2009-07-09

Design and Analysis of a Dual-Mode Cascaded-Loop Frequency Synthesizer

Xiongliang Lai
Brigham Young University - Provo

DESIGN AND ANALYSIS OF A DUAL-MODE CASCADED-LOOP FREQUENCY SYNTHESIZER

by

Xiongliang Lai

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Brigham Young University

August 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Xiongliang Lai

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Donald T. Comer, Chair
Date	David J. Comer
Date	David A. Penry

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Xiongliang Lai in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Donald T. Comer
Chair, Graduate Committee

Accepted for the Department

Michael J. Wirthlin
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology

ABSTRACT

DESIGN AND ANALYSIS OF A DUAL-MODE CASCADED-LOOP FREQUENCY SYNTHESIZER

Xiongliang Lai

Department of Electrical and Computer Engineering

Master of Science

A new architecture for a frequency synthesizer with adjustable output frequency range and channel spacing is introduced. It is intended for the generation of closely spaced frequency channels in the GHz range while producing minimal spurious phase noise components. The architecture employs two independent phase-locked loops that are driven in cascade by a single reference oscillator. The approach provides fine resolution and wide bandwidth as well as low phase noise and should find application in many contemporary communication systems.

The synthesizer can be operated in either of two different modes: *nonfractional* and *mini-denominator fractional* modes. The architecture produces no fractional spurs in the first mode and relatively small phase spurs when operated in the second mode. For example, in an application to a P-GSM 900 system, it is capable of tuning from 890 – 915 MHz with a channel spacing of 200 kHz and shows worst case phase spurs of -100 dBc at

an offset frequency of 833 kHz. Because of the low magnitude and location of the worst case spurs, the phase-locked loop filters can be designed with a wide bandwidth which in turn results in a fast settling time. A linear frequency-switching settling time (to 0.01% of frequency increments) of 128 μs is typical in the P-GSM 900 application.

ACKNOWLEDGMENTS

I want to first express my deepest and sincere gratitude to my research advisor, Dr. Donald T. Comer for his guidance throughout this work. His incisive understanding of circuit designs and enriched experience in phase-locked loop frequency synthesizers have sparked the initial ideas of this thesis. His patience, encouragement and the many valuable discussions and advices have enabled me to overcome the hardships and frustrations when expanding and refining the research topic. He has not only been my generous mentor, but also a life good friend.

I also want to express my thankfulness to Dr. David J. Comer for his early help and guidance of my transition from a mathematician to a circuit engineer. I appreciate the opportunities of being teaching assistants in his circuit classes, which allows me to learn circuit fundamentals and build a solid basis for this research work.

My thanks also go to Dr. David A. Penry for his time to review this thesis.

My last thankfulness should be my parents. I can still remember my father collected used resistors, capacitors and transistors to build my first radio when I was still a child, which has inspired my earnest interest in electronic world. They have always been my primary source of inspiration, wisdom and courage.

To My Parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Abbreviations and Conventions

The abbreviations and labeling conventions contained in Table 1.1 and Table 1.2 are used commonly throughout this thesis. Most are also defined in the text when used for the first time.

Table 1.1: List of conventions

Labels	
f_{ref}	Synthesizer input reference frequency
f_{out}	Synthesizer output frequency
f_v	Divided frequency from feedback frequency divider
f_0	Carrier frequency
N_1^*	First-loop fractional divider modulus
N_2	Second-loop integer divider modulus
M	Bridging divider modulus
A	First-loop accumulator addend
Q	First-loop accumulator modulus
$C_{1,2,a,b}$	Loop filter capacitors
$R_{1a,b}$	Loop filter resistors
K_φ	Phase-frequency detector gain
K_v	VCO gain
$H_{1,2}(s)$	First and second loop open-loop transfer functions
ζ	Damping ratio
ω_n	Undamped natural frequency
τ	Time constant
ω_p	Pole frequency
ω_z	Zero frequency
$\Delta\varphi$	Discontinuous phase error in phase-frequency detector
Δf	Offset frequency from the carrier
a_k	Fourier transform coefficients
$e_m[n]$	Zero-mean quantization noise
$I_{\text{det error}}$	Charge-pump current error due to phase error
$S_{\text{det error}}(f)$	Baseband power spectral density of $I_{\text{det error}}$
$S_{\text{spur}}(\Delta f)$	Discrete spur power spectral density at offset frequency Δf
P_{carrier}	Carrier Power
$P_{\text{spur}}(\Delta f)$	Discrete spur power at offset frequency Δf

Table 1.1 – Continued

$L(\Delta f)$	Logarithm of ratio of spectral power at offset frequency Δf to the carrier power
$W_{\varphi}(\Delta f)$	Continuous phase-noise power spectral density at offset frequency Δf
$\phi(t)$	Phase deviation from steady phase $2\pi f_0 t$
$v_o(t)$	PLL synthesized passband signal
$P_{RF}(f)$	Average power of passband signal $v_o(t)$ within resolution bandwidth
$W_{RF}(f)$	Approximated power spectral density of passband signal $v_o(t)$ within resolution bandwidth
$P_{\phi}(f_m)$	Average power of baseband signal $\phi(t)$ within resolution bandwidth at centered frequency f_m.
$W_{\phi}(f)$	Power spectral density of baseband signal $\phi(t)$
$\phi_{\text{period}}(t)$	Periodic baseband phase noise signal of $\phi(t)$
$S_{\phi\ \text{period}}(\Delta f)$	Power spectral density of $\phi_{\text{period}}(t)$
$\phi_{\text{wss}}(t)$	Wide-sense stationary baseband phase noise signal of $\phi(t)$
$S_{\phi\ \text{wss}}(t)$	Power spectral density of $\phi_{\text{wss}}(t)$
$\phi_{\text{aperiodic}}(t)$	Aperiodic baseband phase noise signal of $\phi(t)$
$S_{\phi\text{-}\ \text{aperiodic}}(t)$	Power spectral density of $\phi_{\text{aperiodic}}(t)$
$\phi_{\text{nonstat}}(t)$	Nonstationary baseband phase noise signal of $\phi(t)$
$S_{\phi\ \text{nonstat}}(t)$	Power spectral density of $\phi_{\text{nonstat}}(t)$
ζ	Damping ratio
ω_n	Undamped natural frequency
τ	Time constant

Table 1.2: List of abbreviations

Abbreviations	
PLL	Phase-locked loop
IC	Integrated circuit
PFD	Phase frequency detector
LPF	Lowpass filter
VCO	Voltage controlled oscillator
CP	Charge pump
WSS	Wide-sense stationary
RBW	Resolution bandwidth
VBW	Video Bandwidth

2 Introduction

Phase-Locked Loops (PLL) were first invented in 1930's and soon found widespread applications in electronics. After nearly 70 years, phase locking continues to find new applications in electronics, communications and instrumentations [1]. While the basic idea of phase comparison and self-adjusted phase locking has not been changed since its invention, its implementation has evolved into different technologies. These technologies benefit from the rapid development of integrated circuits (ICs) and digital signal processing techniques since 1950's. Integrated circuits lead to the development of fully integrated monolithic PLLs and digital signal processing results in the latest developments of fractional-N frequency synthesizers and digital sampled PLLs.

2.1 A Brief History of Phase-Locked Loop Frequency Synthesizers

In 1930's, superheterodyne receiver architecture was dominant in radio receivers. But superheterodyne receivers require heavy number of tuned stages, a simpler method was desired. In 1932, a new type of receiver architecture, called homodyne and later renamed to synchrodyne, was developed by a team of British scientists. It consisted of a local oscillator, a mixer and an audio amplifier. When the input modulated signal and the local oscillator were mixed at the same phase and frequency, the output was an exact modulating audio representation of the modulated carrier. The initial experiments were

encouraging, but the synchronous reception after a period of time became difficult due to the slight drift in frequency of the local oscillator (nowadays known as phase jitter). To counteract the frequency drift, the frequency and phase of the local oscillator was compared with the frequency and phase of the input modulated signal by a phase detector and their phase difference converted into a correction voltage was fed back into the local oscillator to maintain the local oscillator frequency in the same pace as the input modulated signal. This innovation of receiver architecture starts a new chapter of phase locking in today's electronics.

An interesting phenomenon was observed during the first development of PLLs. If the output frequency from the VCO was divided by a factor and then fed back to the phase detector, the corrected lowpass filter output voltage would continue to drive the VCO output frequency the same factor times the input reference frequency. This phenomenon is not hard to understand if we consider the PLL as a phase maintenance device to synchronize the two input signals to its phase detector at an exactly same frequency. And if you trace this synchronized signal from the phase detector back to the VCO in its feedback path, the frequency multiplication effect will be evident. This frequency multiplication effect by PLLs was soon developed into its own field of frequency synthesis and found its extensive application in today's memories, microprocessors, hard disk drive electronics, RF and wireless transceivers, and optical fiber receivers. We can say that, without the invention of frequency synthesizers, nowadays wireless electronics would not even exist.

2.2 Basic Structure of a PLL

There have been a variety of PLLs in different technologies for different applications. Despite the dazzling variations of PLLs, the basic structure of a PLL has not been changed. Figure 2.1 shows the basic structure of a PLL which includes three essential parts: (1) a phase-frequency detector (PFD), (2) a lowpass filter (LPF) and (3) a voltage controlled oscillator (VCO). The phase-frequency detector compares the phase difference between the input reference signal and the VCO output signal and converts this phase difference into a current or voltage output for the lowpass filter. The lowpass filter smoothes this fast fluctuating phase difference and provides an average control signal for the voltage controlled oscillator. This control signal subsequently changes the VCO output frequency in a direction that reduces the phase difference between the input reference signal and the VCO output signal.

When the loop is locked, the control signal from the LPF sets the average frequency of the VCO exactly equal to the average frequency of the input reference signal. And for each cycle of the input reference signal, there is one and only one cycle of the VCO output.

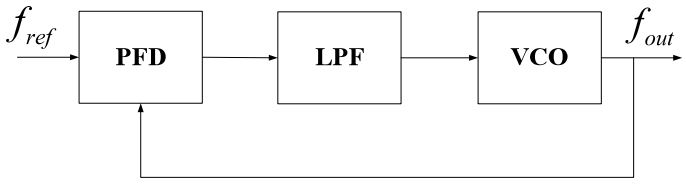


Figure 2.1: Basic structure of a phase-locked loop

2.3 Challenges in Today's PLL Frequency Synthesizer Design

As key components of almost all communication systems and most computing electronics, PLL frequency synthesizers have been imposed on stricter and stricter technical requirements. Examples of requirements for frequency synthesizers for two common wireless communication applications are listed in Table 2.1for their respective frequency range, channel spacing (or frequency resolution) and frequency hopping settling speed.

Table 2.1: Wireless system design requirements for frequency synthesizers

System	Frequency Range	Channel Spacing (Frequency Resolution)	Frequency-Switch Settling Time
P-GSM 900 (Uplink)	890 – 915 MHz	200 kHz	344.3 μs
Bluetooth	2.402 – 2.480 GHz	1 MHz	224 μs
802.11b	2.400 – 2.484 GHz	5 MHz	5 μs
UMTS (Rx)	2.110 – 2.170 GHz	5 MHz	200 μs

2.3.1 PLL Specifications

Essentially, a PLL is a phase feedback system. Although none of PLLs is linear, when the phase variations in a PLL only encounter small changes, each loop components can still be treated as linear models and linear feedback theories can efficiently applied to the analysis of PLLs. Sufficiently, the results obtained from the linear analysis of PLLs can be used to predict the PLL performance in its nonlinear instances. In the following, we are going to list some of the most crucial parameters when designing a PLL frequency synthesizer:

1. *Bandwidth*: Bandwidth is the most fundamental property of a PLL and gives the basic tone of a PLL's overall performance. Even though literatures often mix

them together, bandwidth should be clearly identified from its two distinct identities: open-loop unity-gain bandwidth which is often used to determine the stability of the PLL and closed-loop bandwidth which can be used to estimate the scope of other parameters: linear track-in range, linear track-in settling time, nonlinear pull-in range, nonlinear pull-in settling time and the reduction of phase noise and spurs caused by each of the components in the PLL.

2. *Stability*: PLLs always suffer phase variations of the input reference source and phase interruptions from internal loop components. Stability defines whether the PLL output phase variation converges or diverges during these input phase variations or internal phase interruptions. As measuring stability of a linear feedback system, the stability of a PLL can be conveniently measured by the phase margin at the unity-gain frequency of its open-loop frequency response. A phase margin of 60 degree or more is usually required for a practically stable PLL design to account for temperature and manufacturing process variations.

3. *Tracking*: When the PLL's initial status is locked, which is the output phase has been synchronized with the input phase, any small phase variations in the input will be followed exactly by the same change at the output and this process is called tracking. Tracking is studied through linear approximation of the dynamics of the PLL system when phase errors in the PFD are small so that the VCO will not slip cycles. Because PLL analysis during tracking process has been linearized, the PLL output response can be simulated by linear s-domain transfer functions. Two common specifications are often used to describe a PLL's tracking behavior:

- *PFD Linear Tracking Range*: Frequency variations in the input of a PLL will cause phase errors in its PFD. A type of PFD can only handle a limited range of phase errors in order for it to work linearly without causing the VCO to slip cycles. The PFD linear tracking range is the maximum range of input frequencies so that the phase errors are in the PFD's linear range.
- *Linear Tracking Settling Time*: During a PLL's tracking process, the output frequency follows the input frequency variation and settles gradually to the target frequency. Linear tracking settling time measures how soon the output frequency falls within a percentage error of its target frequency. A common way to measure the linear tracking settling time is to input a unit frequency step and measure the time the PLL takes to settle at the target frequency. Figure 2.2 demonstrates an example of a PLL's tracking process in its locked status for an input frequency step from 1879.85 MHz to 1880.00 MHz. The tracking process takes around 200 us to settle its output frequency within a tolerance band of 20 kHz from the target frequency.

4. *Acquisition*: Acquisition is the process that the PLL is bringing itself back to the locked status from an out-of-locked status. Comparing with the tracking process that is assumed the loop has already been locked and the PLL's behavior can be well approximated by a linear system for small phase errors in its PFD, acquisition is inherently a nonlinear process and nonlinear analysis is generally needed, because out-of-locked phase errors in the PFD will greatly exceed the

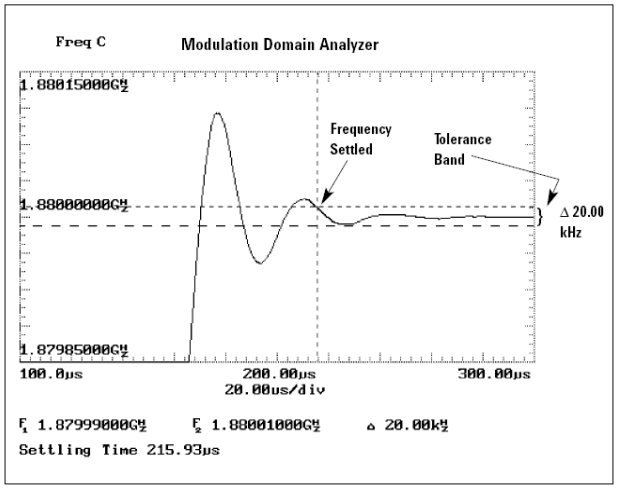


Figure 2.2: Illustration of a PLL's linear tracking process

linear range that the PFD can handle. If the loop acquires lock by itself, the process is called self-acquisition and if it is assisted by auxiliary circuits, the process is called aided acquisition. According to the input signal conditions, acquisition can also be categorized as phase acquisition and frequency acquisition. If an acquisition process is a self-aided frequency acquisition process, this process is also called a pull-in process. Figure 2.3 illustrates (A) linear tracking process and (B) nonlinear pull-in process for an input frequency step from 1649.7 MHz to 1686.8 MHz for an integrated fractional-N frequency synthesizer of ANALOG DEVICES ADF4154. The whipsaws in the pull-in process are caused by PFD phase cycle slips.

5. *Frequency Tuning Resolution*: Frequency tuning resolution is one of the unique properties of PLL frequency synthesizers and is defined as the minimum output frequency step that a frequency synthesizer can generate. Frequency tuning resolution is also known as channel spacing for communication systems because

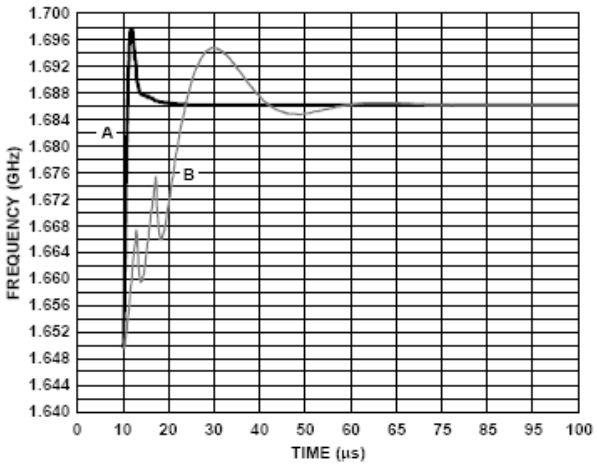


Figure 2.3: Illustration of (A) Linear Tracking Process and (B) Nonlinear Pull-in Process of ANALOG DEVICES ADF4154

of its application in modulation and demodulation. Finer tuning resolution is always desired for advance communication systems which are containing more channels in a limited frequency bandwidth.

6. *Phase Noise and Spur Performance*: Phase noise and phase spurs are two distinct frequency domain representations of phase interruptions in a PLL. Phase noise shows continuous property in its frequency power spectrum, but phase spurs generate discrete power impulses at single frequencies.

   • *Phase Noise*: Phase noise power spectrum is frequency-domain power representation of phase noise's time-domain continuous random variations. Theoretically, phase noise power spectrum is the Fourier transform of phase noise's autocorrelation function in time domain. But autocorrelation functions for phase noises are hard to calculate and even impossible in most cases because phase noises are not wide-sense stationary processes [1]. Practically, engineers

use spectrum analyzers and phase-noise analyzers to approximately estimate power spectrums of passband phase noises and baseband phase noises respectively. Both equipments measure power (mW) of phase noises in a 1-Hz bandwidth at an offset frequency Δf from the carrier frequency, and the result is often displayed as logarithm of the ratio of the 1-Hz bandwidth noise power to the carrier power in the unit of dBc/Hz:

$$L(\Delta f)\ (\text{dBc/Hz}) = 10\log_{10}\left(\frac{\text{Noise Power in 1-Hz Bandwidth}}{\text{Carrier Power}}\right). \tag{2.1}$$

Except for its frequency domain representation, phase noise is also characterized in time domain as jitter with a unit of s/cycle, which measures the average root-mean-square error of the PLL actual output signal periods from the ideal PLL output period in a given time interval.

- *Phase Spur*: Phase spurs are frequency-domain power representation of continuous-time periodic phase interruptions in a PLL. Phase spurs show in the frequency spectrum as discrete spectral components with all their power concentrated at single frequencies. This can be explained by examining the power of a continuous-time periodic signal as the square of the Fourier coefficients of the signal. A phase spur is an infinite impulse and its power is represented by its underlying area which can be calculated by its integration over frequency. Neither passband spectrum analyzer nor baseband phase-noise analyzer can display infinite height impulses, but both equipments can estimate the power of a spectral component in the vicinity of its frequency. Similar to phase noise, the measured power (mW) of a discrete spectral component at an offset frequency Δf from the

carrier is compared with the carrier power and displayed as logarithm in the unit of dBc:

$$L(\Delta f)\ (\text{dBc}) = 10\log_{10}\left(\frac{\text{Spur Power}}{\text{Carrier Power}}\right). \tag{2.2}$$

Figure 2.4 demonstrates the output power spectrum of the ANALOG DEVICES ADF4154 fractional-N frequency synthesizer. As shown in the figure, phase noise is the continuous spectrum around the carrier and phase spurs are the discrete spectral components.

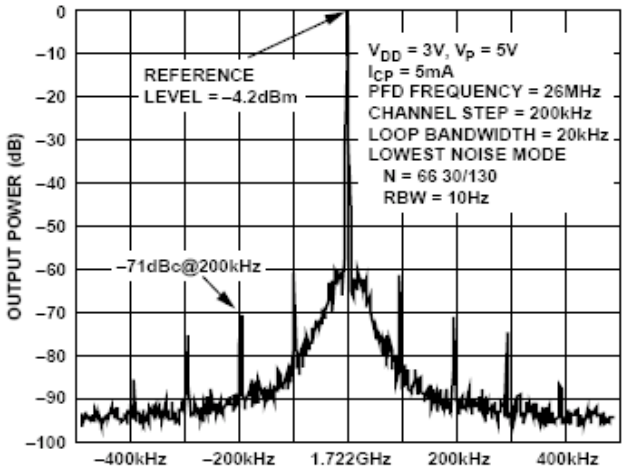


Figure 2.4: Output power spectrum of phase noise and phase spurs of ANALOG DEVICES ADF4154 fractional-N frequency synthesizer

2.3.2 Optimization and Tradeoff

There exist inherent relationships of each of the properties in a PLL frequency synthesizer design, where bandwidth serves as the key connection between all these properties. Figure 2.5 illustrates these relationships of improved bandwidth and the relative performance change of the other major parameters, where ↑ represents a

performance improvement and ↓ represents performance deterioration. Similar results for decreased bandwidth can be obtained by inverting the directions of the arrows.

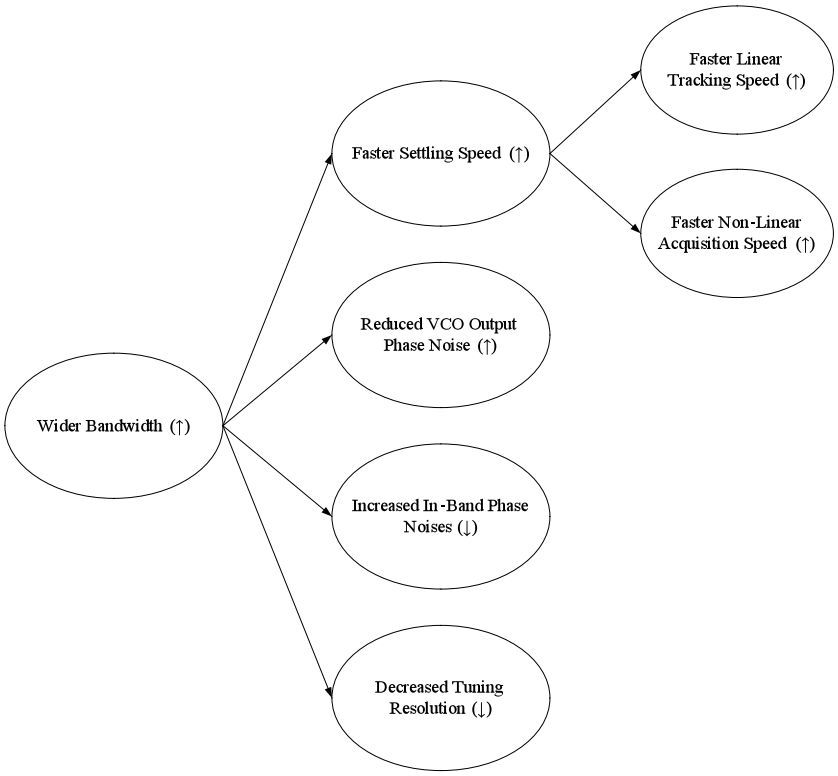


Figure 2.5: Relationships of bandwidth with other major parameters

2.4 Contributions of This Work

This thesis focuses on the design, analysis and application of a novel dual-mode cascaded-loop frequency synthesizer for the generation of GHz carrier frequencies and clocks with high accuracy and closely-spaced channel spacing. The contributions of the thesis are summarized below by improvements of the proposed dual-mode cascaded-loop frequency synthesizer over existing prevailing architectures.

