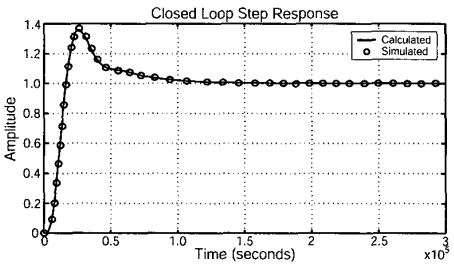

Figure 10: Plot of closed loop step response.

Given the resulting $G(f)$ function from the above design procedure, the phase noise and jitter performance of the PLL were then evaluated using the noise model described in [6] and by simulation as described in [5]. For simplicity, the magnitude of the detector and VCO noise were assumed to be the same as in [6]. Figure 11 displays the resulting calculated phase noise, which is broken up into contributions of the individual PLL noise sources, and the overall phase noise calculated by the simulator. The close correspondence of the overall calculated and simulated phase noise verifies the accuracy of the design approach in estimating the synthesizer noise performance.

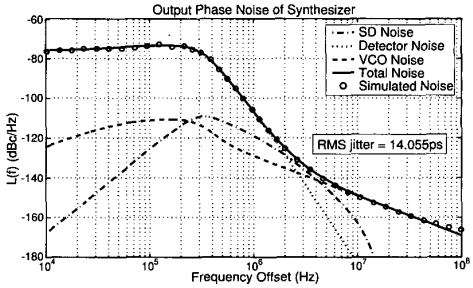

Figure 11: Calculated phase noise.

8. CONCLUSIONS

A closed loop design methodology is presented and verified that allows direct design of fractional-N frequency synthesizers and other PLL circuits at the transfer function level based on closed loop parameter specifications. The technique uses a lookup table based approach to calculate the open loop parameter values in the absence of parasitics and an iterative algorithm to adjust those parameter values to accommodate parasitic poles and zeros. The entire methodology has been incorporated in a GUI-based Matlab tool and provides designers with a fast and simple methodology for realizing high performance PLL circuits.

9. REFERENCES

[1] F. M. Gardner. *Phaselock Techniques.* Wiley, New York, second edition, 1979.

[2] D. Johns and K. Martin. *Analog Integrated Circuit Design.* Wiley, 1997.

[3] T. Lee. *The Design of CMOS Radio-Frequency Integrated Circuits.* Cambridge University Press, 1998.

[4] S. Mirabbasi and K. Martin. Design of Loop Filter in Phase-locked Loops. *Electronics Letters*, 35(21):1801–1802, Oct. 1999.

[5] M. Perrott. Fast and Accurate Behavioral Simulation of Fractional-N Frequency Synthesizers and other PLL/DLL Circuits. In *Proceedings of Design Automation Conference*, pages 498–503, June 2002.

[6] M. Perrott, M. Trott, and C. Sodini. A Modeling Approach for Σ-Δ Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis. *IEEE Journal of Solid State Circuits*, 37(8):1028–1038, Aug. 2002.

[7] B. Razavi. *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design.* IEEE Press, New York, 1996.

[8] T. A. Riley, M. A. Copeland, and T. A. Kwasniewski. Delta-Sigma Modulation in Fractional-N Frequency Synthesis. *JSSC*, 28(5):553–559, May 1993.