- *Improvements over Integer-N Frequency Synthesizers:*

1. Fractional multiplication. Outputs of regular integer-N frequency synthesizers can only assume integer multiples of the input reference frequency. The proposed dual-mode cascaded-loop frequency synthesizer realizes fractional multiplication of the input reference frequency by its novel up(1^{st} loop)-down(divider)-up(2^{nd} loop) architecture while both the 1^{st} and 2^{nd} loops can be still integer-N PLLs individually.
2. Finer frequency resolution. Frequency resolution of integer-N frequency synthesizers is identical to the input reference frequency for its integer multiplication. The fractional multiplication realized by the proposed dual-mode cascaded-loop frequency synthesizer improves its frequency resolution to a small fraction of the input reference frequency.
3. Wider Bandwidth. The fractional multiplication function and finer frequency resolution enable application of large reference frequencies to the input of the proposed frequency synthesizer, which reduces frequency divider modulus in the feedback path of a PLL and equivalently extend the PLL bandwidth. Other benefits of extended PLL bandwidth include.
    - Faster settling speed for both nonlinear and linear frequency variations.
    - Reduced VCO out-of-band phase noise due to highpass filtering effects of the PLL bandwidth.
    - Minimized sizes for loop filter components for monolithic applications.

- *Improvements over Fractional-N Frequency Synthesizers:*

1. Reduced denominator for fractional multiples. For fractional multiplication, frequency resolution is inversely proportional to the denominator size. To

achieve the same frequency resolution, the proposed frequency synthesizer reduces its denominator size significantly smaller than a regular fractional-N frequency synthesizer. This favors circuitry designs for smaller and simpler accumulators whose overflows are commonly used to trigger fractional division mechanisms.

2. Far spaced fractional phase spurs. Fractional-N mechanism generates fractional phase spurs on the output spectrum with spacing from the carrier inversely proportional to its denominator size. The reduced denominator size for the proposed frequency synthesizer than a regular fractional-N frequency synthesizer pushes fractional phase spurs offset at further distances from the carrier with the spur strengths being suppressed by internal loop bandwidth without assistance from auxiliary devices.

- *Improvements over Dual-loop and Multi-loop Frequency Synthesizers:*

1. Elimination of frequency mixing. In all existing literatures of dual-loop and multi-loop frequency synthesizers, combinations and step increments for synthesized frequencies are implemented by mixing of the frequencies with mixers. Mixing is a nonlinear operation which generates large close-in harmonic spurs and increases 1/*f* noises around the carrier. The proposed frequency synthesizer realizes frequency combinations and step increments by applying adjustable multiplying factors and a dividing factor from its architecture to the input reference frequency without the appliance of mixers.
2. Free selection of synthesized frequency bands with arbitrary channel spacing. As per the date of writing of this thesis, all reported dual-loop and multi-loop

frequency synthesizers have preselected synthesis bands and fixed channel spacing. The proposed frequency synthesizer provides a versatile architecture to freely choose synthesized bands with arbitrary channel spacing in GHz ranges by optimally selecting moduli for its variable frequency dividers.

3. Single reference source. Most dual-loop and multi-loop frequency synthesizers require extra independent reference sources for frequency combinations. Due to the degrees of freedom of the free selection of synthesized bands with arbitrary channel spacing, the proposed frequency synthesizer requires just single reference source for all its applicable bands and channel spacing.

Contributions of this work have been presented in a journal paper submitted to IEEE transactions of VLSI system, which is currently being advised for correction and resubmission.

2.5 Organization of the Thesis

The thesis is organized as follows:

Chapter 3 presents a summary of existing prevalent frequency synthesizer architectures beginning with the fundamental integer-N frequency synthesizer. Subsequently, fractional-N frequency synthesizer, dual-loop frequency synthesizer and $\Delta\sum$ fractional-N frequency synthesizer are introduced individually.

Chapter 4 introduces analysis techniques for power spectra of common signals in PLL frequency synthesizers. Physical measuring principles of spectrum analyzers and

phase-noise analyzers are firstly presented. Secondly, calculation and estimation methods for power spectra of various common continuous-time signals in PLLs are presented for computer simulations. Thirdly, discrete approximation is introduced to approximate power spectrum of baseband modulation signals to phase-noise sidebands of modulated passband signals.

Chapter 5 describes the proposed architecture of a new dual-mode cascaded-loop frequency synthesizer with its synthesis modes and the respective synthesis formulas. Computer search results for the proposed architecture to synthesize channel frequencies for P-GSM 900, Bluetooth and an arbitrarily chosen band range and channel spacing are given in tables which demonstrate the degrees of freedom of the proposed architecture and the superiority of its frequency resolution.

Chapter 6 provides quantitative studies of the design and performances of the proposed dual-mode cascaded-loop frequency synthesizer. Passive RC loop filter structure is proposed to derive loop transfer functions of each of the constituent loops and the overall architecture. Stability issues are studied by allocating pole-zero positions of the loop transfer functions to achieve optimal phase margins for fastest settling speeds. Design procedures are proposed to summarize important characteristics of the proposed architecture. Discrete fractional phase spurs and continuous phase noises in PLLs are discussed with respect to the following issues:

- Modeling
- Filtering
- Discrete approximation to compare with carrier power

Chapter 7 simulates performances of the proposed architecture according to the quantitative discussions in Chapter 6 with its application to a P-GSM 900 uplink system. Advantages and superiorities of the proposed architecture are discussed respectively. Similar results can be obtained for other applications.

Chapter 8 summarizes the thesis and presents conclusions about the impact of the research. Additional research topics for future work are suggested.

Appendix A shows Matlab codes for the simulations in Chapter 7 by the quantitative analyses in Chapter 6.

Appendix B illustrates circuitries for a high-speed prescaler running above 10 GHz suitable for the design of high-speed frequency dividers for the proposed dual-mode cascaded-loop frequency synthesizer.

3 Architectures of Existing PLL Frequency Synthesizers

In this chapter, architectures of existing prevalent PLL frequency synthesizers will be discussed from the fundamental to the advanced. Although there are a variety of distinct PLL frequency synthesizer architectures, they are all derivatives of the same prototype of the integer-N frequency synthesizer. The basic properties deduced from the integer-N frequency synthesizer can be generally applied to its derivatives and serve as guidelines when designing more advance frequency synthesizer architectures. Therefore, in Section 3.1, we will first examine the basic properties of an integer-N frequency synthesizer including its linear model, phase-variation transfer functions, definitions of PLLs' order and type, and one of its typical implementations as charge-pump (CP) frequency synthesizers. Secondly, in Section 3.2, we will introduce an important variant of the integer-N frequency synthesizer as fractional-N frequency synthesizers in which the integer divider modulus will be replaced by a fractional divider modulus in the feedback loop from VCO to PFD. Thirdly, in Section 3.3, dual-loop architectures for frequency synthesizers will be introduced to generate fractional multiples of input reference frequency by arithmetic combinations of integer-N PLLs and frequency dividers. And lastly, Section 3.4 studies the advancement of regular fractional-N PLLs to $\Delta\sum$ fractional-N frequency synthesizers where divider quantization noise is randomized

by a digital $\Delta\sum$ modulator and its power spectrum is reshaped to push the majority of its power to further offsets from the carrier.

3.1 Integer-N Frequency Synthesizers

3.1.1 Transfer Functions in an Integer-N Frequency Synthesizer

Figure 3.1 shows the basic structure of an integer-N frequency synthesizer where an integer-N-modulus divider is inserted in the feedback loop of a regular PLL. The input-output frequencies can be written by a simple relationship as

$$f_{out} = N \cdot f_{in} . \tag{3.1}$$

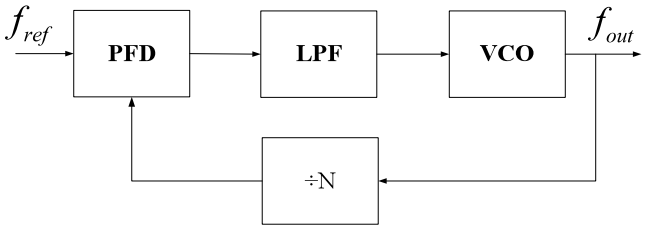


Figure 3.1: Basic structure of an integer-N frequency synthesizer

To study the dynamics of output phase and frequency responses due to small phase and frequency variations in a PLL frequency synthesizer, linear phase and frequency representations of the PLL system are desired. Because of similar natures of phase and frequency, the results obtained from a linear phase model for a PLL frequency synthesizer can be suitably applied to the linear frequency model for the same PLL frequency synthesizer. In this thesis, only phase models of frequency synthesizers will be

discussed, but the results obtained from phase models can be generally applied to their frequency models.

Figure 3.2 gives the linear phase representation of the basic structure of an integer-N frequency synthesizer shown in Figure 3.1 where each of the loop components has been replaced by their respective linear models in Laplace domain and their input-output relationships can be characterized by Laplace-domain linear transfer functions. Table 3.1 summarizes these relationships and affixes the variables with their appropriate units to approximate a real PLL circuit.

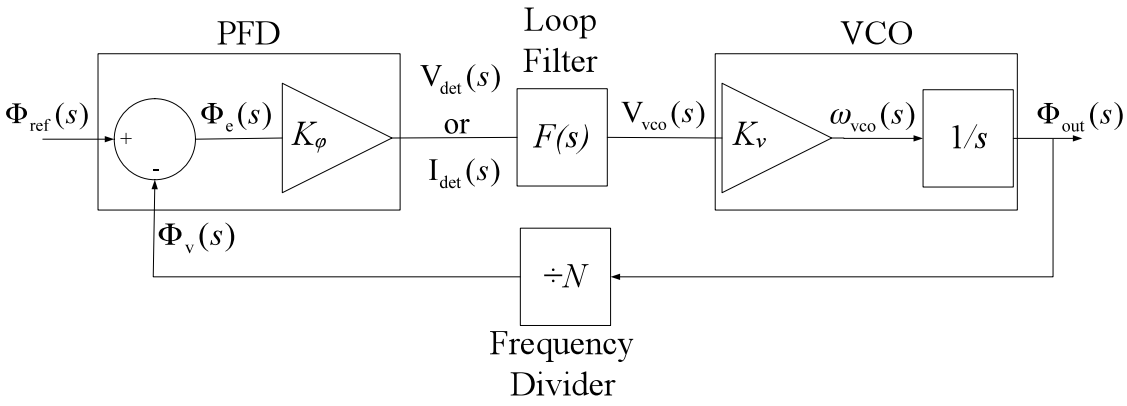


Figure 3.2: Linear phase model of the basic structure of an integer-N frequency synthesizer.

Table 3.1 gives the basic transfer functions of individual elements in an integer-N frequency synthesizer, system loop transfer functions can be obtained by connecting the individual basic transfer functions in specific combinations to describe the dynamic input-output relationships for the variables in the linear phase model shown in Figure 3.2. In the following, the most fundamental and important system transfer functions will be examined respectively, and, generally, these transfer functions will also be well

applicable in the study and design of other PLL frequency synthesizer architectures in later sections and chapters with only minor adjustments.

Table 3.1: Summary of variables in the linear phase model of an integer-N frequency synthesizer

Variable	Description	Unit
$\Phi_{ref}(s)$	Phase of the Input Reference Signal	rad
$\Phi_{v}(s)$	Phase of the Feedback Signal after the Integer-N-Modulus Divider	rad
$\Phi_{e}(s)$	Phase Error of the Two Input Signals to the PFD: $\Phi_{e}(s) = \Phi_{ref}(s) - \Phi_{v}(s)$	rad
K_{φ}	PFD Gain Depending on the type of the PFD, $\Phi_{e}(s)$ can be converted to either a voltage output or a current output. For a voltage output, K_{φ} is in a unit of V/rad; For a current output, K_{φ} is in a unit of A/rad.	V/rad or A/rad
$V_{det}(s)$ or $I_{det}(s)$	PFD Output Voltage or Current • For voltage type PFD, $V_{det}(s) = K_{\varphi} \cdot \Phi_{e}(s)$; • For current type PFD, $I_{det}(s) = K_{\varphi} \cdot \Phi_{e}(s)$.	V or A
$F(s)$	LPF Transfer Function For VCOs, the output of the LPF needs to be a voltage controlling signal. • For $V_{det}(s)$, $V_{vco}(s) = F(s) \cdot V_{det}(s)$ where $F(s)$ attaches no unit or a unit of V/V. • For $I_{det}(s)$, $V_{vco}(s) = F(s) \cdot I_{det}(s)$ where $F(s)$ is a transimpedance with a unit of V/A.	V/V or V/A
$V_{vco}(s)$	VCO Tuning Voltage	V
K_{v}	VCO Gain	rad/(s·V)
$\omega_{vco}(s)$	VCO Output Angular Frequency: $\omega_{vco}(s) = K_{v} \cdot V_{vco}(s)$	rad/s
$\Phi_{out}(s)$	PLL (or VCO) Output Phase: $\Phi_{out}(s) = (1/s) \cdot \omega_{vco}(s)$	rad

- Open-loop Transfer Function (Loop Gain):

$$H_{open}(s) = \frac{K_{\phi}K_{v}}{N}\frac{F(s)}{s}, \tag{3.2}$$

- Closed-loop Transfer Function (System Transfer Function):

$$H_{close}(s) = \frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = N\frac{\frac{K_{\phi}K_{v}}{N}\frac{F(s)}{s}}{1+\frac{K_{\phi}K_{v}}{N}\frac{F(s)}{s}} = N \cdot H_{L}(s), \tag{3.3}$$

- Error Transfer Function:

$$H_{\text{error}}(s) = \frac{\Phi_{\text{e}}(s)}{\Phi_{\text{ref}}(s)} = \frac{1}{1 + \frac{K_\phi K_v}{N}\frac{F(s)}{s}} = 1 - H_L(s), \tag{3.4}$$

- Feedback Transfer Function:

$$H_{\text{L}}(s) = \frac{\Phi_{\text{v}}(s)}{\Phi_{\text{ref}}(s)} = \frac{\frac{K_\phi K_v}{N}\frac{F(s)}{s}}{1 + \frac{K_\phi K_v}{N}\frac{F(s)}{s}} = \frac{H_{\text{open}}(s)}{1 + H_{\text{open}}(s)}. \tag{3.5}$$

As we have seen above, the feedback transfer function, $H_{\text{L}}(s)$ can be used as a convenient shortcut to write the other loop transfer functions. For example, the closed-loop transfer function can be written as $H_{\text{close}}(s) = N \cdot H_{\text{L}}(s)$ which is the divider modulus times the feedback transfer function, and the error transfer function can be written as $H_{\text{e}}(s) = 1\text{-}H_{\text{L}}(s)$ which is the unit complement of the feedback transfer function. This convenience of writing loop transfer functions in terms of $H_{\text{L}}(s)$ will prove useful in later contexts of this thesis for the study of other frequency synthesizer architectures.

3.1.2 PLL Order and PLL Type

In this section, two classifications of PLLs will be discussed as the PLL order and the PLL type. By ignoring constant terms in (3.3)-(3.5) such as K_φ, K_{v} and N, it can be discovered that the only changeable factor for the transfer functions of an integer-N frequency synthesizer is the LPF frequency response $F(s)$. For a practical LPF, its transfer function can be expressed as a rational function with real-coefficient polynomial numerators and denominators:

$$F(s) = \frac{\sum_{k=0}^{N_A} A_k s^k}{\sum_{k=0}^{N_B} B_k s^k}, \qquad N_A \in Z^+ \text{ and } N_B \in Z^+, \tag{3.6}$$

where N_A and N_B are the orders of the numerator and the denominator, and A_k and B_k are the coefficients of their respective polynomials. A LPF exhibits nonzero gain at the zero frequency and zero gain at the infinite frequency, which puts (3.6) into the following constraints:

$$\begin{aligned} &\lim_{s\to 0} F(s) \neq 0 \;\rightarrow\; A_0 \neq 0, \\ &\lim_{s\to \infty} F(s) = 0 \;\rightarrow\; N_B > N_A, \end{aligned} \tag{3.7}$$

where the constant term A_0 in the numerator of (3.6) can not be zero and the order of the denominator must be greater than the order of the numerator to ensure the lowpass characteristic.

The PLL order is defined as the denominator order of the PLL closed-loop transfer function. Substituting (3.6) into the feedback transfer function $H_{\mathrm{L}}(s)$ in (3.5), which is a close resemblance of the closed-loop transfer function of (3.3), we have

$$\begin{aligned} H_{\mathrm{L}}(s) &= \frac{\frac{K_\phi K_v}{N}\frac{F(s)}{s}}{1+\frac{K_\phi K_v}{N}\frac{F(s)}{s}} = \frac{\frac{K_\phi K_v}{N}F(s)}{s+\frac{K_\phi K_v}{N}F(s)} \\ &= \frac{\frac{K_\phi K_v}{N}\frac{\sum_{k=0}^{N_A} A_k s^k}{\sum_{k=0}^{N_B} B_k s^k}}{s+\frac{K_\phi K_v}{N}\frac{\sum_{k=0}^{N_A} A_k s^k}{\sum_{k=0}^{N_B} B_k s^k}} = \frac{\frac{K_\phi K_v}{N}\sum_{k=0}^{N_A} A_k s^k}{\sum_{k=0}^{N_B} B_k s^{k+1}+\frac{K_\phi K_v}{N}\sum_{k=0}^{N_A} A_k s^k}, \end{aligned} \tag{3.8}$$

where the denominator order is N_B+1. Thus, we can conclude that the order of an integer-N frequency synthesizer is its LPF denominator order plus one, and this same rule is suitable for other PLL frequency synthesizer architectures.

Another classification of a PLL is by its type which is defined as the number of zero-frequency poles for its open-loop transfer function. Because the VCO in a PLL has already provided a zero-frequency pole in its open-loop transfer function, which can be seen by the integration-effect Laplace-domain factor, $1/s$, in (3.2), and according to circuit theories, there could only be at most two coincident poles on the imaginary axis including zero frequency for a stable circuitry [7], we can conclude that, for a realistic PLL, the maximum number of zero-frequency poles for its LPF is one. This restriction of maximum one extra pole in the LPF will give us some caution in our future design of LPFs. Therefore, a Type-I PLL contains only one zero-frequency pole in its open-loop transfer function and implies that there is no zero-frequency pole in its LPF; a Type-II PLL contains two zero-frequency poles in its open-loop transfer function and implies that there is another zero-frequency pole in its LPF.

3.1.3 Linear Modeling of Charge-Pump Integer-N Frequency Synthesizers

Figure 3.2 gives the general linear model of PLL frequency synthesizers where PFD has been represented by a linear phase comparator and the phase difference is ideally being converted to a voltage or current output by the multiplying factor K_φ (A/rad or V/rad). In a physical circuit implementation of PLL frequency synthesizers, the PFD can be realized by a variety of circuit architectures. Thus, the determination of the PFD linearization factor K_φ when it is operating in its linear range is the first issue we need to consider before applying the transfer functions of (3.2)-(3.5) to analyze the PLL

performance for a specific PLL circuit. Shown in Figure 3.3 is a common circuit realization of a PFD, which contains two D-flip-flops, an AND gate, and a charge-pump (CP). The two D-flip-flops and the AND gate are used to detect the phase differences of its two input signals. And the charge pump consisting of two identical current sources and two independent switches is used to convert the phase differences into current output signals. Detail operations of the flip-flops and the CP can be referred to [7]. And our interest is the equivalence of the circuit implementation in Figure 3.3 and its linearized model in Figure 3.2. As in [7], the operation of the PFD circuit in Figure 3.3 can be approximated by a linear PFD gain, K_φ, as

$$K_\phi = \frac{I}{2\pi}, \tag{3.9}$$

where I (A) is the current of both current sources in the CP. Shown in Figure 3.3, there is also a simple circuit implementation of the LPF, which is a first-order lowpass filter consisting of a single resistor and a single capacitor. In a CP PLL, the LPF smoothes the fast-fluctuating current pulses from the PFD and convert them into a relatively steady control voltage for the following VCO with a transimpedance transfer function as

$$F(s) = R + \frac{1}{sC}, \tag{3.10}$$

where the unit of $F(s)$ is V/A. The linearization of PFD by (3.9) and the loop filter transfer function (3.10) provide us the opportunity to study dynamic behaviors of the CP PLL integer-N frequency synthesizer in Figure 3.3 for its locked status and the derived results nonetheless will be inspiring for its non-locked status and other PLL frequency synthesizer architectures. Substituting (3.9) and (3.10) into (3.2) and (3.3) respectively, we have

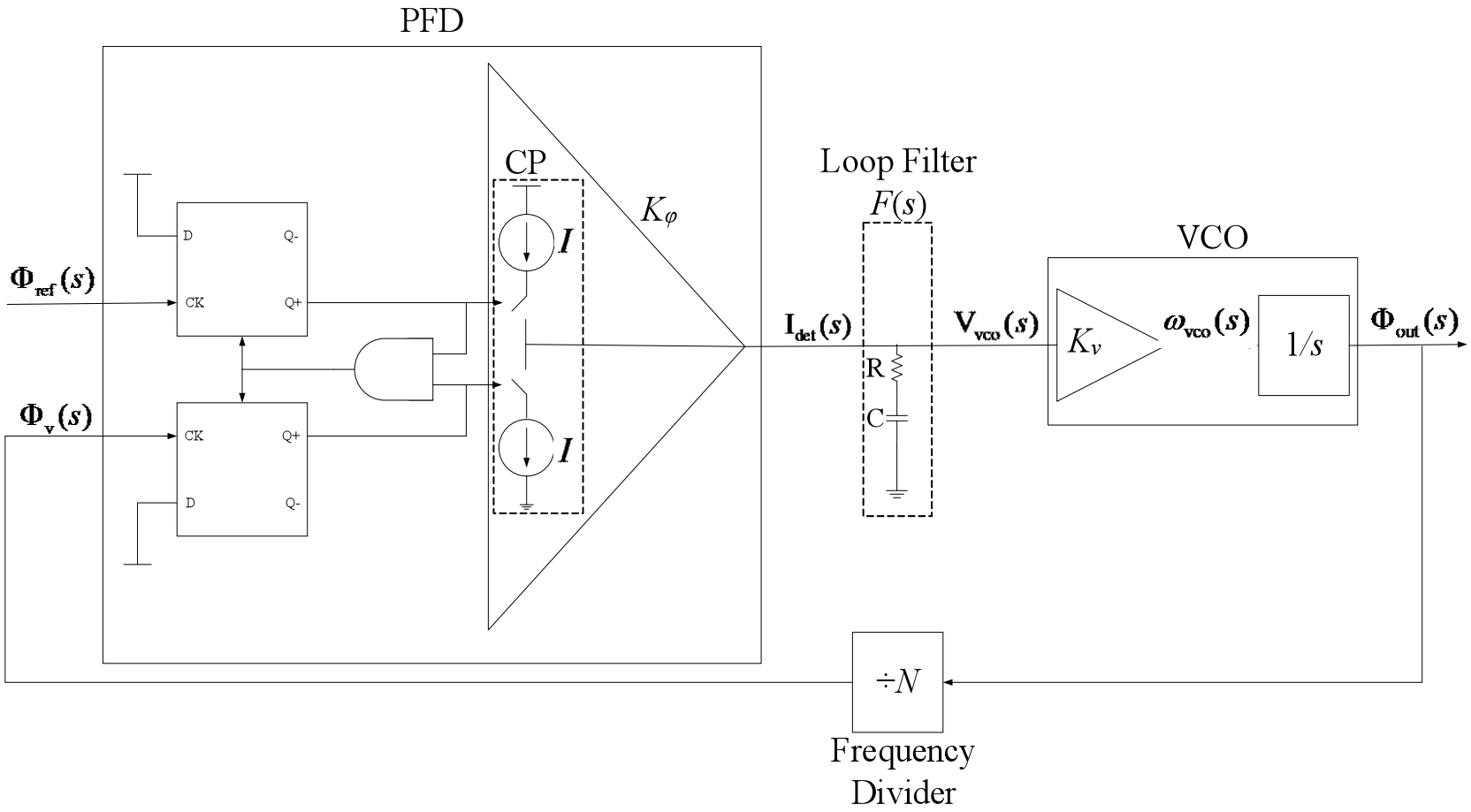


Figure 3.3: A circuit implementation of the PFD, CP and LPF for the linear phase model of an integer-N frequency synthesizer in Figure 3.2

- Open-loop Transfer Function (Loop Gain) of Figure 3.3:

$$H_{\text{open}}(s) = \frac{K_\phi K_v}{N}\frac{F(s)}{s} = \frac{I}{2\pi}\frac{1}{N}\left(R + \frac{1}{sC}\right)\frac{K_v}{s}, \tag{3.11}$$

and

- Closed-loop Transfer Function (System Transfer Function) of Figure 3.3:

$$H_{\text{close}}(s) = \frac{\Phi_{\text{out}}(s)}{\Phi_{\text{ref}}(s)} = N\frac{\frac{K_\phi K_v}{N}\frac{F(s)}{s}}{1 + \frac{K_\phi K_v}{N}\frac{F(s)}{s}} = \frac{\frac{IK_v}{2\pi C}(RCs+1)}{s^2 + \frac{I}{2\pi}\frac{K_v}{N}Rs + \frac{I}{2\pi C}\frac{K_v}{N}}. \tag{3.12}$$

The open-loop transfer function (3.11) has two poles at its zero frequency and the closed-loop transfer function (3.11) contains a 2nd-order term in its denominator. Thus, the CP PLL integer-N frequency synthesizer shown in Figure 3.3 is a type-II 2nd-order PLL system. For a 2nd-order linear system, the theories of signals and systems provide it a general model by a rational 2nd-order transfer function [12]:

$$H(s)\big|_{\text{General}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{3.13}$$

where ζ is referred to as the damping ratio and ω_n as the undamped natural frequency. Comparing the closed-loop transfer function in (3.12) and the general transfer function in (3.13), and ignoring their respective numerators, we can identify the damping ratio, ζ and the undamped natural frequency, ω_n, for the PLL system in Figure 3.3:

$$\begin{cases} 2\zeta\omega_n = \dfrac{I}{2\pi}\dfrac{K_v}{N} \\ \omega_n^2 = \dfrac{I}{2\pi C}\dfrac{K_v}{N} \end{cases} \Rightarrow \begin{cases} \omega_n = \sqrt{\dfrac{I}{2\pi C}\dfrac{K_v}{N}} \\ \zeta = \dfrac{R}{2}\sqrt{\dfrac{IC}{2\pi}\dfrac{K_v}{N}} \end{cases}. \tag{3.14}$$

Both ω_n and ζ are in a reverse relationship with the divider modulus, N, which will give us enough cautions in our future designs, because, for a 2nd-order system, the closed-

loop bandwidth can be well approximated by the undamped natural frequency, ω_n, and its time-domain settling speeds for impulse and step responses are characterized by the damping ratio, ζ. A larger divider modulus, N, results in a smaller ω_n and a smaller ζ, which in turn implies a smaller closed-loop bandwidth and a slower settling speed for a specifically designed PLL system. As in a 1st-order system where a time constant is defined, a similar definition can be applied for a time constant for a 2nd-order system from its explicit time-domain impulse and step responses [12]:

$$\tau = \frac{1}{\zeta\omega_n} = \frac{4\pi N}{RIK_v}, \tag{3.15}$$

where the time constant, τ, is linear with the divider modulus, N, which indicates that a larger divider modulus, N, results in a larger time constant and a slower system response and once again confirms the reverse relationship mentioned above.

3.1.4 Frequency Resolution and PLL Bandwidth

Even though the reverse relationship of the divider modulus, N, and the PLL closed-loop bandwidth was demonstrated by a 2nd-order integer-N frequency synthesizer, it is also a general relationship for higher-order PLLs and other PLL frequency synthesizer architectures. As in (3.1), the output frequency of an integer-N frequency synthesizer can only assume integer multiples of the input reference frequency with a minimum output frequency increment the same as the input reference frequency. For modern communication applications with high carrier frequency band and small channel spacing, the integer-N frequency synthesizer suffers from a number of critical drawbacks. For example, to generate the GSM carrier frequency band listed in Table 2.1 from 890 MHz to 915 MHz with a channel spacing of 200 kHz, an integer-N frequency synthesizer

has to assume an input reference frequency of 200 kHz and the divider modulus has to vary from 4450 for 890MHz to 4575 for 915 MHz. This large divider modulus, or, equivalently, this small input reference frequency, causes an extremely small PLL bandwidth and slow PLL system response and the resulted frequency-switching settling time is far beyond the standard requirements of 500 μs to 850 μs. In addition, another concern of narrowed closed-loop bandwidth is the increased amount of the VCO noise being conveyed to the output of a PLL system. The VCO is one of the noisiest components in a PLL and its self-generated phase noise is highpass filtered by the closed-loop bandwidth before it arrives at the output of the PLL system [7]. A small closed-loop bandwidth implies there will be a large portion of unfiltered VCO noises leaking out to the PLL output, which will seriously downgrade the overall performance of the PLL and make it unacceptable for most communication systems.

3.2 Fractional-N Frequency Synthesizers

The inherent contradiction of frequency resolution (channel spacing) and the PLL bandwidth in an integer-N frequency synthesizer prompted the application of fractional frequency division in the feedback loop of a PLL frequency synthesizer. Considering the generation of a 912.2-MHz carrier frequency in a GSM communication system with 200-kHz channel spacing, an integer-N frequency synthesizer requires a 200-kHz reference frequency and an integer divider modulus of 912.2 MHz / 200 kHz = 4561, which, as indicated in Section 3.1.4, causes seriously reduced PLL bandwidth and slow system response. Instead of restricting its divider moduli to integer numbers, a fractional-N frequency synthesizer applies a larger reference frequency, e.g., 10 MHz to generate the

same carrier frequency and channel spacing with a fractional divider modulus of 912.2 MHz / 10 MHz = 91.22 = 91 + (11/50), where the fractional portion of 11/50 enables the output frequency to be a fractional multiple of the input reference frequency and the denominator guarantees the required frequency resolution. The substantially reduced divider modulus in a fractional-N frequency synthesizer provides opportunities for large reference frequencies and resulted increased PLL bandwidths.

But an actual frequency divider circuit can only divide integer numbers. The fractional division can be realized by the averaging effect of the togging of two integer moduli for an integer-N frequency divider. Shown in Figure 3.4 is the block diagram of an implementation of a fractional-N frequency divider. The *Q*-modulus accumulator is clocked by the divided VCO frequency f_v and sums its own output with the given addend *A* for each clock period. If the sum is over its modulus *Q*, there will be an overflow signal from the accumulator to switch the modulus of the integer-N frequency divider from *N* to *N*+1.

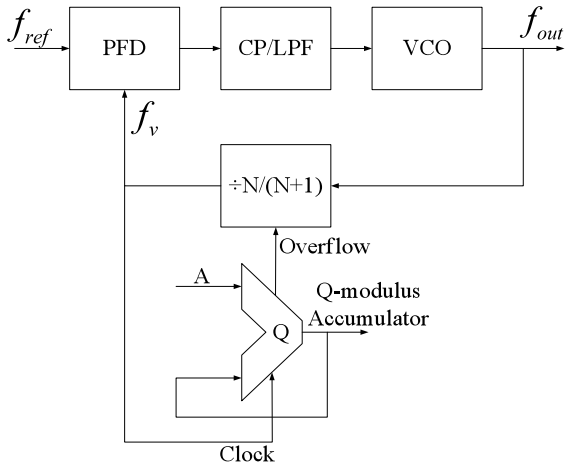


Figure 3.4: An implementation of a fractional-N frequency divider

The overflow signal is a periodic signal with period $Q \cdot T_v$, $T_v = 1/f_v$, during which there are A overflows, so the dual-modulus integer-N frequency divider divides by $N+1$ in A clock periods and by N in the rest Q-A clock periods and the average division factor can be written as:

$$N^* = \frac{(N+1)AT_v + N(Q-A)T_v}{QT_v} = N + \frac{A}{Q}. \tag{3.16}$$

The output frequency of a fractional-N frequency synthesizer can be written as:

$$f_{out} = N^* f_{ref} = \left(N + \frac{A}{Q}\right) f_{ref}, \tag{3.17}$$

where the frequency resolution is controlled by the accumulator modulus Q and can be written as

$$\Delta f_{out} = f_{ref} \cdot \frac{1}{Q}. \tag{3.18}$$

It seems from (3.18) that a fractional-N frequency synthesizer can achieve frequency resolutions arbitrarily small and reference frequencies arbitrarily large by keeping the accumulator modulus Q as large as possible. But it is also observed from Figure 3.4 that the overflow signal from the accumulator periodically modulates the feedback VCO-divided frequency f_v and generates periodic phase errors in the PFD. The phase errors pass through the LPF and modulate the output frequency of the VCO, which in the frequency domain manifests themselves as phase spurs around the carrier frequency. As discussed in later context, the larger the accumulator modulus Q is, the closer the spurs are located around the carrier frequency and the harder the PLL can filter out these spurs.

Assuming the PLL is locked, the feedback VCO-divided frequency f_v can be regarded as changing immediately after the switching of the modulus of the dual-modulus integer-N frequency divider and f_v can be written as

$$f_v = \begin{cases} f_{ref}\left(N+\dfrac{A}{Q}\right)\dfrac{1}{N} > f_{ref}, & \text{if overflow = 0} \\ f_{ref}\left(N+\dfrac{A}{Q}\right)\dfrac{1}{N+1} < f_{ref}, & \text{if overflow =1} \end{cases}, \tag{3.19}$$

where f_v is always above or below the reference frequency f_{ref} during each of the clock periods T_v. And the resulted phase error in the PFD relative to the input reference frequency in one of the clock periods can be written as

$$\Delta\varphi_j = 2\pi\frac{T_v - T_{ref}}{T_{ref}} = 2\pi\frac{f_{ref} - f_v}{f_v} = \frac{1}{N+\dfrac{A}{Q}}\frac{2\pi}{Q}\cdot\begin{cases} -A, & \text{if overflow=0} \\ Q-A, & \text{if overflow=1'} \end{cases} \tag{3.20}$$

where $T_v = 1/f_v$, $T_{ref} = 1/f_{ref}$ and the subscript index j is to record the time moments for phase errors during a specific clock period. $\Delta\varphi_j$ constitutes a staircase phase-error sequence in the PFD and is converted to voltage or current error signals for the PFD gain K_φ as:

$$\begin{aligned} I_{\text{det\_error}}(\text{A}) &= K_\varphi(\text{A/rad})\cdot\sum_j \Delta\varphi_j, \quad j\in N, \\ \text{or } V_{\text{det\_error}}(\text{V}) &= K_\varphi(\text{V/rad})\cdot\sum_j \Delta\varphi_j, \quad j\in N, \end{aligned} \tag{3.21}$$

where K_φ takes the unit of A/rad for current-type CP-PFD and V/rad for voltage-type PFD. Because of the periodic overflow signal from the accumulator in Figure 3.4, the phase-error signals $I_{\text{det\_error}}$ and $V_{\text{det\_error}}$ are also periodic with period $Q\cdot T_v$ and according to chapter 4.1.2, their power spectra can be calculated as:

$$S_{\text{det\_error}}(f)\left(\text{A}^2/\text{Hz or V}^2/\text{Hz}\right)=\sum_{k=-\infty}^{+\infty}|a_k|^2\,\delta\left(f-k\frac{1}{Q\cdot T_{ref}}\right)$$
$$=\sum_{k=-\infty}^{+\infty}|a_k|^2\,\delta\left(f-k\frac{f_{ref}}{Q}\right),\qquad k=\pm1,\pm2,\pm3,\ldots \qquad (3.22)$$

where a_k is the Fourier series coefficients of the periodic signals $I_{\text{det\_error}}$ and $V_{\text{det\_error}}$ and their discrete spectral powers (phase spurs) are concentrated at the harmonics of the fundamental frequency $1/(Q\cdot T_{\text{ref}})$ as infinite-height zero-width impulses. $I_{\text{det\_error}}$ and $V_{\text{det\_error}}$ are baseband signals and their power spectrum $S_{\text{det\_error}}(f)$ modulates the VCO and appears as discrete sideband phase spurs around the carrier frequency on the output spectrum of the fractional-N PLL frequency synthesizer, which is denoted by $S_{\text{spur}}(\Delta f)$ and can be derived from the PLL's system transfer function as:

$$S_{\text{spur}}(\Delta f)\ \left(\text{A}^2/\text{Hz or V}^2/\text{Hz}\right)=\left|\frac{N^*H_L(\Delta f)}{K_\varphi}\right|^2 S_{\text{det\_error}}(\Delta f)$$
$$=\frac{K_v\dfrac{F(j2\pi\Delta f)}{j2\pi\Delta f}}{1+\dfrac{K_\phi K_v}{N}\dfrac{F(j2\pi\Delta f)}{j2\pi\Delta f}}S_{\text{det\_error}}(\Delta f) \qquad (3.23)$$
$$=\sum_{k=-\infty}^{+\infty}|a_k|^2\frac{K_v\dfrac{F(j2\pi\Delta f)}{j2\pi\Delta f}}{1+\dfrac{K_\phi K_v}{N}\dfrac{F(j2\pi\Delta f)}{j2\pi\Delta f}}\delta\left(\Delta f-k\frac{f_{ref}}{Q}\right)$$

where Δf is the offset frequencies from the carrier frequency; $H_L(\Delta f)$ is the feedback transfer function in (3.5) with the s-domain variable s replaced by its frequency response $j2\pi\Delta f$.

As seen in (3.22) and (3.23), discrete sideband phase spurs originated from the fractional-N architecture in Figure 3.4 are located at offset frequencies Δf_{spur} around the carrier frequency as

$$\Delta f_{spur} = k\frac{f_{ref}}{Q}, \qquad k = \pm 1, \pm 2, \pm 3, \ldots, \tag{3.24}$$

where the spacing between two adjacent spurs is the same as the frequency resolution Δf_{out} in (3.18), which manifests the conflict of frequency resolution and phase-spur spacing for a fractional-N frequency synthesizer.

Illustrated in Figure 3.5 is a Matlab simulation of phase-spur strength and locations on the output spectrum of a fractional-N frequency synthesizer to generate 912.2-MHz carrier frequency for a P-GSM communication system with f_{ref}=10 MHz, Q =50 and A =11. The generated phase spurs on the output spectrum are closely located at offset frequencies $\Delta f_{spur} = k\cdot(f_{ref}/Q) = k\cdot$(10 MHz /50) $= k\cdot$(200 kHz), $k = \pm 1, \pm 2, \pm 3, \ldots$

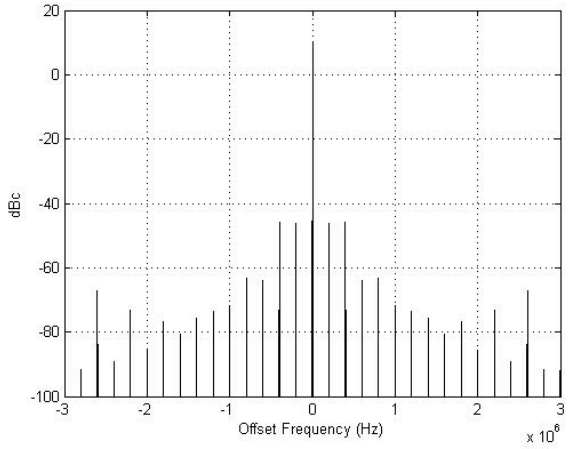


Figure 3.5: Simulated double-sided phase-spur strength and locations on the output

3.3 Dual-Loop Frequency Synthesizers Based on Integer-N PLLs

The fractional-N frequency synthesizer architecture presented in Section 3.2 overcomes the conflict of frequency resolution (channel spacing) and PLL bandwidth (settling speed) for integer-N frequency synthesizers, but, at the same time, generates

another conflict of its frequency resolution (channel spacing) and the spacing of fractional phase spurs arising from the periodic phase errors in its PFD. The fractional phase spurs are closely located around the synthesized carrier frequency and can be very large such that unless the PLL bandwidth is very small to suppress the spurious tones, the fractional-N frequency synthesizer can hardly be applied to most practical applications, but a small PLL bandwidth negates the potential benefits of applying the fractional-N technique. In order to generate fractional multiples of the input reference frequency by solely applying integer-N PLLs, dual-loop frequency synthesizer architectures have been proposed in which integer-N PLLs serve as frequency multipliers (numerators) and frequency dividers serve as division factors (denominators) and the overall effect of the dual-loop frequency synthesizer is a fractional multiple of the input reference frequency.

Shown in Figure 3.6 is the block diagram of a dual-loop frequency synthesizer architecture proposed in [2]. The final output frequency from its VCO1 can be represented in terms of the two input reference frequencies, f_{ref1} and f_{ref2} as:

$$f_{\text{out}} = N \times f_{\text{ref1}} + M \times \left(\frac{f_{\text{ref2}}}{X} \right), \tag{3.25}$$

where the integer-frequency multiplications are realized by the integer-N PLLs; the integer-frequency division is realized by the bridging frequency divider "/X"; the frequency addition is realized by the operation of the mixer; and the minimum synthesized output frequency step is controlled by the term, f_{ref2}/X.

Dual-loop frequency synthesizer architectures realize fractional multiplications of the input reference frequency without generating fractional phase spurs as in a fractional-N frequency synthesizer by deploying an extra integer-N PLL and an integer-frequency divider. The architecture also offers designers extra degrees of freedom to tradeoff

bandwidths, settling speeds and phase-noise reductions in between the two PLLs. But a significant drawback of the current dual-loop architectures is that they inevitably apply mixers to implement arithmetic operations of the different frequency combinations, which may generate large harmonic components around the desired carrier frequency because of the nonlinearities of the mixers. The large harmonic components are closely located around the carrier frequency and are difficult to remove solely by the filtering of the constituent loops' bandwidths. The nonlinearities of the mixers may also increase the 1/*f* noises around the carrier frequency [5], [6]. Another drawback of the current dual-loop architectures is the requirements of extra independent reference sources for each PLL, which is not practical for many applications.

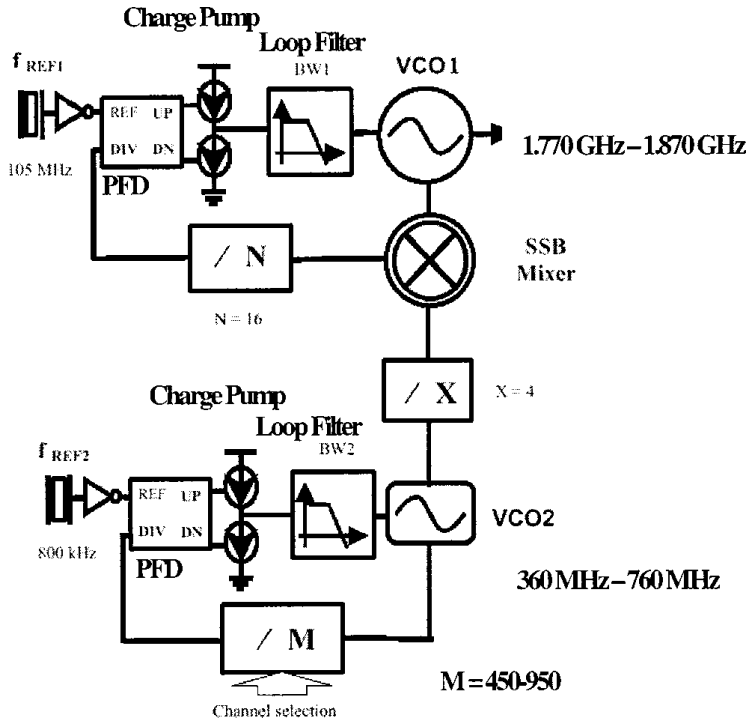


Figure 3.6: Block diagram of a dual-loop frequency synthesizer architecture proposed in [2]

3.4 Delta-Sigma (∆∑) Frequency Synthesizers Based on Fractional-N PLLs

Fractional phase spurs on the output spectrum of the fractional-N frequency synthesizer architecture presented in Section 3.2 arise from the periodic phase errors in its PFD which are caused by the periodic toggling between integer moduli to achieve average fractional division for the frequency divider in the feedback loop of the PLL. For the fractional frequency divider in Figure 3.4, it can be redrawn as in Figure 3.7 in which the fractional frequency divider can be alternatively represented as $\div(N + y[n])$, where $y[n] = +1$ or -1 and n denotes the clock sequence. The fractional average effect of $y[n]$ can be decomposed as $y[n] = x + e_m[n]$, where the x is the desired fractional part of the average divider modulus, i.e. $x = A/Q$ in (3.16); and $e_m[n]$ is undesired zero-mean quantization noise caused by using integer moduli in place of the ideal fractional value. $e_m[n]$ corresponds to the phase errors it causes in the PFD and is periodic for the modulation of the overflow signals from the Q-modulus accumulator. Therefore, there are discrete phase spurs on the spectrum of the output synthesized carrier frequency.

If the periodicity of $e_m[n]$ can be broken, there will not exist discrete phase spurs on the output spectrum of the fractional-N frequency synthesizer in Figure 3.7. This is the basic principle for a ∆∑ fractional-N frequency synthesizer which generates the sequence of moduli $y[n]$ such that the quantization noise $e_m[n]$ is not periodic and has most of its power in a frequency band well above the desired bandwidth of the PLL. Shown in Figure 3.8 is an example of a ∆∑ fractional-N frequency synthesizer introduced in [7], where the PLL core is similar to the one in Figure 3.7 with the Q-modulus accumulator replaced by a digital ∆∑ modulator. The details of how a digital ∆∑ modulator works are presented in [1], [7] and [9] and its main purpose is to coarsely quantize its input

sequence, $x[n]$, such that $y[n]$ is integer values and has the form: $y[n] = x[n-k] + e_m[n]$, where the parameter k is determined by the order of the digital $\Delta\sum$ modulator with a specific configuration and $e_m[n]$ is dc-free quantization noise with most of its power outside the PLL bandwidth. The sequence $x[n]$ consists of the desired fractional part of the divider modulus, A/Q, plus a small, pseudo-random, 1-bit sequence. The pseudo-random sequence is necessary to avoid spurious tones in the $\Delta\sum$ modulator's quantization noise, but its amplitude is very small so it does not appreciably increase the phase noise of the PLL [9]

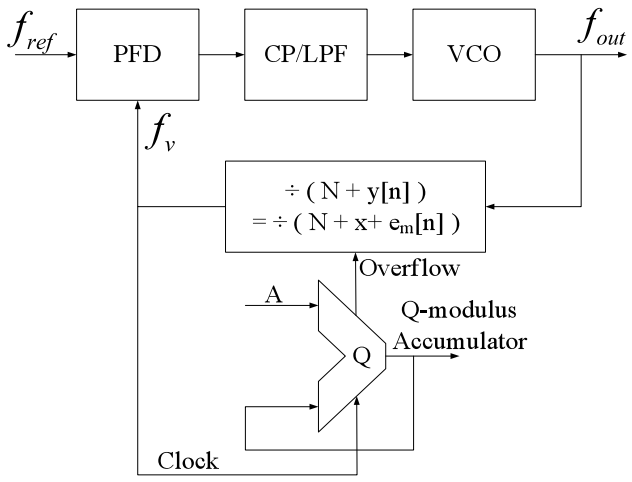


Figure 3.7: Remodeling of the fractional frequency divider in Figure 3.4

Shown in Figure 3.9 is a Matlab simulation of the power spectrum of the quantization noise $e_m[n]$ after being modulated by the VCO and appearing at the output of a $\Delta\sum$ fractional-N frequency synthesizer with $x = A/Q = 11/50$ and f_{ref}=10 MHz for a 912.2 MHz carrier frequency in a P-GSM communication system. The illustrated power spectrum has been truncated from 300 kHz to 2 MHz and its power increases from low

offset frequencies to high offset frequencies with the majority of the power being pushed above a practical PLL's bandwidth. This is the key property for the ΔΣ fractional-N architecture. A ΔΣ fractional-N PLL's bandwidth can be designed significantly wider than its regular fractional-N counterpart while maintaining a cleaner output power spectrum, because the power spectrum of its quantization noise $e_m[n]$ contains no discrete phase spurs and has most of its power at far offsets from the carrier, which can be effectively filtered solely by the PLL's bandwidth

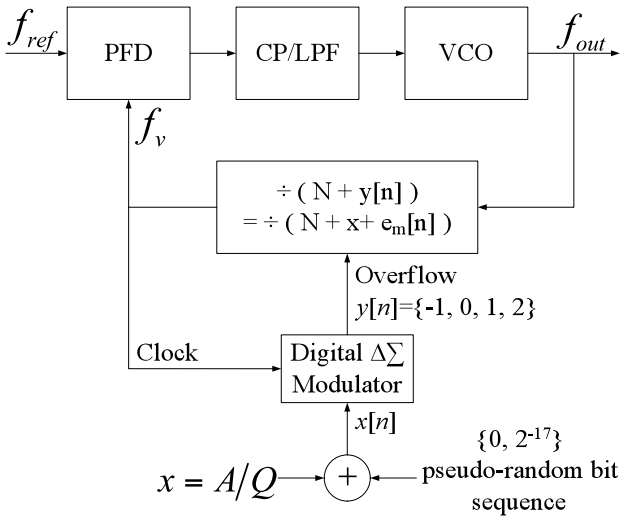


Figure 3.8: A ΔΣ fractional-N frequency synthesizer example

The wider bandwidth and cleaner output power spectrum make ΔΣ fractional-N frequency synthesizers much more attractive than the regular fractional-N architecture, but the cost of reshaping the quantization noise is the large extra chip area for the digital ΔΣ modulator and the generation of the psuedo-random bit sequence. The processes of digital ΔΣ modulation increase the complexity of the circuit and may not be convenient for monolithic applications.

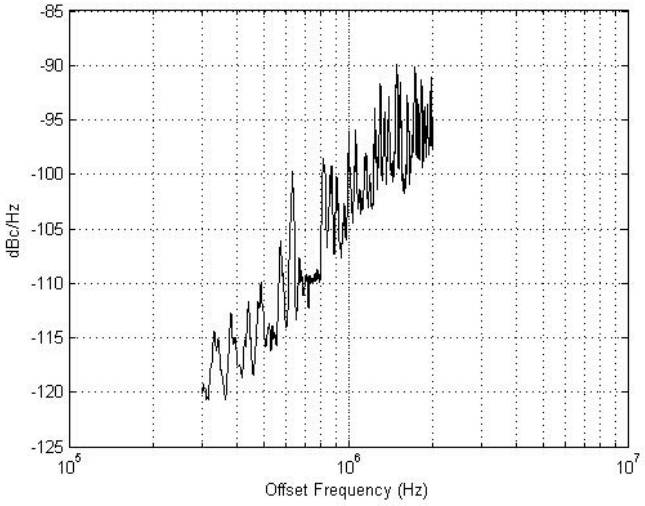


Figure 3.9: Simulated power spectrum of the quantization noise $e_m[n]$ at the output of a $\Delta\sum$ fractional-N frequency synthesizer with offset frequencies truncated from 300 kHz to 2 MHz

3.5 Summary

This chapter presented the architectures, model linearization and working principles of existing frequency synthesizers prevailing in academia and in industry. The pros and cons of each of the architectures can be summarized as follows:

1. The integer-N frequency synthesizer architecture introduced in Section 3.1 is the fundamental core for other PLL frequency synthesizer architectures. It is simple to design but suffers from its inherent contradiction of frequency resolution (channel spacing) and PLL bandwidth (settling speed), which makes the integer-N architecture not suitable for many modern communication applications with high carrier frequency and close channel spacing.

2. The fractional-N frequency synthesizer architecture introduced in Section 3.2 overcomes the inherent contradiction of an integer-N frequency synthesizer by

replacing integer divider moduli with fractional divider moduli to achieve smaller divider moduli, finer frequency resolution and wider PLL bandwidth. But the periodic toggling of the divider moduli, at the same time, generates periodic phase errors in its PFD, which manifest themselves as discrete phase spurs on the output spectrum. The phase spurs are very strong and located at offset frequencies equal to the designed frequency resolution. In order to filter the discrete phase spurs, the bandwidth of a fractional-N frequency synthesizer has to be set at a very small value, which creates another contradiction of frequency resolution and PLL bandwidth.

3. The dual-loop frequency synthesizer architecture introduced in Section 3.3 generates fractional multiples of the input reference frequency by utilizing integer-N PLLs as frequency multipliers and frequency dividers as frequency division factors. Because the constituent loops are integer-N PLLs, the dual-loop architecture does not generate fractional phase spurs as in the fractional-N architecture in Section 3.2 and it also provides designers opportunities to tradeoff bandwidths, settling speeds and phase-noise reductions in between the two loops. But this architecture inevitably applies mixers to realize arithmetic combinations of the different fractional frequencies from its individual components. Mixers are nonlinear devices and generate large amount of harmonics around the carrier frequency which is present within the PLL bandwidth, plus the application of dual reference frequencies is inconvenient for many applications.

4. The $\Delta\sum$ fractional-N frequency synthesizer architecture introduced in Section 3.4 removes fractional phase spurs from its output spectrum by randomizing the quantization noise in its feedback-loop frequency divider. The randomization reshapes the power spectrum of the resulted phase errors in its PFD to have small power components on its lower frequency band and most of the powers pushed to its higher frequency band. This reshaped power spectrum allows a wider PLL bandwidth and cleaner output spectrum for the $\Delta\sum$ fractional-N architecture. But the cost of the performance superiority is the requirements of large chip area and complexity of the digital $\Delta\sum$ modulator.

4 Analysis of Power Spectra of Signals in PLL Frequency Synthesizers

Signals in a PLL frequency synthesizer can be categorized as passband signals and baseband signals. Passband Signals are usually referred to the output signals from PLLs with their general form written as:

$$v_o(t) = (A + a(t))\cos(2\pi f_o t + \phi(t)), \tag{4.1}$$

where A is the mean amplitude of the passband signal; $a(t)$ is a zero-mean random process and can be ignored in passband signals' power spectrum analyses because random amplitude fluctuations in a VCO can be largely suppressed by amplitude-control mechanisms; f_0 is the synthesized output carrier frequency with a steady phase of $2\pi f_0 t$; $\phi(t)$ is an assembly of all phase deviations from the steady phase $2\pi f_0 t$, which includes transferred phase noises shown at the output of the PLL from all phase-noise sources in the PLL, the initial phase of the VCO and the integrated effects of VCO frequency drifts. $\phi(t)$ is a baseband signal because its frequency is much slower than the carrier frequency f_0 and the study of the power spectrum of $\phi(t)$ can provide us a window to speculate the power spectrum of the passband signal $v_o(t)$, which will be discussed in Section 4.3 of this chapter.

Not every signal has its power spectrum defined in theory. For example, a continuous-time aperiodic signal only has an energy spectrum and no defined power spectrum [13]; and more commonly in PLLs, phase noises in PLL components are often

multiplicative and nonstationary and there is no firm theoretical definitions for such signals [1]. But we do see power spectra displayed on measuring equipments such as spectrum analyzers and phase-noise analyzers in laboratories. To explain this gap between the theoretical power spectra and the physically measured power spectra, in this chapter, we first discuss the working principles of a spectrum analyzer and a phase-noise analyzer in Section 4.1. As we will see, spectrum analyzers are suitable for measuring power spectra for passband signals, $v_o(t)$, which have a high carrier frequency and a large frequency bandwidth, and phase-noise analyzers are used to measure close-in phase-noise spectra for baseband signals, $\phi(t)$, which are of relatively small frequency components but require finer frequency resolutions. Then, in Section 4.2, calculation and estimation methods for power spectra of various kinds of common continuous-time signals are introduced. For signals with theoretically defined power spectra such as periodic signals and wide-sense stationary (WSS) stochastic signals, their power spectra can be directly calculated. For signals without theoretically defined power spectra such as aperiodic signals and nonstationary stochastic signals, estimation methods need to be applied by sampling a portion of the original signal. In Section 4.3, discrete approximation will be introduced to reveal the approximation relationship of the power spectrum of the baseband phase-noise modulation signal $\phi(t)$ to the phase-noise sidebands of its phase-modulated passband signal $v_o(t)$. Lastly, a summary of this chapter is drawn in Section 4.4.

4.1 Measurements of Power Spectra of Signals

4.1.1 General Measuring Principle of Power Spectra of Signals

The general model of measuring power spectra of signals applies to both power spectrum analyzer and phase-noise analyzer and is shown in Figure 4.1, where $x(t)$ is an arbitrary continuous-time signal passing through an ideal 1-Hz range bandpass filter centered at the frequency f_1. The resulted continuous-time signal, $x_{1\text{-Hz}}(t)$, is then passed through a squaring device and subsequently being smoothed over a long period of time to achieve the signal's average power at the frequency f_1 as $S_{xx}(j2\pi f)|_{f=f1}$. The physical implication of $S_{xx}(j2\pi f)|_{f=f1}$ coincides with the unit for power spectral densities: V^2/Hz, which can be approximately interpreted as a signal's average power concentrated in a 1-Hz range at a certain frequency. The measuring processes in Figure 4.1 can be abstracted as a single formula as:

$$S_{\text{xx}}\left(j2\pi f\right)=\lim_{T\to\infty}\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}}\left|x_{1\text{-Hz}}(t)\right|^2 dt\,, \tag{4.2}$$

where T is the smoothing time period of the averaging process and, in practical measurement equipments, as long as $1/T$ is much smaller than the frequencies of the measured signal $x(t)$, T can be treated as infinity. The unit of $S_{xx}(j2\pi f)$ can be taken as V^2/Hz or A^2/Hz, depending on the type of the measured signal: voltage or current, and more commonly, $S_{xx}(j2\pi f)$ is affixed with a unit of W/Hz because of the input impedance of measurement equipments.

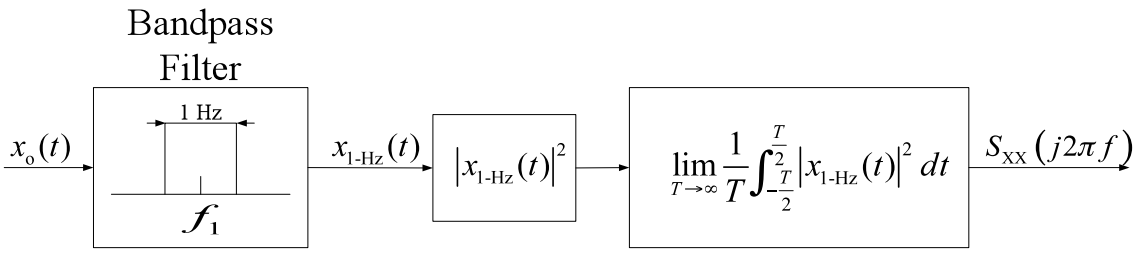


Figure 4.1: General model of measuring power spectra of signals

4.1.2 Measurement of Power Spectra of Passband Signals

Because a practical bandpass filter has a bandwidth which is much greater than 1-Hz range and its center frequency can not be altered easily with the input frequencies, the general model in Figure 4.1 needs to be modified to measure the power spectra of passband signals. Shown in Figure 4.2 is a simplified model of a spectrum analyzer for passband signals. Comparing with the general model in Figure 4.1, a passband spectrum analyzer utilizes a mixer and a swept local oscillator with sweeping frequency, f_{LO} to convert the high-frequency input passband signal to an intermediate-frequency signal, f_{IF}; the intermediate-frequency signal is then fed into a bandpass filter with resolution bandwidth, RBW and its center frequency aligned with the intermediate frequency, f_{IF}, to generate a filtered output signal of $v_{RBW}(t)$; the following squared-law detector and the smoothing filter are similar to the ones in Figure 4.1 with the smoothing filter attaining a video bandwidth of VBW. The output of the smoothing filter represented by $P_{RF}(f)$ is the average power of the signal within the resolution bandwidth, RBW. But the definition of power spectral density can be approximately interpreted as the average power of the signal within a 1-Hz range. Thus, $P_{RF}(f)$ can be converted to a nominal power spectral density, $W_{RF}(f)$, by dividing its RBW as

$$W_{\mathrm{RF}}(f)\ \left(\frac{\mathrm{mW}}{\mathrm{Hz}}\right)=\frac{P_{\mathrm{RF}}(f)\ (\mathrm{mW})}{\mathrm{RBW\ (Hz)}}, \tag{4.3}$$

where $W_{RF}(f)$ is an approximation of the signal's actual power spectral density because its resolution is the same as the spectrum analyzer's RBW instead of the 1-Hz range defined by an actual power spectral density. The only way to improve the accuracy of $W_{RF}(f)$ to an actual power spectral density is to reduce the RBW at the cost of increasing the scanning time of the power spectrum analyzer.

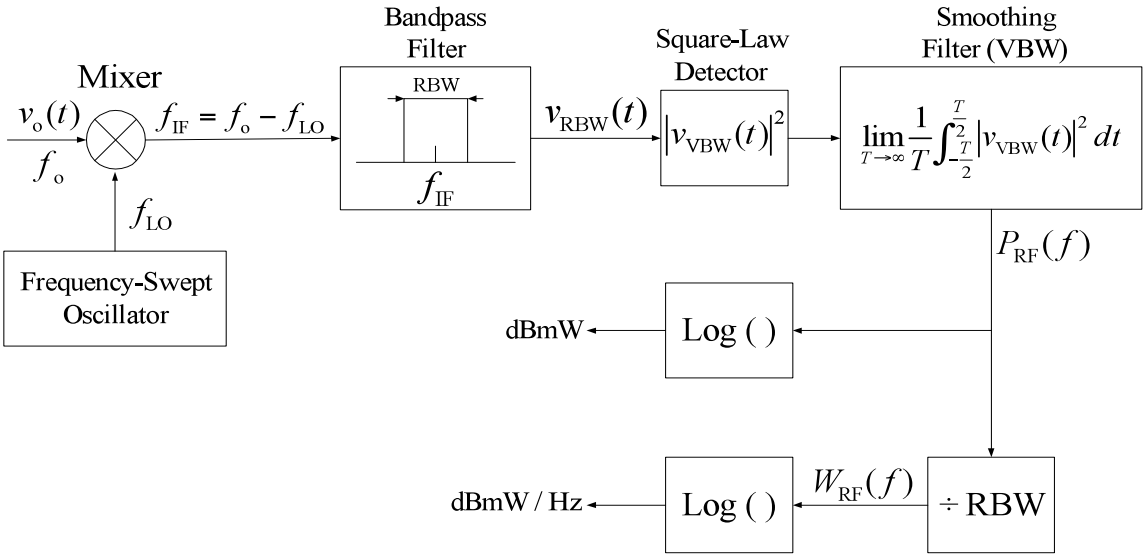


Figure 4.2: Simplified model of power spectrum analyzers for passband signals

Shown in Figure 4.3 are simulated power spectrum, $P_{RF}(f)$, and power spectral density, $W_{RF}(f)$, of a 10-MHz sinusoidal oscillator output signal measured by a passband spectrum analyzer shown in Figure 4.2 with a RWB = 1 kHz. Figure 4.3(a) shows the reading of $P_{RF}(f)$, which represents the signal's average power within the RBW at a certain frequency. $W_{RF}(f)$ shown in Figure 4.3(b) is the scaled version of $P_{RF}(f)$ by (4.3) to

represent the power spectral density of average power within a 1-Hz range at a certain frequency.

To better show noise sidebands of passband signals, more commonly, dB scales are used in spectrum analyzers by comparing the signal's power at a certain frequency with a standard unit power. $P_{RF}(f)$ can be converted to a dB-scale reading with a unit of dBmW by dividing its power at a certain frequency within the RBW by a unit power of 1 mW as

$$10\log_{10}\left(\frac{P_{\mathrm{RF}}(f)\ (\mathrm{mW})}{1\ \mathrm{mW}}\right)\ \ (\mathrm{dBmW}). \tag{4.4}$$

Figure 4.4(a) shows the dB-scale reading of $P_{RF}(f)$ in Figure 4.3(a), where noise sidebands are better demonstrated than in Figure 4.3(a). Similarly, $W_{RF}(f)$ can be converted to a dB-scale reading with a unit of dBmW/Hz by dividing its power spectral density at a certain frequency by a unit power spectral density of 1 mW/Hz as

$$10\log_{10}\left(\frac{W_{\mathrm{RF}}(f)\ \left(\frac{\mathrm{mW}}{\mathrm{Hz}}\right)}{1\ \left(\frac{\mathrm{mW}}{\mathrm{Hz}}\right)}\right)\ \ (\mathrm{dBmW/Hz}). \tag{4.5}$$

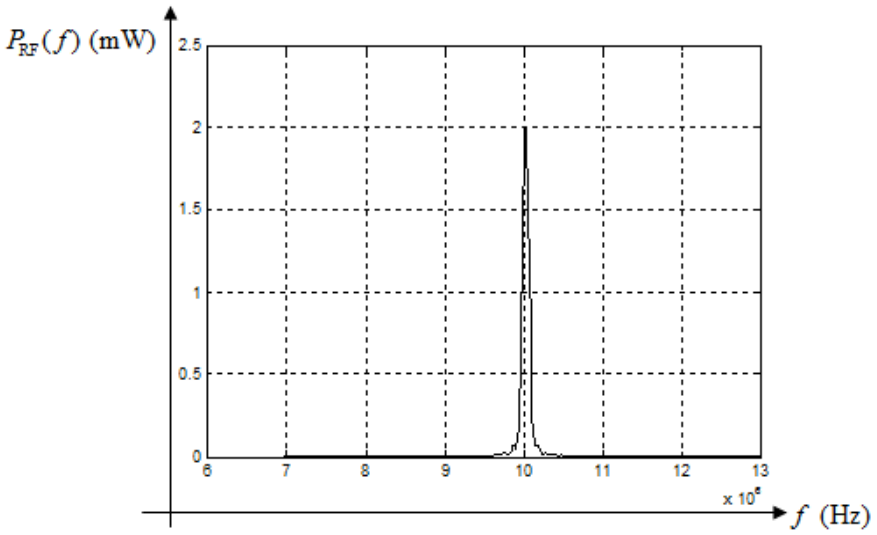
Figure 4.4(b) shows this process of (4.5) by converting the actual power spectral density reading of $W_{RF}(f)$ in Figure 4.3(b) to its dB-scale reading. Because $W_{RF}(f)$ is just an ordinate-scaled version of $P_{RF}(f)$, the dB-scale reading of $W_{RF}(f)$ in Figure 4.4(b) can be obtained from dB-scale reading of $P_{RF}(f)$ in Figure 4.4(a) by subtracting a logarithm of RBW from the latter as

$$
\begin{aligned}
&10\log_{10}\left(W_{\mathrm{RF}}(f)\ \left(\frac{\mathrm{mW}}{\mathrm{Hz}}\right)\Big/1\ \left(\frac{\mathrm{mW}}{\mathrm{Hz}}\right)\right)\ \ (\mathrm{dBmW/Hz})\\
&=10\log_{10}\left(W_{\mathrm{RF}}(f)\right)=10\log_{10}\left(P_{\mathrm{RF}}(f)/RBW\right)\\
&=10\log_{10}\left(P_{\mathrm{RF}}(f)\right)-10\log_{10}\left(RBW\right)\\
&=10\log_{10}\left(\frac{P_{\mathrm{RF}}(f)\ (\mathrm{mW})}{1\ \mathrm{mW}}\right)\ (\mathrm{dBmW})-10\log_{10}\left(RBW\right)
\end{aligned}
\quad . \tag{4.6}
$$

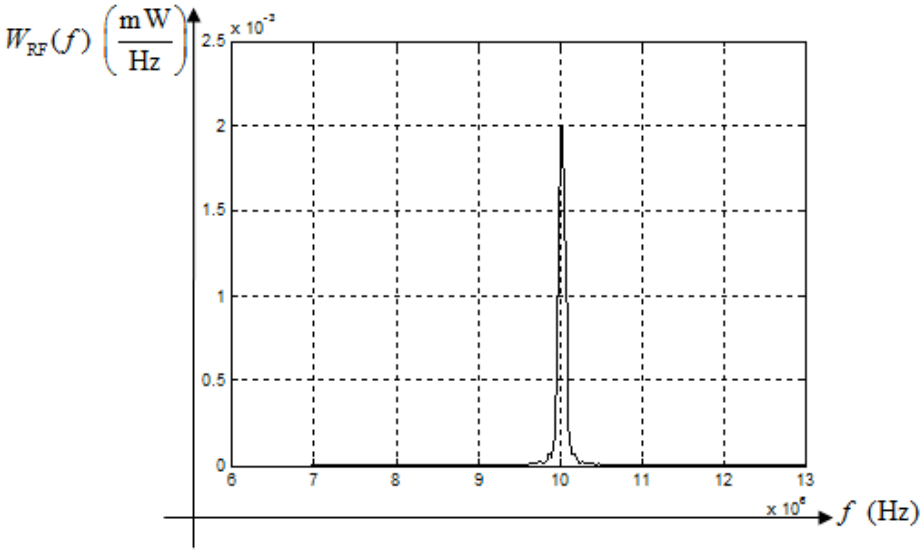
The ratio of noise powers at different frequencies on sidebands of a passband signal can be conveniently obtained by subtracting the readings of noise powers on dB-scale charts of $P_{RF}(f)$ and $W_{RF}(f)$. For example, to compare noise sideband powers at 11 MHz and 12 MHz of the passband signal in Figure 4.4(a), the noise-power ratio at the two frequencies can be directly obtained by subtracting their respective dB-scale readings as

$$
\begin{aligned}
&10\log_{10}\left(\frac{P_{\mathrm{RF}}\left(f_1\right)\big|_{f_1=11\ \mathrm{MHz}}}{P_{\mathrm{RF}}\left(f_2\right)\big|_{f_2=12\ \mathrm{MHz}}}\right)\ \ (\mathrm{dB})\\
&=10\log_{10}\left(\frac{P_{\mathrm{RF}}\left(f_1\right)\big|_{f_1=11\ \mathrm{MHz}}/1\mathrm{mW}}{P_{\mathrm{RF}}\left(f_2\right)\big|_{f_2=12\ \mathrm{MHz}}/1\mathrm{mW}}\right)\\
&=10\log_{10}\left(\frac{P_{\mathrm{RF}}\left(f_1\right)\big|_{f_1=11\ \mathrm{MHz}}}{1\mathrm{mW}}\right)-10\log_{10}\left(\frac{P_{\mathrm{RF}}\left(f_2\right)\big|_{f_2=12\ \mathrm{MHz}}}{1\mathrm{mW}}\right)\\
&=\mathrm{dBmW}\left(f_1\right)\big|_{f_1=11\ \mathrm{MHz}}-\mathrm{dBmW}\left(f_2\right)\big|_{f_2=12\ \mathrm{MHz}}\\
&\approx\left(-30\ \mathrm{dBmW}\right)-\left(-35\ \mathrm{dBmW}\right)\\
&=5\ \mathrm{dB}
\end{aligned}
\quad . \tag{4.7}
$$

Similar principles can be applied to dB-scale charts for $W_{RF}(f)$.

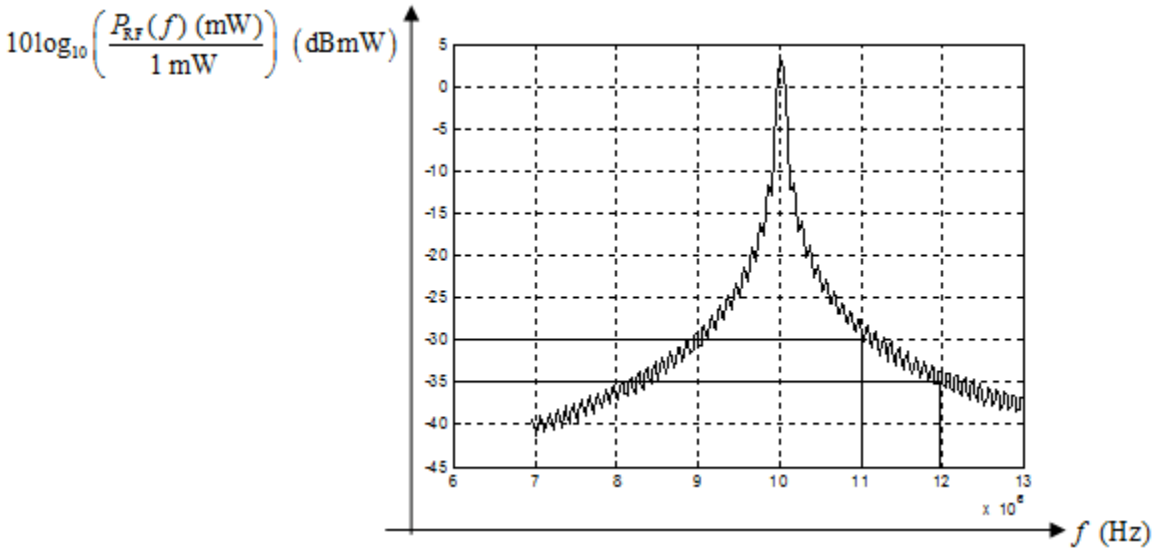


(a)

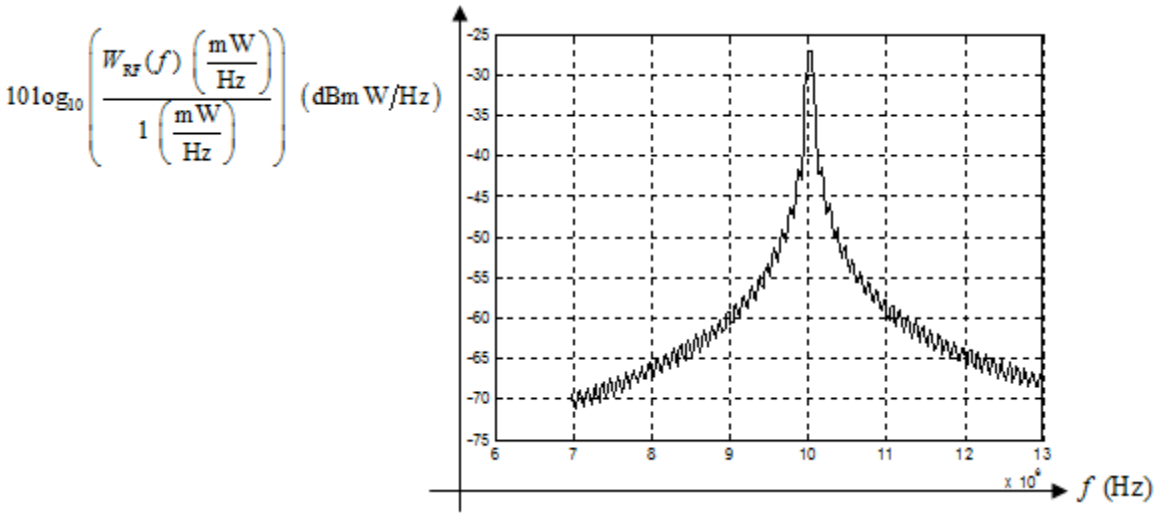


(b)

Figure 4.3: Simulated measured power spectrum and power spectral density of a 10-MHz sinusoidal oscillator signal by a passband spectrum analyzer with RBW = 1 kHz: (a) $P_{RF}(f)$ and (b) $W_{RF}(f)$



(a)



(b)

Figure 4.4 Conversion of actual power readings in Figure 4.3 to relative dB-scale readings: (a) dB-scale reading of power spectrum, $P_{RF}(f)$, in Figure 4.3(a); (b) dB-scale reading of power spectral density, $W_{RF}(f)$, in Figure 4.3(b)

4.1.3 Measurement of Power Spectra of Baseband Signals

Various noise sources exist in a PLL and their final results at the output of the PLL can be assembled as time-domain phase-noise modulation, $\phi(t)$, in (4.1) or

frequency-domain spectrum sidebands illustrated in Figure 4.3 and Figure 4.4. The study of close-in phase-noise sidebands for passband signals is difficult for a spectrum analyzer because its RBW is required to be large to accommodate a widespread frequency range of the passband signal, which results that details of the close-in sidebands have to be ignored. In addition, in order to measure power spectra of input signals with wide power-dynamic ranges without overloading the measuring system, weak sidebands are often hardly able to be displayed noticeably in spectrum analyzers. Because of these major drawbacks for a spectrum analyzer, the frequency-domain power spectrum of the baseband phase-noise modulation signal, $\phi(t)$, of the corresponding passband signal is used to represent the close-in noise sidebands of the passband signal. This approximation has been quite successful in practical observations and has become the underlying principle for a phase-noise analyzer where the power spectrum of a phase-noise modulation signal, $\phi(t)$, will be displayed. Conditions for this approximation to hold will be discussed in Section 4.3 and we are going to use the derived results from this approximation to simulate phase-noise and phase-spur performance of our proposed PLL frequency synthesizer architecture in this thesis.

Because the phase-noise modulation, $\phi(t)$, is a baseband continuous-time real signal with its power spectrum symmetrically even on both sides of the frequency origin, by folding the two-sided power spectrum to the positive-frequency range and doubling the corresponding spectrum components, we can use $W_\phi(f)$ to represent its one-sided power spectrum on the positive frequencies. Shown in Figure 4.5 is a simulated one-sided $W_\phi(f)$ of a phase-noise modulation signal, $\phi(t)$, at the output of a fractional-N frequency synthesizer. It is prominent to note that the power spectrum, $W_\phi(f)$, is composed of a

continuous power spectrum and a discrete power spectrum with individual impulse lines. A well-known polynomial of $1/f$ is often used to approximate the continuous power spectrum [1] as

$$W_\phi(f) \approx \frac{h_3}{f^3} + \frac{h_2}{f^2} + \frac{h_1}{f^1} + h_0 \quad (\text{mW/Hz}), \tag{4.8}$$

where h_v with v = 0, 1, 2 and 3 are coefficients for each of the polynomial terms and with units of $\text{mW}\cdot\text{Hz}^{v-1}$.

The continuous power spectrum and discrete power spectrum of $W_\phi(f)$ in Figure 4.5 are due to different kinds of noise sources in the phase-noise modulation signal $\phi(t)$. Continuous power spectra are caused by continuous-time aperiodic or random noise

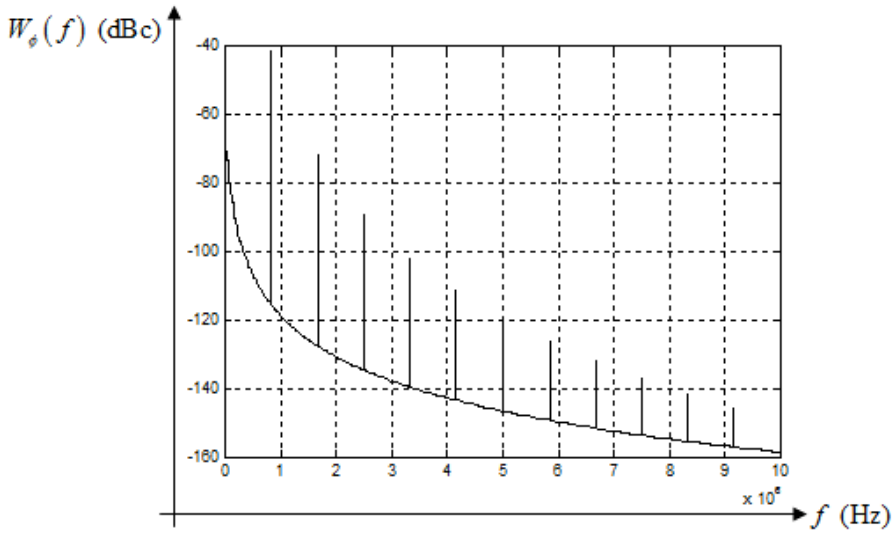


Figure 4.5: Simulated one-sided power spectrum, $W_\phi(f)$, of a phase-noise modulation signal, $\phi(t)$, at the output of a fractional-N frequency synthesizer

sources in a PLL, but discrete power spectra are caused by continuous-time periodic noise sources in the PLL. The different kinds of noise sources in a PLL pass though the PLL system and form a mixed signal at the output of the PLL, which, in the baseband

form, is the phase-noise modulation signal $\phi(t)$. Because of the distinct natures of a continuous power spectrum and a discrete power spectrum, measuring algorithms for a phase-noise analyzer need to be adjusted correspondingly to display the correct power for each kind of the spectra. Shown in Figure 4.6 is a simplified block diagram of a phase-noise analyzer. The phase demodulator is to demodulate the baseband phase-noise modulation signal $\phi(t)$ from its passband carrier signal, $v_o(t)$. (Detailed operations of a phase demodulator can be referred to [1].) The following spectrum analyzer is a baseband spectrum analyzer which is similar to the passband spectrum analyzer in Figure 4.2 but with a much finer RBW to accommodate the relatively narrower frequency range of a baseband signal. The final logarithmic converter is to display the measured baseband phase-noise power spectrum, $W_\phi(f)$, approximately as the one-sided noise sideband of the modulated passband signal $v_o(t)$ by comparing powers at offset frequencies to the carrier-frequency power in a unit of dBc/Hz or dBc, which will be discussed further in Section 4.3.

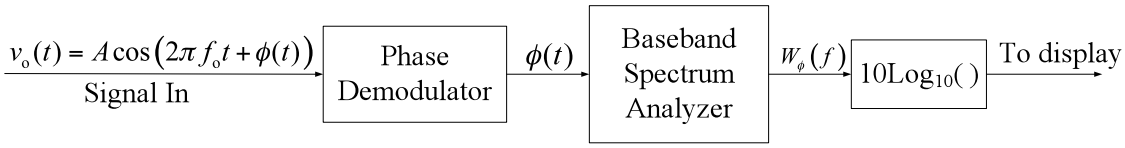


Figure 4.6: Simplified block diagram of a phase-noise analyzer

Measurements of continuous power spectra and discrete power spectra are carried out in the baseband spectrum analyzer for a phase-noise analyzer. The operation for a baseband spectrum analyzer is similar to that for a passband spectrum analyzer, which consists of the major processes of mixing, bandpass filtering, squaring and smoothing as

seen in Figure 4.2, where the mixing and bandpass filtering have been illustrated in their frequency-domain representations and the squaring and smoothing have been illustrated in their time-domain representations instead. But the overall power measuring process in a spectrum analyzer for a one-sided power spectrum $W_\phi(f)$ can also be concluded into a concise frequency-domain equation as:

$$P_\phi\left(f_m\right)\ (\text{mW}) = \int_0^\infty W_\phi\left(f\right)\left|Y\left(f, f_m\right)\right|^2 df\,, \tag{4.9}$$

where $W_\phi(f)$ can assume either a continuous power spectrum or a discrete power spectrum; $Y(f, f_m)$ is the frequency response of the bandpass filter in the spectrum analyzer with its bandwidth centered at a frequency of f_m; the operations of squaring the magnitude of $Y(f, f_m)$ and integration over the frequency range from 0 to $+\infty$ represent the process of passing the measured signal's power through the spectrum analyzer; and the resulted $P_\phi(f_m)$ (mW) is the estimated average power of the measured signal within the RBW of the bandpass filter at the center frequency, f_m.

To measure the continuous power spectrum, $W_\phi(f)$ is denoted as $W_{\phi\_c}(f)$, where the subscript "c" indicates "continuous." Since a spectrum analyzer is to measure a signal's average power within its RBW which is comparatively smaller than the measured signal's frequency range, in the vicinity of f_m, both $Y(f, f_m)$ and $W_{\phi\_c}(f)$ can be approximated as constants by their center values as $Y(f_m, f_m)$ and $W_{\phi\_c}(f_m)$ respectively. And (4.9) can be rewritten as:

$$\begin{aligned} P_{\phi\_\mathrm{c}}\left(f_m\right)\ (\text{mW}) &= \int_0^\infty W_{\phi\_\mathrm{c}}\left(f\right)\left|Y\left(f, f_m\right)\right|^2 df \\ &\approx W_{\phi\_\mathrm{c}}\left(f_m\right)\int_0^\infty \left|Y\left(f, f_m\right)\right|^2 df \\ &= W_{\phi\_\mathrm{c}}\left(f_m\right)\left|Y\left(f_m, f_m\right)\right|^2 B_N\left(f_m\right) \end{aligned}\,, \tag{4.10}$$

where $P_{\phi\_c}(f_m)$ is the spectrum analyzer's output measurement of the input signal's average power within the RBW; the integration of $|Y(f, f_m)|^2$ can be approximated by the product of its center value at the frequency f_m and the $B_N(f_m)$ which is the bandpass filter's effective bandwidth when approximating the bandpass filter as an ideal rectangular frequency response, and, in many cases, $B_N(f_m)$ can be approximately equal to the bandpass filter's RBW. Similar to (4.3), (4.10) indicates that the signal's nominal power spectral density at the frequency f_m can be obtained as

$$W_{\phi\_c}(f_m)\ (\mathrm{mW/Hz}) = \frac{P_{\phi\_c}\ (\mathrm{mW}) \Big/ \left|Y(f_m, f_m)\right|^2}{B_N(f_m)\ (\mathrm{Hz})}, \tag{4.11}$$

where $P_{\phi\_c}(f_m)$ is the spectrum analyzer's measurement; $|Y(f_m\ ,\ f_m)|^2$ and $B_N(f_m)$ are already-known specifications for the spectrum analyzer.

The power measuring processes of (4.10) and (4.11) for a spectrum analyzer have been idealized and simplified such as replacing $Y(f, f_m)$ and $W_{\phi\_c}(f)$ with their respective center values and approximating the integration of $|Y(f, f_m)|^2$ by a product of $|Y(f_m, f_m)|^2$ and $B_N(f_m)$. Measurement calibrations for an actual spectrum analyzer will take into account the actual bandpass filter's characteristic frequency response, $Y(f, f_m)$ and the signal's nonflat power spectral density, $W_{\phi\_c}(f)$. The operations of such calibrations are somewhat complicated by the underlying principles remain the same.

To measure the discrete power spectrum, $W_{\phi}(f)$ is denoted as $W_{\phi\_d}(f)$, where the subscript "d" indicates "discrete." A discrete power spectrum consists of a series of infinite-height, zero-width, finite area, impulses located at the harmonics of its fundamental frequencies f_0 with its single-sided power spectrum represented as

$$W_{\phi\_\mathrm{d}}(f) = |a_0|^2 + \sum_{k=1}^{+\infty} 2|a_k|^2\, \delta(f - k \cdot f_0), \tag{4.12}$$

where a_k is the fourier series coefficients for the continuous-time periodic components in the baseband phase-noise modulation signal $\phi(t)$; and the resulted spurs are located at $k f_0$ for $k = 1, 2, \ldots, +\infty$ with the dc term, $|a_0|^2$, being ignored in our future discussion for the approximation of the baseband power spectrum $W_\phi(f)$ to the one-sided noise sideband of the passband signal $v_o(t)$ only holds for offset frequencies sufficiently away from the carrier frequency as explained in Section 4.3. The bandpass filter in the spectrum analyzer continues to sweep the frequency range of the measured signal and its center frequency f_m automatically aligns closely with each of the impulses in (4.12). For a single impulse at $k_1 \cdot f_0$, $k_1 \in Z^+$, its power spectrum can be written as

$$W'_{\phi\_\mathrm{d}}(f) = 2\left|a_{k_1}\right|^2 \delta(f - k_1 \cdot f_0), \tag{4.13}$$

and, similar to (4.9), its average power within the RBW of the bandpass filter can be calculated as

$$\begin{aligned} P_{\phi\_\mathrm{d}}(k_1 f_0)\ (\mathrm{mW}) &= \int_0^\infty W'_{\phi\_\mathrm{d}}(f)\left|Y(f, f_m)\right|^2 df \\ &= \int_0^\infty 2\left|a_{k_1}\right|^2 \delta(f - k_1 f_0)\left|Y(f, f_m)\right|^2 df \\ &\approx 2\left|a_{k_1}\right|^2 \left|Y(k_1 f_0, k_1 f_0)\right|^2 \int_0^\infty \delta(f - k_1 f_0) df \\ &= 2\left|a_{k_1}\right|^2 \left|Y(k_1 f_0, k_1 f_0)\right|^2 \end{aligned}, \tag{4.14}$$

where $f_m \approx k_1 \cdot f_0$ for the impulse being captured in the bandwidth of the bandpass filter. Therefore its power (the underlying area of the impulse) can be estimated from (4.14) as

$$2\left|a_{k_1}\right|^2\ (\mathrm{mW}) = \frac{P_{\phi\_\mathrm{d}}(k_1 f_0)\ (\mathrm{mW})}{\left|Y(k_1 f_0, k_1 f_0)\right|^2}. \tag{4.15}$$

Equation (4.15) indicates that the power measuring process for discrete spectral spurs is independent of the bandpass filter's effective bandwidth, $B_N(f_m)$, which is substantially different from the power measuring process of (4.11) for continuous spectral

components. It will be seriously wrong to apply the distinct power measuring processes of (4.11) and (4.15) for each kind of power spectral densities to the other one.

4.2 Calculations and Estimations of Power Spectra of Signals

At the beginning of this chapter, it mentioned that not every type of signal has its power spectrum well defined in theory. But we do see the measurement results from a spectrum analyzer or a phase-noise analyzer in the last section. To explain this paradox, we need to understand that spectrum analyzers and phase-noise analyzers only mechanically scan the input signal's average power within the RBW of its bandpass filter and the measurement results of the average powers are displayed in alignment with the center frequencies of the sweeping bandpass filter and named as "power spectrum", regardless of the type of the input signal. However, for computer simulation of a signal's power spectrum, there are no physically-existed sweeping bandpass filters available and the signal's power spectrum must be either calculated or estimated mathematically. In this section, we are going to introduce the power spectral calculation and estimation methods for signals with and without formally-defined spectral densities. And the derived results will be applied to Matlab simulations of phase-noise and phase-spur performances of the proposed PLL frequency synthesizer architecture in this thesis.

4.2.1 Calculation of Power Spectra of Continuous-Time Periodic Signals

For a continuous-time periodic signal $\phi_{\text{period}}(t)$ with Fourier coefficients a_k, $k \square Z$, its power spectrum consists of a series of infinite-height, zero-width, finite-area, impulses located at the harmonics of its fundamental frequency:

$$S_{\phi\_\text{periodic}}(f) = \sum_{k=-\infty}^{+\infty} |a_k|^2 \, \delta(f - k \cdot f_0), \quad \text{for} f_0 = \frac{1}{T}, \tag{4.16}$$

where T is the fundamental period of the periodic signal.

4.2.2 Calculation of Power Spectra of Continuous-Time Wide-Sense Stationary (WSS) Stochastic Signals

For a continuous-time wide-sense stationary (WSS) stochastic signal $\phi_{\text{wss}}(t)$, its autocorrelation function is defined as

$$R_\phi(\tau) = E\left[\phi_{\text{WSS}}(t+\tau) \cdot \phi_{\text{WSS}}(t)\right], \tag{4.17}$$

where τ is the difference of two time instants of the WSS stochastic signal. And its power spectrum is the Fourier transform of the autocorrelation function $R_\phi(\tau)$ as

$$S_{\phi\_\text{wss}}(f) = \int_{-\infty}^{+\infty} R_\phi(\tau) e^{-j2\pi f\tau} d\tau, \tag{4.18}$$

where the power spectrum, $S_{\phi\_\text{wss}}(t)$, can be proved to be always real and positive [13].

4.2.3 Estimation of Power Spectra of Continuous-Time Aperiodic Signals

A continuous-time aperiodic signal $\phi_{\text{aperiodic}}(t)$ does not have a theoretically-defined power spectrum [13], but with an energy spectrum defined as the magnitude square of its Fourier transform as

$$E_{\text{aperiodic}}(f) = \left|\Phi_{\text{aperiodic}}(f)\right|^2, \tag{4.19}$$

where $\Phi_{\text{aperiodic}}(f)$ is the Fourier transform of the aperiodic signal $\phi_{\text{aperiodic}}(t)$. However, a presumed power spectrum for $\phi_{\text{aperiodic}}(t)$ can be derived from the RMS (root mean squared) power of its truncated sample with sufficiently long period. A truncated sample of the original aperiodic signal in the time period from $-T$ to $+T$ is denoted as

$$\phi'_{\text{aperiodic}}(t) = \phi_{\text{aperiodic}}(t) \cdot \left[u(t+T) - u(t-T) \right], \tag{4.20}$$

where $u(t)$ is a unit step function. And the RMS power of $\phi'_{\text{aperiodic}}(t)$ can be used to approximate the RMS power of $\phi_{\text{aperiodic}}(t)$ as

$$\begin{aligned} P_{RMS\_\phi'} &= \frac{1}{2T}\int_{-T}^{+T} \left|\phi'_{\text{aperiodic}}(t)\right|^2 dt \approx P_{RMS\_\phi} = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{+T} \left|\phi_{\text{aperiodic}}(t)\right|^2 dt \\ &= \frac{1}{2T}\int_{-\infty}^{+\infty} \left|\Phi'_{\text{aperiodic}}(f)\right|^2 df \\ &= \int_{-\infty}^{+\infty} \frac{\left|\Phi'_{\text{aperiodic}}(f)\right|^2}{2T} df \end{aligned}, \tag{4.21}$$

where $\Phi'_{\text{aperiodic}}(f)$ is the Fourier transform of $\phi'_{\text{aperiodic}}(t)$; the equation in the 2nd row holds for Parseval's relation; and the last equation implies that the power spectrum of the aperiodic signal $\phi_{\text{aperiodic}}(t)$ can be estimated from $\Phi'_{\text{aperiodic}}(f)$ as

$$S_{\phi\_\text{aperiodic}}(f) = \frac{\left|\Phi'_{\text{aperiodic}}(f)\right|^2}{2T}. \tag{4.22}$$

4.2.4 Estimation of Power Spectra of Continuous-Time Nonstationary Stochastic Signals

For a continuous-time nonstationary stochastic signal $\phi_{\text{nonstat}}(t)$, its power spectrum is not formally defined because autocorrelation functions do not exist for nonstationry stochastic signals. Nonetheless, similar to (4.22), its power spectrum can be estimated through the energy spectrum of a truncated period of a sample of the stochastic signal as

$$S_{\phi\_\text{nonstat}}(f) = \frac{1}{2T}\left|\int_{-T}^{+T} \phi_{\text{nonstat}}(t) e^{-j2\pi f t} dt\right|^2, \tag{4.23}$$

where the magnitude square of the integral is the energy spectrum of the truncated period of the stochastic signal $\phi_{\text{nonstat}}(t)$; and T is the truncation period. This estimation process is called "periodogram" in signal estimation theories [13].

4.2.5 Physical Units for Calculated and Estimated Power Spectra

Physical units can be attached to theoretically calculated and estimated power spectra for practical analyses. To emulate a physical measurement environment for computer simulations, the above mentioned theoretical power spectra can attach a unit of mW/Hz; for circuit analyses, V^2/Hz or A^2/Hz can be used; and for theoretical analyses, rad^2/Hz is popular.

4.3 Discrete Approximation

In previous sections, it has been mentioned that the power spectrum of the baseband phase-noise modulation signal $\phi(t)$ in (4.1) can be used to represent the noise sidebands of its corresponding passband signal $v_o(t)$. This process is called discrete approximation [1], [14]. And its working principle can be demonstrated from the power spectrum of a phase-modulated passband signal with its phase being modulated by a single-tone sinusoid:

$$\begin{aligned} v_o(t) &= A\cos\left(2\pi f_o t + \phi(t)\right) \\ &= A\cos\left(2\pi f_o t\right)\cos\left(\phi(t)\right) - A\sin\left(2\pi f_o t\right)\sin\left(\phi(t)\right) \\ &\approx A\cos\left(2\pi f_o t\right) - A\sin\left(2\pi f_o t\right)\phi(t) \\ &= A\cos\left(2\pi f_o t\right) - A\sin\left(2\pi f_o t\right)\cdot m\sin\left(2\pi f_m t\right) \\ &= A\cos\left(2\pi f_o t\right) + \frac{A\cdot m}{2}\left[\cos\left(2\pi\left(f_0 + f_m\right)t\right) - \cos\left(2\pi\left(f_0 - f_m\right)t\right)\right] \end{aligned} \quad , \quad (4.24)$$

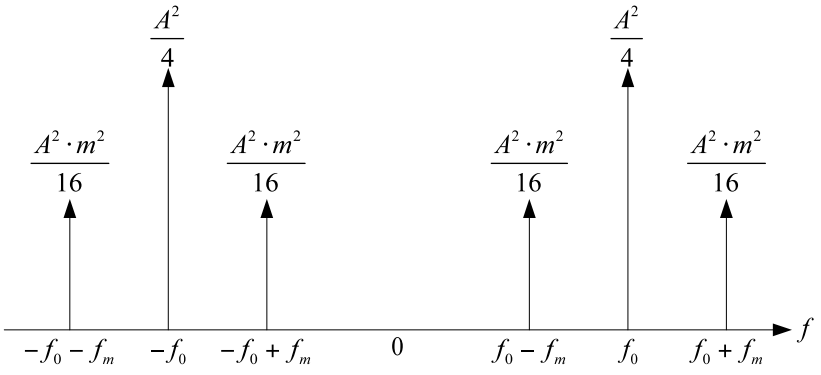
where $\phi(t)$ is the phase modulation for $v_o(t)$; for $\phi(t) << 1$, $\cos(\phi(t)) \approx 1$ and $\sin(\phi(t)) \approx \phi(t)$; and $\phi(t) = m \cdot \sin(2\pi f_o t)$ for single-tone phase modulation. Its double-sided power spectrum denoted by $S_{\text{vo\_double}}(f)$ consists of a series of impulses and is illustrated in Figure 4.7(a) as

$$S_{v_o\_\text{double}}(f) = \frac{A^2}{4}\left[\delta(f - f_0) + \delta(f + f_0)\right] + \frac{A^2 \cdot m^2}{16}\left[\delta(f - f_0 - f_m) + \delta(f + f_0 + f_m) + \delta(f - f_0 + f_m) + \delta(f + f_0 - f_m)\right], \tag{4.25}$$

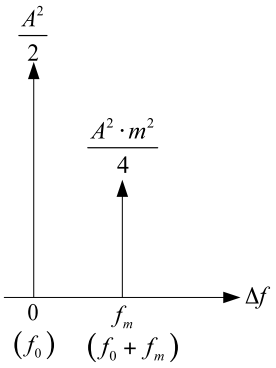
where the power spectral impulses are symmetrical to the origin and to the carrier frequency respectively. By folding Figure 4.7(a) to the right twice with respect to the origin and to the carrier frequency, a single-sided power spectrum $S_{\text{vo\_single}}(f)$ shown in Figure 4.7(b) can be constructed:

$$S_{v_0\_\text{single}}(f) = \frac{A^2}{2}\delta(\Delta f) + \frac{A^2 \cdot m^2}{4}\delta(\Delta f - f_m), \tag{4.26}$$

where Δf denotes the offset frequencies from the carrier frequency f_o; the total power of the carrier component is $A^2/2$; and the total power of the noise-sideband component at the offset frequency f_m is $(A^2 \cdot m^2)/4$.



(a)



(b)

Figure 4.7: Power spectral representations of the passband signal $v_o(t)$ in (4.24): (a) double-sided power spectrum, $S_{vo\_double}(f)$; (b) single-sided power spectrum, $S_{vo\_single}(f)$

Implications of discrete approximation can be revealed from Figure 4.7(b) by the ratio of noise-sideband powers to the carrier power in a single-sided power spectrum:

$$L\left(\Delta f = f_m\right) = \frac{P_{\text{sideband\_single}}\left(\Delta f = f_m\right)}{P_{\text{carrier}}} = \frac{A^2 \cdot m^2/4}{A^2/2} = \frac{m^2}{2}, \tag{4.27}$$

where $L(\Delta f)$ represents the ratio at an offset frequency Δf; $P_{\text{sideband}}(\Delta f)$ is the noise-sideband components' power at the offset frequency Δf; and P_{carrier} is the carrier's power. Observing (4.27), the ratio $L(\Delta f)$ at the offset frequency $\Delta f = f_m$ equals the single-sided power of the baseband phase-noise modulation signal $\phi(t) = m \cdot \sin(2\pi f_o t)$, which is the essential conclusion of discrete approximation.

Discrete approximation indicates that, for a sinusoidal passband signal $v_o(t)$ with its phase being modulated by a baseband signal $\phi(t)$, the ratio of the powers of its noise-sideband components to the carrier's power can be approximately equal to the powers of the baseband phase-noise modulation signal $\phi(t)$ at offset frequencies Δf on the conditions that (1) the baseband phase-noise modulation signal $\phi(t)$ is relatively small and (2) the

offset frequencies are relatively far way from the carrier [1]. But discrete approximation needs to be understood differently for the distinct natures of the kinds of a continuous power spectrum and a discrete power spectrum, and the ratio of the powers is often converted to logarithmic scales to emphasize the weak noise sidebands at farther offset frequencies.

For a continuous power spectrum, its power spectral density at an offset frequency Δf can be interpreted as its average power in a 1-Hz range at that offset frequency. Correspondingly, discrete approximation is interpreted as:

$$\begin{aligned} 10\log_{10}\left(L\left(\Delta f\right)\right)\ \left(\text{dBc/Hz}\right) &= 10\log_{10}\left(\frac{S_{v_o\_\text{single}}\left(\Delta f\right)\ \left(\text{mW/Hz}\right)}{P_{\text{carrier}}\ \left(\text{mW}\right)}\right), \\ &\approx 10\log_{10}\left(W_{\phi\_c}\left(\Delta f\right)\ \right) \end{aligned} \tag{4.28}$$

where $S_{\text{vo\_single}}(\Delta f)$ is the single-sided continuous power spectral density of the passband signal $v_o(t)$ at an offset frequency Δf; and the ratio is approximated to $W_{\phi\_c}(\Delta f)$, the single-sided continuous power spectral density of its baseband phase-noise modulation signal $\phi(t)$ at the same offset frequency Δf, which can be measured as in (4.11) for practical measurement equipments or calculated as in Section 4.2 for computer simulations.

For a discrete power spectrum, its powers are concentrated at individual offset frequencies as infinite-height zero-width impulses. Accordingly, discrete approximation can be written as:

$$\begin{aligned} 10\log_{10}\left(L\left(\Delta f\right)\right)\ \left(\text{dBc}\right) &= 10\log_{10}\left(\frac{P_{\text{spur\_single}}\left(\Delta f\right)\ \left(\text{mW}\right)}{P_{\text{carrier}}\ \left(\text{mW}\right)}\right), \\ &= 10\log_{10}\left(2\left|a_k\right|^2\right) \end{aligned} \tag{4.29}$$

where $P_{\text{spur\_single}}(\Delta f)$ is the single-sided powers of discrete spurs at offset frequencies Δf for the passband signal $v_o(t)$; and the ratio is approximated to the single-sided Fourier

series powers $|a_k|^2$, $k \square Z$ of the underlying baseband phase-noise modulation signal $\phi(t)$, which can be measured as in (4.15) for practical measurement equipments or calculated as in Section 4.2.1 for computer simulations.

4.4 Summary

In this chapter, signals in a PLL have been categorized as passband signals, denoted by $v_o(t)$ and baseband signals, denoted by $\phi(t)$. Measuring principles of a spectrum analyzer have been introduced to display passband signals' average powers within a RBW at an offset frequency, f, as $P_{RF}(f)$ and its nominal power spectral density, $W_{RF}(f)$, can be calculated by dividing the RBW to approximate the signal's actual power spectral density. To better display weak noise sidebands and easily compare the noise power strengths at different offsets, both $P_{RF}(f)$ and $W_{RF}(f)$ have been shown to be commonly converted to dB scales by being compared to their respective physical unities. For baseband signals, phase-noise analyzers have been introduced to demodulate baseband phase noises from their passband signals. And the powers of the baseband noises are displayed in low frequency bands with finer RWBs. Because of the distinct natures of baseband continuous power spectra, denoted by $W_{\phi\_c}(f)$, and baseband discrete power spectra, denoted by $W_{\phi\_d}(f)$, the measuring methods for both spectral types of baseband noises have been discussed separately and their similarities and differences have been emphasized.

To simulate continuous phase-noise and discrete phase-spur sidebands at the output of a PLL system, it is necessary for a computer program to recognize the types of noises in the PLL and choose the correct method to either calculate or estimate the power

spectra of the corresponding noise types. For continuous-time periodic noise signals and continuous-time wide-sense stationary stochastic noise signals, their power spectra are formally defined and can be calculated as introduced in Section 4.2.1 and 4.2.2. For continuous-time aperiodic noise signals and continuous-time nonstationary stochastic noise signals, their power spectra are not theoretically defined, but can be estimated as introduced in Section 4.2.3 and 4.2.4. Phase spurs on the output spectrum of a PLL are caused by continuous-time periodic noise errors in the PLL, the calculation method introduced in Section 4.2.1 will prove to be useful when analyzing the quantization errors in a fractional-N PLL in later chapters.

Discrete approximation serves as key connection between the power spectrum of a baseband modulation signal $\phi(t)$ and the noise sidebands of its corresponding carrier signal $v_o(t)$. The conditions for the discrete approximation to hold were introduced. And interpretations of discrete approximation for continuous power spectra and discrete power spectra were explained individually.

5 New Architecture of a Dual-Mode Cascaded-Loop Frequency Synthesizer

In this chapter, a proposed new architecture of a dual-mode cascaded-loop frequency synthesizer and its synthesis methods will be introduced. The new architecture is designed to solve and tradeoff the advantages and disadvantages of the existing prevailing PLL frequency synthesizers discussed in Chapter 3.

The proposed architecture merges the advantages from the integer-N, fractional-N and dual-loop architectures with its overall configuration resembling the dual-loop architecture but eliminating the application of mixers to reduce phase noises. Its performances such as bandwidths, phase-noise and phase-spur reductions and settling speeds are similar to the $\Delta\sum$ fractional-N architecture, but with a much simpler circuitry for monolithic applications.

5.1 System Architecture

Figure 5.1 shows the architecture of the proposed dual-mode cascaded-loop frequency synthesizer where the PFDs and VCOs are replaced by their linear models in Laplace domain. The first loop is a regular fractional-N PLL but can be converted to an integer-N PLL by disabling the *Q*-modulus accumulator, and the second loop is an integer-N PLL. The two loops are in series and bridged by a variable *M*-modulus divider.

The output frequency can be formulated, as discussed in chapter 3, from the input reference frequency as:

$$f_{out} = \frac{N_1^* N_2}{M} f_{ref} = \frac{\left(N_1 + \frac{A}{Q}\right) N_2}{M} f_{ref}, \tag{5.1}$$

with the 1st-loop equivalent divider modulus

$$N_1^* = N_1 + \frac{A}{Q}. \tag{5.2}$$

The variable M-modulus divider is an essential component in the architecture and provides the following three major functions in the architecture's frequency synthesis:

1. It improves the architecture's frequency resolution by serving as a denominator in (5.1), which makes f_{out} possibly to be a fractional multiple of f_{ref}, but with both constituent loops still working as integer-N PLLs (see Section 5.2.1).
2. For carrier frequencies which require the 1st-loop to be a fractional-N PLL (see Section 5.2.2), the size of the accumulator Q can be substantially minimized. From (3.25), we can see that a smaller Q for a given f_{ref}, will result in factional spurious tones further away from the carrier frequency and make the architecture possible to suppress the spurious tones solely by both the constituent loops' bandwidths.
3. The variable M-modulus divider itself attenuates the powers of phase noises and spurs from the 1st-loop by $10\times\log_{10}(M^2)$ dB.

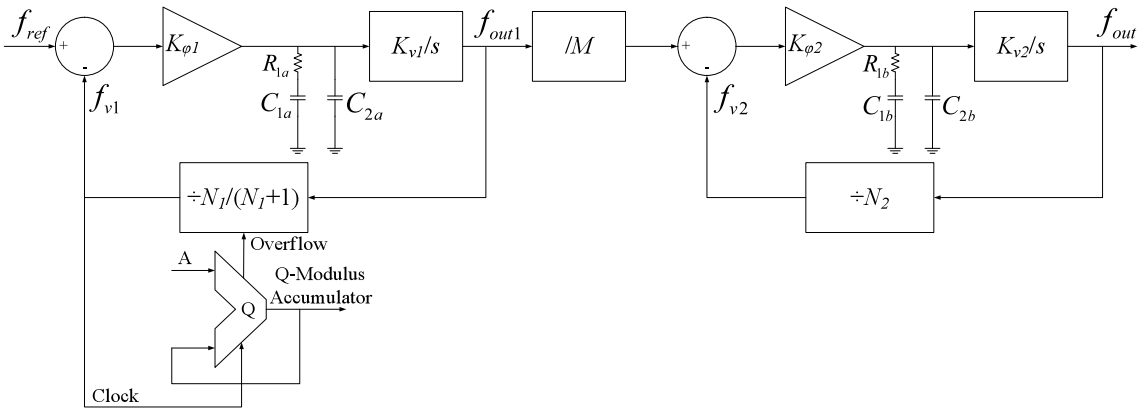


Figure 5.1: Architecture of the proposed dual-mode cascaded-loop frequency synthesizer

5.2 Synthesis Methods

5.2.1 *Nonfractional* Mode (A=0)

For the proposed architecture, carrier frequencies which are not integer multiples of the reference frequency can be synthesized in its *nonfractional* mode. By setting A=0 for the Q-modulus accumulator in the first loop, the fractional-N PLL is converted to an integer-N PLL and (5.1) can be written as:

$$f_{out} = \frac{N_1^* N_2}{M} f_{ref} = \frac{\left(N_1 + \frac{0}{Q}\right) N_2}{M} f_{ref} = \frac{N_1 N_2}{M} f_{ref}, \tag{5.3}$$

\where $(N_1N_2)/M$ constitutes a fractional multiplier for the input reference frequency, whereas both constituent loops are integer-N PLLs. For example, to generate a Bluetooth carrier frequency of 2.479 GHz with 1 MHz channel spacing, a single-loop integer-N frequency synthesizer requires an input reference frequency of 1 MHz and a divider modulus of

$$N_{\text{integer}} = 2.479 \text{ GHz} \,/\, 1 \text{ MHz} = 2479 .$$

But with the proposed architecture, a larger reference frequency of 20 MHz can be used and the cascaded-loop fractional multiplier is

$$N_{\text{cascaded}} = \frac{2.479\ \text{MHz}}{20\ \text{MHz}} = 123.95 = \frac{N_1 N_2}{M} = \frac{296 \times 335}{800},$$

where each constituent loop's divider modulus has been reduced to hundreds. And according to the discussion in Section 3.1.4, this implies the overall bandwidth of the cascaded-loop system can be greatly improved over a single-loop integer-N PLL.

The other great benefit of using *nonfractional* mode is that it does not generate fractional phase spurs when producing the fractional multiples of the input reference frequency because both constituent loops are integer-N PLLs

5.2.2 *Mini-denominator* Mode (0<A<Q)

Although a large number of carrier frequencies in a given application can be synthesized in the *nonfractional* mode, there are frequencies which can not be synthesized precisely solely by the cascade of the two integer-N PLLs. This situation can be proved by a computer search program which searches all possible combinations of N_1, N_2 and M for given sizes of the feedback-loop frequency divider moduli and bridging divider modulus. For the frequencies which can not be synthesized precisely in the *nonfractional* mode, it is necessary that the first loop can be turned into a fractional-N PLL but with a minor accumulator size, Q, to enhance the main fractional effect arising from the bridging M-modulus divider to reach all frequencies. When the first loop is working as a fractional-N PLL, the proposed architecture is working in its *mini-denominator* mode and its synthesis formula has been given in (5.1) with $0<A<Q$.

In *mini-denominator* mode, the accumulator size, Q, can be significantly smaller than that for a single-loop fractional-N PLL, which, according to (3.25), will result in fractional phase spurs located at offset frequencies further away from the carrier frequency. For example, to generate a GSM carrier frequency of 912.2 MHz with 200 kHz channel spacing from a 10 MHz input reference frequency, the synthesis formula for a single-loop fractional-N PLL is

$$\begin{aligned} f_{out} &= \left(N + \frac{A}{Q} \right) f_{ref} \\ &= \left(91 + \frac{11}{50} \right) \times 10 \text{ MHz} = 912.2 \text{ MHz} \end{aligned},$$

with a divider modulus, Q, to be at least 50 to meet the channel spacing requirement. But with the proposed architecture, the same carrier frequency can be synthesized by (5.1) as

$$\begin{aligned} f_{out} &= \frac{N_1^* N_2}{M} f_{ref} = \frac{\left(N_1 + \frac{A}{Q} \right) N_2}{M} f_{ref} \\ &= \frac{\left(380 + \frac{1}{12} \right) \times 240}{1000} \times 10 \text{ MHz} = 912.2 \text{ MHz} \end{aligned},$$

with accumulator size Q=12, which is more than four times smaller than that in the single-loop fractional-N PLL. As with (3.25), in *mini-denominator* mode, the proposed architecture has pushed output fractional phase spurs at offset frequencies at least four times further from the carrier than those generated by a single-loop fractional-N PLL.

For hardware implementations, the accumulator size, Q, is fixed, but the resulted fractional phase spurs can be pushed to even further offset frequencies in *mini-denominator* mode when A and Q in (5.1) contain common factors. For example, to

generate 905.8 MHz in the above GSM communication system, mini-denominator mode provides a synthesis equation:

$$f_{out} = \frac{\left(431 + \frac{4}{12}\right) \times 210}{1000} \times 10 \text{ MHz} = 905.8 \text{ MHz},$$

where $A/Q = 4/12 = 1/3$ with the effective accumulator size of 3 instead of the nominal value of 12. In this case, fractional phase spurs are being pushed more than 16 times further from the carrier than a single-loop fractional-N PLL. It is observed, through computer search, that more that half of the frequencies synthesized in *mini-denominator* mode are of common factors in their A and Q.

Mini-denominator mode indeed is a fractional mechanism. Thus, a relatively large reference frequency can be chosen. From (3.25), a larger reference frequency combined with a smaller Q provides greater frequency margins between the carrier and the fractional phase spurs, which results in opportunities for wider PLL bandwidths and the suppression of phase spurs solely by the cascaded loops' bandwidths and the M-modulus divider in the proposed architecture

5.2.3 Search Results for *Nonfractional* and *Mini-denominator* Modes

For the proposed architecture, a simple computer program can be used to search all possible combinations of N_1, A, Q, N_2 and M to reach every carrier frequency in a given frequency range with specified channel spacing either in its *nonfractional* mode or in its *mini-denominator* mode. For applications in GHz range, in order to maximize the constituent loops' bandwidths and the number of frequency channels which can be synthesized in the *nonfractional* mode, an input reference frequency of a multiple of 10

MHz is suggested such that M can be set to vary around 1000 and 9-bit binary dividers are used for both N_1 and N_2.

As simulation shows, a critical design problem for the proposed architecture is the wide tuning range of the 1st-loop VCO because of the wide variation of N_1. To ease its hardware implementation and reduce its output phase noise, N_1 has been restricted to vary within 10% from its midpoint in the computer search program.

Table 5.1 shows computer search results of the proposed *nonfractional* mode and *mini-denominator* mode for carrier frequencies in GSM, Bluetooth and an arbitrarily specified communication system. In the table, numbers of channels synthesized in each mode are counted respectively.

Table 5.1: Computer search results for channel synthesis in *nonfractional* mode and in *mini-denominator* mode

	P-GSM 900 (Uplink)	Bluetooth	Others (Arbitrary)
Frequency Range	890 – 915 MHz	2.402 – 2.480 GHz	1500 – 1520 MHz
Channel Spacing	200 kHz	1 MHz	100 kHz
Number of Channels	126	79	201
M	800 – 1200	800 – 1200	800 – 1000
N_1 (10% Variation; 9-bit Binary Divider)	360 – 440	270 – 330	360 – 440
N_2 (9-bit Binary Divider)	162 – 302	292 – 491	273 – 371
Accumulator Size (Q)	$Q = 12$	$Q = 8$	$Q = 16$
Input Reference Frequency	10 MHz	20 MHz	10 MHz
Frequency Error	< 1Hz	< 1 Hz	< 100 Hz
Nonfractional Mode [a]	71	52	92
Mini-denominator Mode [a]	55	27	109

[a]Denotes the number of channels in either mode.

5.2.4 Strictly *Nonfractional* Mode

For applications requiring only approximate frequency synthesis, i.e., allowing the synthesized carrier frequencies in the vicinity of the desired values within an error of

a few percentages of the channel spacing, the *nonfractinal* mode can be solely applied to reach every channel frequency. Since both cascaded loops are integer-N PLLs in the *strictly nonfractional* mode, the Q-modulus accumulator in the 1st-loop can be disabled and each of the channel frequencies is synthesized by optimally selecting the integer combinations of N_1, N_2 and M. Shown in Table 5.2 is the computer search results of applying *strictly nonfractional* mode to the same carrier frequencies in applications in Table 5.1. The numbers of channels fallen within difference error percentages of the channel spacing are categorized in the table, which shows that most of the channels synthesized in *strictly nonfractional* mode are located within 1% error of the given channel spacing.

Table 5.2: Frequency errors for all channel frequencies synthesized in *strictly nonfractional* mode

	P-GSM 900 (Uplink)	Bluetooth	Others (Arbitrary)
Frequency Range	890 – 915 MHz	2.402 – 2.480 GHz	1500 – 1520 MHz
Channel Spacing	200 kHz	1 MHz	100 kHz
Number of Channels	126	79	201
M	1000 – 1200	800 – 1200	800 – 1000
N_1 (10% Variation; 9-bit Binary Divider)	360 – 440	270 – 330	360 – 440
N_2 (9-bit Binary Divider)	203 – 302	292 – 491	273 – 371
Input Reference Frequency	10 MHz	20 MHz	10 MHz
Frequency Error $\leq$ 0.1% [b]	96	73	93
0.1% < Frequency Error $\leq$ 1% [b]	29	6	101
1% < Frequency Error $\leq$ 2% [b]	0	0	2
2% < Frequency Error $\leq$ 3% [b]	1	0	5
Frequency Error > 3% [b]	0	0	0

[b]Represents the number of channels fallen within a specified percentage error of the channel spacing.

5.3 Summary

This chapter proposed the new architecture of a dual-mode cascaded-loop frequency synthesizer and introduced its synthesis methods in the *nonfractional mode* and the *mini-denominator* mode:

- The *nonfractional mode* inherits advantages from both the integer-N architecture and the dual-loop architecture where it realizes fractional multiplication by solely applying the cascaded two integer-N PLLs as numerators and the bridging frequency divider as a denominator.
- The *mini-denominator* mode inherits advantages from both the fractional-N architecture and the dual-loop architecture but with a significantly smaller modulus Q for its accumulator compared to a single-loop fractional-N architecture to result in fractional phase spurs being pushed to further offsets from the carrier and filtered by the cascaded PLLs' own bandwidths without auxiliary circuitry.
- For applications requiring only approximate frequency synthesis, *strictly nonfractional* mode was proposed to generate channel frequencies in the vicinity of desired values within a specified small percentage error with respect to the channel spacing.
- Computer search results for specifications of the proposed architecture to apply to GSM, Bluetooth and an arbitrarily sampled system have been given in tables and the frequency synthesis errors were specified by the respective maximum errors and the number of synthesized channels in each of the error categories.

6 System Analysis

This chapter describes the system analyses of the proposed dual-mode cascaded-loop frequency synthesizer architecture for its loop filter structures, system transfer functions, stability and settling speeds, and phase-noise and phase-spur reductions. A PLL's open-loop and closed-loop bandwidths are tightly related to its loop filter structure. In Section 6.1, we first introduce a simple RC loop filter structure which is suitable for the proposed architecture for monolithic applications. Section 6.2 discusses system transfer functions and a stabilization method for the proposed architecture. Design procedures for the optimal selection of synthesis modes and the calculation of loop filter parameters are proposed in Section 6.3. The analysis of discrete phase spurs in mini-denominator mode arising from the 1st-loop fractional-N mechanism will be discussed in Section 6.4. The analysis of continuous phase noises from the 1st-loop VCO which is usually the noisiest component in the proposed architecture is subsequently discussed in Section 6.5. Finally, a conclusion of this chapter is drawn in Section 6.6

6.1 Loop Filters

The loop filter design is crucial for a PLL's bandwidth, stability, settling speed, phase-noise and phase-spur reductions. With the proposed architecture, it is possible to use relatively simple loop filter designs, because, in nonfractional mode, there is no

generated fractional phase spurs and, in mini-denominator mode, the generated fractional phase spurs are pushed to further offsets from the carrier. In addition to the further offset fractional phase spurs, the reference phase spurs caused by the input reference source are also far away from the carrier frequency because of the relatively large reference frequencies used in the proposed architecture, which can be seen in Table 5.1 and Table 5.2. Compared with many other works [2], [3] that apply active high-order loop filters, this work uses a simple passive second-order RC loop filter for both the proposed cascaded loops to achieve a relatively wide bandwidth and substantially reduced phase spurs. A simpler loop filter design with wider bandwidth also favors the sizes of loop filter components for monolithic applications.

As seen in Figure 5.1, identical passive RC loop filter structures have been applied to both loops in the proposed architecture. Assuming both loops employ charge pumps to convert phase differences to current errors, the loop filter transfer function of either loop is a transimpedance and can be written as

$$Z_{a,b}(s) = \frac{1}{\left(C_{1a,b} + C_{2a,b}\right)s} \frac{R_{1a,b}C_{1a,b}s + 1}{1 + R_{1a,b}\dfrac{C_{1a,b}C_{2a,b}}{C_{1a,b} + C_{2a,b}}s}, \tag{6.1}$$

where the subscript “a” and “b” are used to indicate the first and second loop respectively; and $C_{1a,b}$, $C_{2a,b}$, and $R_{1a,b}$ are the loop filter components for the respective loops.

6.2 Stability and Settling

The overall system transfer function of the proposed architecture, denoted by $H(s)$, is the cascade of the transfer functions of its two constituent loops:

$$H(s) = \frac{1}{M} \cdot \frac{N_1^* H_1(s)}{1 + H_1(s)} \cdot \frac{N_2 H_2(s)}{1 + H_2(s)}, \tag{6.2}$$

where the 2nd and 3rd fractional factors represent the closed-loop transfer functions of the two cascaded loops respectively [refer to (3.3) and (3.5)]; and $H_{1,2}(s)$ are their respective open-loop transfer function. As illustrated in Figure 5.1, the open-loop transfer function of the 1st loop, $H_1(s)$ can be written as

$$\begin{aligned} H_1(s) &= K_{\phi 1} Z_a(s) \frac{K_{v1}}{s} \frac{1}{N_1^*} \\ &= \frac{K_{\phi 1} K_{v1}}{N_1^* (C_{1a} + C_{2a}) s^2} \frac{R_{1a} C_{1a} s + 1}{1 + R_{1a} \frac{C_{1a} C_{2a}}{C_{1a} + C_{2a}} s}, \end{aligned} \tag{6.3}$$

where $K_{\phi 1} = I_1/(2\pi)$ (A/rad) is the linearized PFD gain [refer to (3.9)] with its charge pump current denoted by I_1; K_{v1} is the VCO gain [rad/(s·V)]; and N_1^* is the ideal fractional divider modulus for the 1st loop. Because of the identical loop filter structures in the proposed two cascaded loops, $H_2(s)$ has the same transfer function as $H_1(s)$ with its variables in (6.3) replaced by the corresponding variables in the 2nd loop and the study of $H_1(s)$ will be equally applied to the study of $H_2(s)$. It can be seen from (6.2) and (6.3) that each of the cascaded loops in the proposed architecture is a third-order type-II PLL and the overall transfer function $H(s)$ represents a sixth-order system.

Because the two cascaded loops are independent of each other, the overall system's stability can be concluded if each of the constituent loops is stable. To study the stability of $H_1(s)$ in (6.3), its zero-pole form is desired:

$$H_1(s) = \frac{A_1}{s^2} \left(\frac{\frac{s}{\omega_{z1}} + 1}{1 + \frac{s}{\omega_{p1}}} \right), \tag{6.4}$$

where the zero, $\omega_{z1,}$ the pole, ω_{p1} and the coefficient A_1 can be parameterized by the physical constants in (6.3) as

$$\begin{aligned}\omega_{z1} &= 1/\left(R_{1a}C_{1a}\right)\\ \omega_{p1} &= \left(C_{1a}+C_{2a}\right)/\left(R_{1a}C_{1a}C_{2a}\right).\\ A_1 &= \frac{K_{\phi 1}K_{v1}}{N_1^*\left(C_{1a}+C_{2a}\right)}\end{aligned} \tag{6.5}$$

One of the common measures for PLL stability is to adjust its phase margin when the open-loop transfer function crosses its unity-gain frequency which is denoted by ωu1 and is often preselected at the beginning of the design. The phase margin can be adjusted by the relative locations of the zero ωz1 and the pole ωp1 to the unity-gain frequency ωu1. A popular practice [10] as shown in Figure 6.1is to place the zero a factor of r1 below ωu1, i.e., ωz1 = ωu1/ r1, and the pole the same factor of r1 above ωu1, i.e., ωp1 = ωu1×r1. With such an arrangement, the first loop's open-loop transfer function (6.4) can be re-parameterized by the unity-gain frequency ω_{u1} and the location factor r_1 as

$$H_1(s) = \frac{A_1}{s^2}\left(\frac{\frac{s}{\omega_{u1}}r_1+1}{1+\frac{s}{\omega_{u1}r_1}}\right), \tag{6.6}$$

and its magnitude and phase margin at the unity-gain frequency, $s = j\cdot\omega_{u1}$, can be calculated as

$$\left|H_1(j\omega_{u1})\right| = 1 = \frac{A_1}{{\omega_{u1}}^2}\sqrt{\frac{1+r_1^2}{1+1/r_1^2}}\,, \tag{6.7}$$

and

$$\begin{aligned}\text{PhaseMargin} &= \angle H_1(j\omega_{u1}) - (-180^\circ) \\ &= -180^\circ + \tan^{-1}(r_1) - \tan^{-1}(1/r_1) - (-180^\circ), \\ &= \tan^{-1}(r_1) - \tan^{-1}(1/r_1)\end{aligned} \tag{6.8}$$

where the phase margin can be controlled by solely adjusting the location factor r_1. And the same procedures can be applied to the stabilization of the 2nd loop.

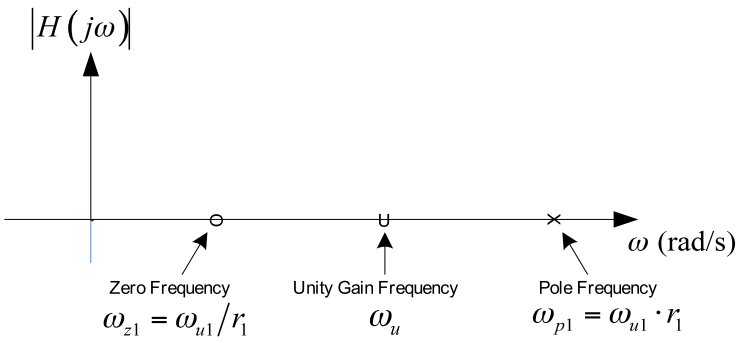
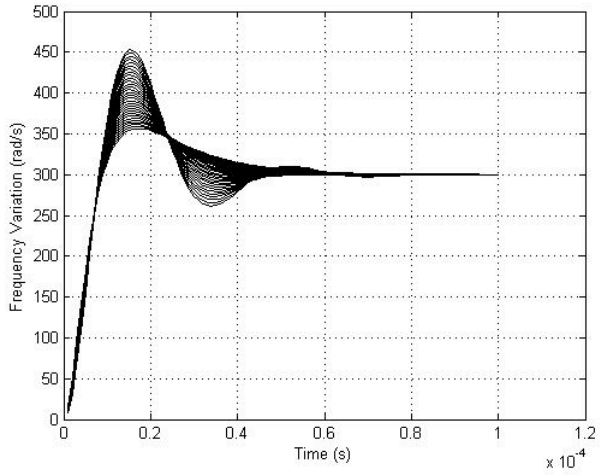


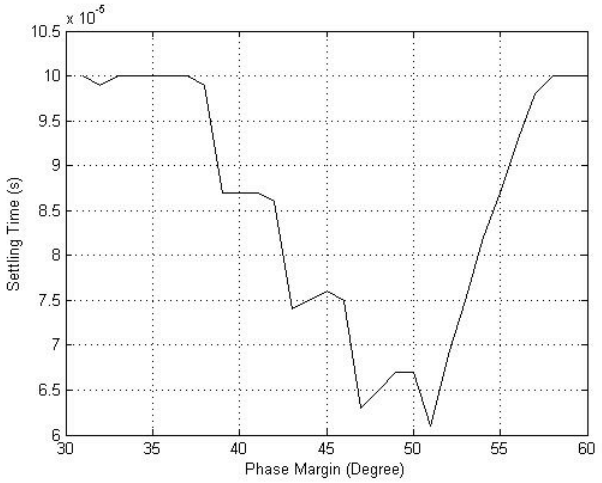
Figure 6.1: Arrangement of zero and pole frequencies relative to the unity gain frequency

As discussed in Chapter 1, a PLL's settling speed is mainly dependent on its closed-loop bandwidth, but a PLL's settling speed can be also affected by its open-loop phase margins when the closed-loop bandwidth is given. Shown in Figure 6.2(a) is a Matlab simulation of linear frequency-switching settling behaviors of a constituent loop (e.g. the 1st-loop) in the proposed architecture for an input frequency unit step of 1 rad/s, where the constituent loop's open-loop phase margins vary from 30° to 60° with the feedback divider modulus $N_1 = 300$ and the preselected open-loop unity-gain frequency $\omega_{u1} = 30$ kHz. The settling times for the PLL's output frequency to settle within 0.01% of its variation are summarized in Figure 6.2(b) for the respective phase margins from 30° to 60°, where it can be found that the fastest settling happens around 52°. But in order to

allow sufficient phase margins for stability, we choose 60° in our design for PLLs in this thesis.



(a)



(b)

Figure 6.2: (a) Linear frequency-switching settling behaviors of a constituent loop in the proposed architecture; (b) Settling times for phase margins varying from 30° to 60°

6.3 System Design Procedures

The discussions of the synthesis methods in Chapter 5, the loop filter structures in Section 6.1 and the allocation of zero-pole positions for stability and settling in Section 6.2 can be summarized as design procedures for the proposed dual-mode cascaded-loop architecture as:

1. Computer search to optimize the selection of N_1, A, Q, M, and N_2 for a given application with a specified frequency range and channel spacing. N_1 is to be designed to vary within a small percentage (e.g., 10%) from its center value to minimize the tuning range of the 1st-loop VCO, which in turn greatly reduces the VCO's output close-in phase noise. Q is to be set at a possibly smallest number to push the phase spurs in the mini-denominator mode to the furthest locations from the carrier while maintaining the accuracy of the fractionally synthesized frequencies within an ignorable error (e.g., frequency error < 0.1% of the channel spacing).

2. Initialize the design by preselecting each component loop's open-loop unity-gain frequency: ω_{u1} and ω_{u2}. And as a quick estimation, each individual loop's closed-loop bandwidth can be roughly approximated as 1.5 times the respective ω_{u1} or ω_{u2}, which allows further quick estimations of the loop's setting speed and phase-noise and phase-spur performances.

3. From (14), calculate r_1 and r_2 for phase margins designed for each loop. Usually, a phase margin of 60° is required for a PLL to account for its loop gain variations.

4. Substitute the preselected $\omega_{u1,2}$ and calculated $r_{1,2}$ to (13) to calculate each individual loop's coefficient $A_{1,2}$.

5. With the known parameters: $\omega_{u1,2}$, $r_{1,2}$, $A_{1,2}$, and specified circuit constants: $K_{\varphi 1,2}$ and $K_{v1,2}$, the linear group (11) can be used to solve for the corresponding loop's loop filter components: $R_{1a,b}$, $C_{1a,b}$ and $C_{2a,b}$.

6.4 Discrete Phase Spur Analysis

In mini-denominator mode, the first loop works as a fractional-N frequency synthesizer with its divider modulus toggling between N_1 and N_1+1 periodically. The toggling is controlled by the overflow from the Q-modulus accumulator shown in Fig. 5.1 where the divider modulus switches from N_1 to N_1+1 every time there is an overflow carry. When the loop is locked, the divider output frequency can assume to follow the toggling immediately, which results in a discontinuous feedback frequency f_{v1} as (refer to Section 3.2):

$$f_{v1} = \left\{ \begin{array}{l} f_{ref}\left(N_1 + \dfrac{A}{Q}\right)\dfrac{1}{N_1}, \text{ if overflow = 0} \\ f_{ref}\left(N_1 + \dfrac{A}{Q}\right)\dfrac{1}{N_1+1}, \text{ if overflow = 1} \end{array} \right\}. \tag{6.9}$$

This discontinuity in f_{v1} causes discontinuous phase errors, $\Delta\varphi$, in the PFD, which can be calculated as:

$$\begin{aligned} \Delta\varphi_j \text{ (rad)} &= 2\pi\left(T_{v1} - T_{ref}\right)\Big/T_{ref} = 2\pi\left(f_{ref} - f_{v1}\right)\Big/f_{v1} \\ &= \frac{1}{N_1^*}\frac{2\pi}{Q}\cdot\left\{ \begin{array}{ll} -A, & \text{if overflow = 0} \\ Q-A, & \text{if overlow = 1} \end{array} \right\} \end{aligned}, \tag{6.10}$$

where T_{ref} is the input reference signal's period; T_{v1} is the period of the feedback frequency f_{v1}; and the subscript index j is to indicate the phase error during a specific reference-clock period. $\Delta\varphi_j$ constitutes a phase-error sequence and is being converted to a continuous-time periodic current-error signal in a CP-PFD as:

$$I_{\text{det\_error}}\ (\text{A}) = K_{\phi 1} \sum_{j} \Delta\varphi_j \ , \tag{6.11}$$

with its power spectrum consisting of a series of infinite impulses spaced at:

$$S_{\text{det\_error}}(f)\ \left(\text{A}^2\right) = \sum_{k=-\infty}^{+\infty} |a_k|^2\, \delta\left(f - k\frac{f_{ref}}{Q}\right) \ , \quad \text{for } k \in Z \text{ and } k \neq 0 \tag{6.12}$$

where a_k is the fourier series coefficients for $I_{\text{det\_error}}$; and its DC component a_0 has been ignored for phase-spur simulation in (6.12) which utilizes discrete approximation (refer to Section 4.3) approximate a PLL's output sinusoidally-modulated phase-noise sidebands to the corresponding baseband power spectra for offset frequencies sufficiently removed from the carrier and the baseband phase-noise deviations are relatively small in time domain [1].

$I_{\text{det\_error}}$ can be regarded as a phase-noise source inserted in between the first-loop's CP-PFD and LPF as shown in Figure 6.3. Its power spectrum $S_{\text{det\_error}}(f)$ is modulated by the VCO and transmitted through the PLL to become phase spurs on both sides of the carrier frequency at the output of the first loop. The phase spurs continue to pass through the M-modulus divider and the second loop and their final powers at the output of the cascaded-loop can be calculated as:

$$
\begin{aligned}
& S_{\text{spur}}\left(\Delta f\right)\ \left(\mathrm{A}^2\right) \\
&= \left|\frac{N_1^*}{K_{\phi 1}}\frac{H_1\left(j2\pi\Delta f\right)}{1+H_1\left(j2\pi\Delta f\right)}\right|^2 \cdot \frac{1}{M^2} \cdot \left|\frac{N_2 H_2\left(j2\pi\Delta f\right)}{1+H_2\left(j2\pi\Delta f\right)}\right|^2, \\
& \cdot S_{\text{det\_error}}\left(\Delta f\right),\ \ \text{for}\ \Delta f \neq 0
\end{aligned}
\tag{6.13}
$$

where Δf is the spurs' offset frequencies from the carrier and the multiplying factors represent the corresponding cascaded components' transfer functions weighing on the original fractional-N phase spurs, $S_{\text{det\_error}}(f)$.

Phase spurs at the output of the proposed cascaded loops are measured by spectrum analyzers which compare the modulated spurs' powers with the carrier power and convert their ratios into logarithmic scales in a unit of dBc as:

$$
\begin{aligned}
L\left(\Delta f\right)\ \left(\mathrm{dBc}\right) &= 10\log_{10}\left(\frac{P_{\text{spur}}\left(\Delta f\right)}{P_{\text{carrier}}}\right) \\
&\approx 10\log_{10}\left(\lim_{f'\to 0}\int_{\Delta f - f'}^{\Delta f + f'} S_{\text{spur}}\left(\Delta f\right)\right)\ \ \text{for}\ \Delta f \neq 0
\end{aligned}
\tag{6.14}
$$

where $P_{\text{spur}}(\Delta f)$ represents the modulated spurs' powers at offset frequencies Δf; P_{carrier} is the carrier's power; and their ratio, by discrete approximation, can be approximated to the spurs' baseband powers at the corresponding offset frequencies Δf, where the integration over a zero-width interval is used to represents the powers of the baseband spurs.

6.5 1st-loop VCO Continuous Phase Noise Analysis

Phase-noise spectrum at the output of the proposed architecture consists of discrete spectral components (phase spurs) and continuous spectral components. The continuous spectral components arise from random phase deviations in the cascaded PLLs, among which the VCOs contribute the most phase noises at the output of the cascaded loops. The amount of phase noises a VCO produces is tightly correlated to its

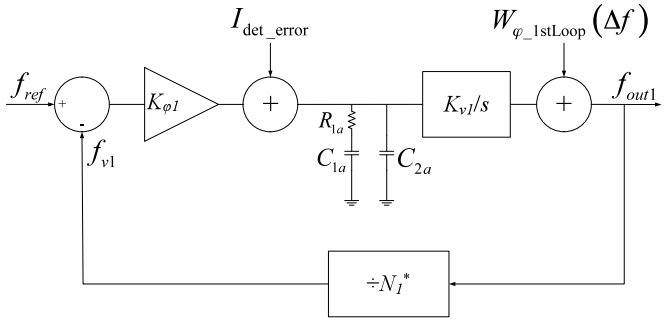


Figure 6.3: Addition of discrete phase spurs and continuous phase noises into the 1st loop of the proposed architecture

tuning range, or equivalently, the VCO gain K_v. The larger the tuning range or the larger the VCO gain K_v, the noisier the resulted VCO is. In Table 5.1 and Table 5.2, variation ranges of the divider moduli, N_1 and N_2, are shown for tuning ranges of the VCOs in each of the loops. For most frequency-synthesis applications, the target frequency range is small, so the tuning range for the second-loop VCO is significantly smaller than that for the first-loop VCO. For example, to apply the proposed architecture to a P-GSM 900 uplink system, the second-loop VCO is required to vary in a 25 MHz range, but the first-loop VCO needs to vary in an 800 MHz range from 3.6 GHz to 4.4 GHz for a reference frequency of 10 MHz despite the restricted 10% change of N_1 from its center value. For low-voltage circuit designs, this wide tuning range of the first-loop VCO results in a large VCO gain, K_{v1}, which in turn produces a large amount of continuous phase noises at its output, but the following derivation will demonstrate that the proposed architecture is capable of efficiently suppressing the phase noises from the first-loop VCO, so relatively simple circuit designs can be applied to the first-loop VCO and the resulted phase noises at the output of the cascaded loops will still meet practical application requirements.

As shown in Figure 6.3, VCO phase noises can be treated as a separate noise source inserted at the output of an ideal VCO and the continuous phase-noise power spectrum can be approximated by a polynomial of $1/\Delta f$ as [1]:

$$W_{\varphi\_1\text{stLoop}}(\Delta f)\ (\text{mW/Hz}) \approx \frac{h_3}{\Delta f^3} + \frac{h_2}{\Delta f^2} + \frac{h_1}{\Delta f} + h_0, \quad \text{for } \Delta f > 0 \tag{6.15}$$

where Δf is the offset frequencies from the carrier and set to be greater than zero to represent a single-sided power spectral density. Similar to (6.13), $W_{\varphi\_1\text{stLoop}}(\Delta f)$ passes through the first loop, the M-modulus divider and the second loop to become part of the total phase noises at the output of the cascaded architecture as:

$$\begin{aligned} &W_{\varphi\_\text{cascaded}}(\Delta f)\ (\text{mW/Hz}) \\ &= \left|1 - \frac{H_1(j2\pi\Delta f)}{1 + H_1(j2\pi\Delta f)}\right|^2 \cdot \frac{1}{M^2} \cdot \left|\frac{N_2 H_2(j2\pi\Delta f)}{1 + H_2(j2\pi\Delta f)}\right|^2, \\ &\cdot W_{\varphi\_1\text{stLoop}}(\Delta f), \ \text{for } \Delta f > 0 \end{aligned} \tag{6.16}$$

where each of the squared norms represents the corresponding loop's transfer function weighing on the $W_{\varphi\_1\text{stLoop}}(\Delta f)$ with the first loop serving as a highpass filter and the second loop as a lowpass filter.

By discrete approximation, spectrum analyzers approximate the continuous baseband $W_{\varphi\_\text{cascaded}}(\Delta f)$ to the ratio of its modulated single-sided power spectral density, $P_{1\text{-Hz\_Sideband}}(\Delta f)$ to the carrier's power, P_{carrier}, at the output of the proposed cascaded loops as:

$$\begin{aligned} &L(\Delta f)\ (\text{dBc/Hz}) \\ &= 10\log_{10}\left(\frac{P_{1\text{-Hz\_Sideband}}(\Delta f)}{P_{\text{carrier}}}\right), \\ &\approx 10\log_{10}\left(W_{\varphi\_\text{cascaded}}(\Delta f)\right), \ \text{for } \Delta f > 0 \end{aligned} \tag{6.17}$$

where the power spectral density $P_{1\text{-Hz\_Sideband}}(\Delta f)$ can also be interpreted as the average power of the phase-noise sideband within a 1-Hz range at an offset frequency Δf; and the ratio is converted to a logarithmic scale with a unit of dBc/Hz for the spectrum analyzer.

6.6 Summary

In this chapter, the important design issues and analysis methods for the proposed dual-mode cascaded-loop frequency synthesizer architecture have been discussed:

- Because there is no phase spurs in *nonfractional* mode and in *mini-denominator* mode, phase spurs are offset to further bands from the carrier, relatively simple RC loop filter structure has be chosen for the proposed architecture for smaller loop filter component sizes favoring monolithic applications.

- For the given loop filter structure, the system transfer function of each of the constituent loops has been derived and transformed into appropriate pole-zero form to study the stability of the loop where the expressions for the phase margin and magnitude at the unity-gain frequency have been calculated explicitly.

- Design procedures for the proposed architecture to be stabilized have been proposed where the optimal selection of synthesis modes for a given application, the preselection of unity-gain frequency and phase margin, and the calculation of loop filter components have been presented.

- Strength and position of discrete phase spurs due to the fractional mechanism in *mini-denominator* mode have been discussed. The internal transfer process of the

discrete phase spurs to appear at the output of the cascaded loops has been represented as filtering effects of the cascaded-loop structure. And the ratio of the powers of the modulated discrete spurs to the carrier power has been approximated to the corresponding baseband powers by discrete approximation.

- Continuous phase noise of the 1st-loop VCO has been modeled by a polynomial. The internal transfer process of the 1st-loop VCO noise to appear at the output of the cascaded loops was represented as filtering effects of the cascaded-loop structure similar to the one for discrete phase spurs. By discrete approximation, the ratio of the power of the modulated continuous phase noise at an offset frequency to the carrier power has been approximated to the baseband power of the continuous phase noise.

7 Simulation of System Performances and Dynamic Behaviors

Simulations of bandwidths, phase spurs, phase noises, and dynamic settling speeds are performed in Matlab. Results are illustrated for a P-GSM 900 uplink system with a frequency range of 890–915 MHz and channel spacing of 200 kHz. Results of applying the proposed architecture to other communication systems can be derived in a similar manner.

From Table 5.1, to synthesize the carrier frequencies in a P-GSM 900 uplink system, an input reference frequency of 10 MHz and an accumulator size $Q = 12$ for the 1st loop fractional frequency divider can be chosen. Both PLL bandwidths are desired to be around 50 kHz for relatively short settling times and sufficient reduction of phase spurs. Following the design procedures in Section 6.3, it is appropriate to set $\omega_{u1} = 30$ kHz and $\omega_{u2} = 35$ kHz, which derives $r_1 = r_2 = 3.7321$ for 60° phase margin on both loops, $A_1 = 9.5204\times10^9$ and $A_2 = 1.2958\times10^{10}$. To account for loop gain variations for stability, loop divider moduli are set at their midpoints: $N_1^* = 400$ and $N_2 = 250$. Charge pump currents and VCO gains are set as: $I_1 = 20$ uA, $K_{v1} = 800$ MHz/V and $I_2 = 200$ uA, $K_{v2} = 25$ MHz/V. From (11), each loop's loop filter components can be calculated as: $C_{1a} = 3.9$ nF, $C_{2a} = 0.3$ nF, $R_{1a} = 5$ kΩ and $C_{1b} = 1.4$ nF, $C_{2b} = 0.1$ nF, $R_{1b} = 11.8$ kΩ.

7.1 Loop Bandwidths

With the above specified and derived parameters for the P-GSM 900 uplink system, Figure 7.1 plots the bandwidths of the 1st, 2nd loops and the proposed architecture. The 1st loop gives a closed-loop bandwidth of 46.92 kHz; the 2nd loop gives a closed-loop bandwidth of 54.74 kHz; and the bandwidth of the overall system is 41.6 kHz.

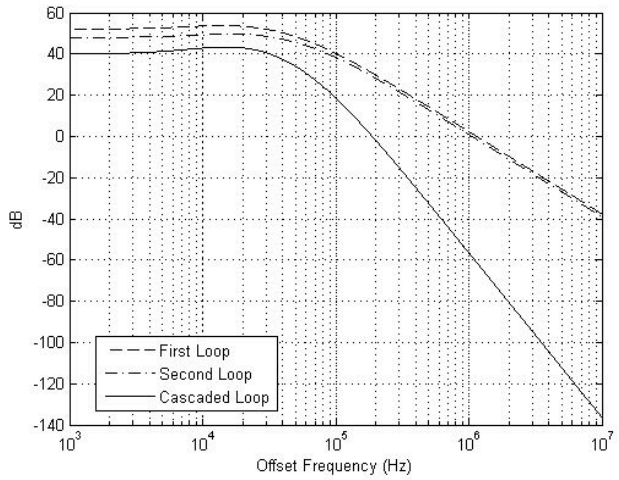
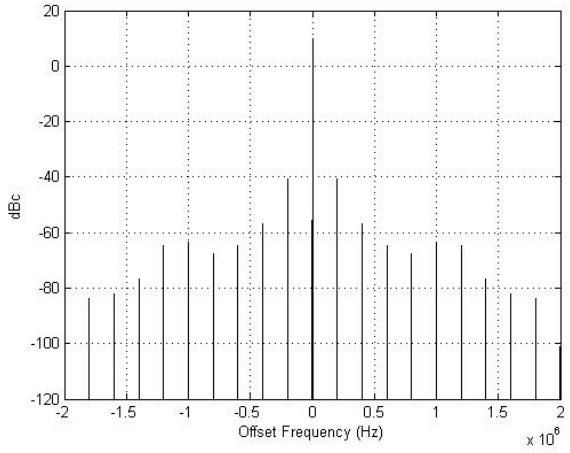


Figure 7.1: Bandwidths of the 1st loop, the 2nd loop, and the cascaded architecture

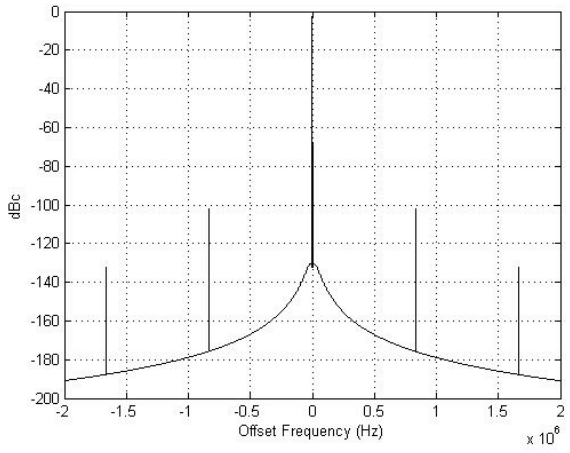
7.2 Filtering of Phase Spurs

In the P-GSM 900 uplink system, one of the frequencies showing the worst-case phase spur scenario is the channel at 912.2MHz, which can be synthesized by a single-loop fractional-N PLL with the phase-spur strength and locations shown in Figure 7.2(a) and the spur offset spacing given by (3.24) as f_{ref}/Q=10 MHz/50=200 kHz. As discussed in Section 5.2.2, for this frequency, Q can be reduced to 12 by applying the *mini-denominator* mode, and the new phase spurs are located at further offset frequencies

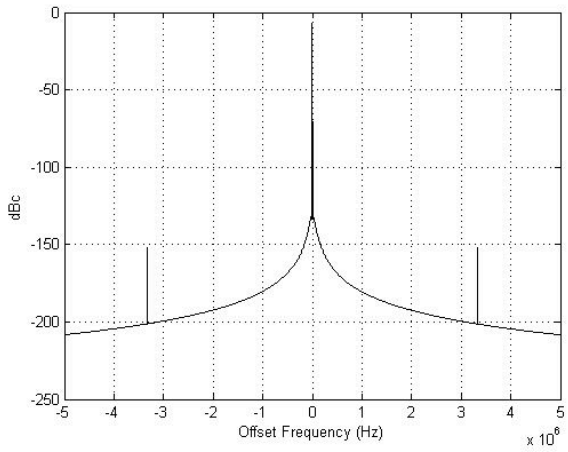
shown in Figure 7.2(b) where offset spacing is integer multiples of f_{ref} /Q=10 MHz/12=833.33 kHz and the strength has been suppressed to below -100 dBc. Another carrier frequency of 905.8 MHz can be synthesized in *mini-denominator* mode with A = 3 and Q = 12 containing a common factor of 3. Figure 7.2(c) shows that phase-spur spacing has spread out to integer multiples of f_{ref} /(Q/A)=10 MHz/4=2.5 MHz and strength is suppressed below -150 dBc.



(a)



(b)



(c)

Figure 7.2: Simulation of phase-spur strength and positions: (a) phase spurs at the output of a single-loop fractional-N PLL for a carrier frequency of 912.2 MHz with Q = 50; (b) phase spurs at the output of the proposed architecture for the same carrier frequency, but with Q = 12; (c) phase spurs at the output of the proposed architecture for a carrier frequency of 905.8 MHz with Q = 12 and A = 4

7.3 Filtering of 1st-loop VCO Phase Noise

Figure 7.3 shows the process of the 1st-loop VCO continuous phase noise passing through the proposed architecture at the outputs of the 1st loop, the bridging divider and the 2nd loop. The original 1st-loop VCO phase noise is modeled by (6.15) as a $1/\Delta f$ polynomial with coefficients extracted from an actual VCO [7]: h_3=7.3×10^6 rad^2·Hz^2, h_2=21.7 rad^2·Hz, h_3=0 rad^2 and h_0= 9.8×10^{-15} rad^2/Hz.

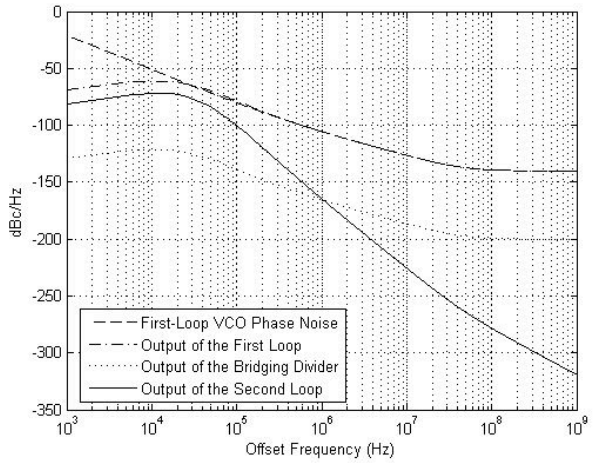


Figure 7.3: 1st-loop VCO close-in phase-noise variations through the cascaded architecture

7.4 Dynamic Settling Behaviors

Figure 7.4 plots the linear settling behaviors of both loops and the architecture for an input angular frequency step of 1 rad/s to have the respective outputs to settle within 0.01% of their frequency increments. The 1st loop settles in 116 μs; 2nd loop in 99 μs and the cascaded loops in 128 μs.

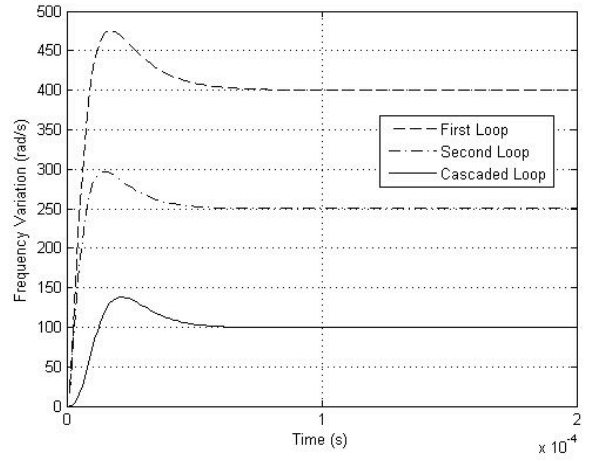


Figure 7.4: Linear frequency-switching settling behaviors of the 1st loop, the 2nd loop, and the cascaded system

7.5 Summary

This chapter simulated the system performances and dynamic behaviors of the proposed architecture for its application toward a P-GSM 900 uplink system. For given design parameters, the bandwidth of the cascaded-loops is achieved to be around 50 kHz, which results in a settling time of 128 μs for 0.01% accuracy. Discrete phase spurs have been suppressed below -100 dBc for the worst-case scenarios and the continuous phase noise from the noisiest 1st-loop VCO has been reduced below -100 dBc/Hz above offset frequencies at 100 kHz. All of the simulated specifications meet GSM standards.

8 Conclusion

This thesis presented the architecture, design procedures and system-performance simulations for the proposed dual mode cascaded-loop frequency synthesizer. The proposed architecture inherits the advantages from both the integer-N and the fractional-N frequency synthesizers, which is that, for a given application with a specified frequency range and channel spacing, each channel frequency can be optimally chosen to be synthesized either in its nonfractional mode or in its mini-denominator mode. In the nonfractional mode, synthesized output frequency can be a fractional multiple of the input reference frequency but without the generation of fractional phase spurs. And in the mini-denominator mode, phase spurs are to be pushed to further offsets and to be reduced substantially by the cascaded-loop filtering effects. The cascaded-loop architecture as well offers opportunities to optimally tradeoff the loop bandwidths, settling speeds, and phase-noise and phase-spur reductions in between the two loops. Because of the removed or further located phase spurs in either of its working modes, the proposed architecture can choose to apply simple passive RC loop filter architectures to achieve a wide bandwidth while maintaining effective suppression of its phase noise and phase spurs for monolithic applications.

8.1 Contributions of the Thesis

The specific contributions of this thesis are:

1. *An analysis of the optimization and tradeoff of the major specifications of a PLL frequency synthesizer, leading to the proposal of a dual-mode cascaded-loop frequency synthesizer architecture*. Because of the inherent contradictory properties of the major specifications of a PLL, bandwidth can be served as the key connection among the specifications. This thesis is to develop a novel architecture to optimize the frequency synthesizer's overall performance by trading off some of the downside properties of the major specifications.

2. *Examination of the pros and cons of existing PLL frequency synthesizer architectures*. Integer-N frequency synthesizers have the advantage of architectural simplicity but suffer from the inherent contradiction of frequency resolution and PLL bandwidth. Fractional-N frequency synthesizers have the advantages of fine resolution and wide bandwidth, but suffer from fractional phase spurs on channel spacing. Dual-loop frequency synthesizers utilize integer-N PLLs to generate fractional multiples, but suffer from large close-in harmonics and increase $1/f$ noises. $\Delta\sum$ fractional-N frequency synthesizers effectively suppress fractional phase spurs in a fractional-N frequency synthesizer, but the cost of the performance superiority is the requirements of large chip area and complexity of the digital $\Delta\sum$ modulator. This thesis proposes an architecture that merges the advantages of the existing architectures and trades off the respective disadvantages.

3. *An analysis of continuous power spectral densities and discrete power spurs of phase-noise signals in PLL frequency synthesizers*. Power measuring principles of a spectrum analyzer for passband signals and a phase-noise analyzer for baseband signals are explained. Differences between baseband continuous power spectra and baseband discrete power spurs are distinguished. Methods of calculation and estimation of power spectra of various continuous-time phase-noise signals in PLL frequency synthesizers are analyzed for computer simulations. Discrete approximation is presented to approximate power spectrum of a baseband phase-noise modulation signal to phase-noise sidebands of its phase-modulated passband signal. Distinction of discrete approximation when applying to continuous phase-noise spectra and discrete phase-noise spurs is clarified.

4. *Development of architecture of a dual-mode cascaded-loop frequency synthesizer*. The architecture is cascaded by two single PLL frequency synthesizers connected through a modulus-variable frequency divider. The 1st loop is a fractional-N PLL and can be converted an integer-N PLL by disabling its accumulator, and the 2nd loop is an integer-N PLL. The architecture inherits advantages from integer-N and fractional-N PLL frequency synthesizers and its overall configuration resembles a dual-loop frequency synthesizer but eliminating the application of mixers to avoid large harmonic spurs and generate less close-in 1/*f* noises.

5. *Development of synthesis modes for both precise and approximate frequency resolution*. *Strictly nonfractional* mode, *nonfractional* mode and *mini-denominator* mode are developed to produce synthesized frequencies as fractional

multiples of input reference frequency where the mode switching is controlled by enabling and disabling an accumulator in the 1st loop of the proposed architecture. *Strictly nonfractional* mode is suitable for approximate frequency synthesis which allows synthesized frequencies away from desired values within a specified small percentage error (e.g. < 3%) of the channel spacing. *Nonfractional* mode and *mini-denominator* mode are suitable for precise frequency resolution where synthesized frequencies are away from desired values less than 0.001% of channel spacing.

6. *Elimination and diffusion of fractional phase spurs*. *Strictly nonfractional* mode and *mini-denominator* mode apply the two loops in the proposed architecture as integer-N PLL frequency synthesizers, which eliminates fractional-N mechanism in the architecture and the resulted fractional phase spurs. *Mini-denominator* mode converts the 1st loop into a fractional-N PLL frequency synthesizer but with a significantly smaller accumulator size, which equivalently reduces denominator size in its synthesis formula and results in diffused fractional phase spurs to further offsets from the carrier. Far offset phase spurs can be filtered by internal loop bandwidths of the cascaded-loop architecture.

7. *Development of a computer search program for selection of optimal synthesis modes for channel frequencies in an application with given frequency ranges and channel spacing*. Because *nonfractional* mode does not generate fractional phase spurs, the computer search program will first start its search for possible configurations of the variable parameters in the proposed architecture to reach

every channel frequency within a specified frequency error. If a specific channel frequency can not be synthesized precisely in *nonfractional* mode, the program automatically switch its searching mode to *mini-denominator* mode to reach the channel frequency with possibly smallest accumulator size for furthest offset fractional phase spurs. Application of the computer search program to channel frequencies in GSM, Bluetooth and an arbitrarily specified GHz band are given in tables with maximum frequency errors.

8. *Development of quick design procedures for stabilization of the proposed dual-mode cascaded-loop frequency synthesizer architecture*. Because of the removed and far offset fractional phase spurs in *nonfractional* and *mini-denominator* modes, passive RC loop filters can be assumed in both cascaded loops to achieve wider bandwidths. Based on loop transfer functions derived from transimpedances of the loop filters, this thesis develops quick design procedures for stabilization of the proposed architecture, which includes optimal selection of synthesis modes for a given application, preselection of unity-gain frequency and phase margin, allocation of zeros and poles in loop transfer functions and synthesis of loop filter components to achieve smaller sizes for monolithic chips.

9. *An analysis of discrete phase spurs arising from fractional-N mechanism in mini-denominator mode*. Discrete phase spurs arise from periodic phase errors in PFD due to fractional-N mechanism. This thesis analyzes the strength and locations of discrete phase spurs on the output spectrum of PLLs by Fourier analysis. The internal transfer process of discrete phase spurs is represented as filtering process

by the proposed cascaded loops. And discrete powers of individual spurs are denoted by ratios compared to carrier power by discrete approximation.

10. *An analysis of continuous phase-noise spectrum in PLL frequency synthesizers*. This thesis models continuous phase-noise spectra by polynomials and analyzes close-in continuous phase noises contributed by the noisiest 1st-loop VCO in the proposed architecture. The internal transfer process of continuous phase noises is represented as filtering process by the proposed cascaded loops. And power of continuous phase-noise spectrum at an offset frequency is denoted by its ratio compared to carrier power by discrete approximation.

8.2 Future Work

In the course of this work, the following topics have been identified as areas of future research:

1. *Development of advanced loop filter structures for reduced capacitor and resistor areas on integrated chips*. The passive RC loop filter structure discussed in Chapter 6 is suitable for the proposed architecture and is integrable for GHz applications. But more advanced loop filter structures can be explored to achieve smaller chip area by reducing the sizes of the integrating capacitors and resistors. One of the possible techniques is dual-path loop filters which consist of an integration path, a lowpass path and a voltage adder for its dual charge pumps and, by scaling the dual charge-pump currents, the dual-path loop filter is equivalent to scaling up integration capacitance by a current scaling factor of the

dual charge pumps. Another possible technique is capacitance multiplier which is a special example of impedance scaling based on current amplifier where the loop filter resistances and capacitances are divided and multiplied respectively by the current gain factor.

2. *Analog compensation for fractional phase errors in mini-denominator mode*. The proposed architecture in Chapter 5 spreads fractional phase spurs to far offsets from the carrier by reducing its accumulator size in the 1st loop of the cascaded loops. A careful study of the output of the accumulator reveals that it is inversely proportional to the fractional phase errors generated in the PFD. This prompts a potential method to suppress fractional phase spurs by compensating the corresponding fractional phase errors in the PFD by converting the digital outputs from the accumulator to an analog signal by a DAC and adding the analog signal inversely back to the PFD to cancel the fractional phase errors.

3. *Delta-sigma (ΔΣ) noise shaping techniques to suppress fractional phase spurs in mini-denominator mode*. As discussed in Section 3.4, fractional phase spurs around a carrier are Fourier power spectra of quantization errors $e_m[n]$ in the feedback fractional frequency divider. This prompts a potential method of eliminating fractional phase spurs by reshaping the power spectra of quantization errors $e_m[n]$ in mini-denominator mode by a ΔΣ modulator with overflows fed to the feedback fractional frequency divider. A ΔΣ modulator randomizes the instantaneous division ratio and pushes phase-noise spectra associated with $e_m[n]$ from low offset frequencies to high offset frequencies. A ΔΣ modulator can be

realized by either analog techniques or digital techniques. An analog $\Delta\sum$ modulator can be possibly implemented by switched-capacitor (SC) or switched-current (SI) techniques and contain basic components such as integrators, operational transconductance amplifiers (OTA), a single-bit quantizer and a clock generator. A digital $\Delta\sum$ modulator can be implanted by an ASIC programmed in the language of Verilog HDL and embedded with other analog components in Cadence environment.

4. *Development of fast-locking techniques*. The passive RC loop filter structure in the proposed architecture can be modified to speed up acquisition process by increasing its loop bandwidth for a short period of time. Potential fast-locking techniques can be: 1) creation of a separate port to charge directly the largest capacitor in the loop filter; and 2) analog switching that bypasses shunt resistors to allow charging of loop filter capacitors directly.

Appendix. A Matlab Codes

A.1 Search for Synthesis Modes

The following Matlab script was used to search the optimal selections of the 1st-loop divider modulus N_1, the 2nd-loop divider modulus N_2, the bridging divider modulus M and the accumulator size Q for the proposed architecture to synthesize every channel frequency in a given application with ignorable errors. The search results have been demonstrated in Table 5.1 and 5.2 for the different synthesis modes respectively.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Computer Search for Optimal Selection of Synthesis Modes          %%
%% Author: Xiongliang Lai                                            %%
%% Date: Mar. 25th 2009                                              %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;  % use format short g
n=1;
status=0;
status_frac=0;
num_integer=0;
num_decimal=0;
num_impossible=0;

Q=12;   % Accumulator Modulus
f_r=10; % Reference Frequency in the unit of MHz
M=800;  % Bridging Divider Modulus

% Target Frequency in MHz
for f_n1n2=890:.2:915
  while M<=1200 & status<1

    for n2=1:512 % 9-bit cascaded D-flip-flops
        n1=round(f_n1n2*M/f_r/n2);
```

```
    % n2*n1/M*f_r = Actual output frequency in MHz; f_n1n2 =
    % target frequency
        if abs(n2*n1/M*f_r-f_n1n2)<1e-6 & 360<=n1 & n1<=440
         factors(n,:)=[f_n1n2, n2,n1,0, Q, M, n2*n1/M*f_r-f_n1n2];
         status=1;
            n=n+1;
        end
    end

    if status==0 & status_frac==0
        % Minimum of N2 = Minimum of f_n1n2 / Maximum of N1
        for n2=1:512
            n1_integer=floor(f_n1n2*M/f_r/n2);
            A=round(Q*(f_n1n2*M/f_r/n2-n1_integer));
        % n2*(n1_integer+A/Q)/M*f_r = Actual Output Frequency in
          MHz;
        % f_n1n2 = target frequency
            if abs(n2*(n1_integer+A/Q)/M*f_r-f_n1n2)<1e-6 &
360<=n1_integer & n1_integer<=440
        % Unit or Error in MHz
                factors(n,:)=[f_n1n2, n2, n1_integer, A, Q, M,
n2*(n1_integer+A/Q)/M*f_r-f_n1n2];
                n=n+1;
                status_frac=1;
            end
        end
    end

    M=M+1;
  end

  if status==1
      num_integer=num_integer+1;
  elseif status_frac==1
      num_decimal=num_decimal+1;
  else
      factors(n,:)=[f_n1n2, 0, 0, 0, 0, 0, 0];
      n=n+1;
      num_impossible=num_impossible+1;
  end

  status=0;
  status_frac=0;
  M=800;
end

% Display
num_impossible
num_integer
num_decimal
```

A.2 Settling Time vs. Phase Margin

The following Matlab script was used to plot a PLL's settling behaviors for different phase margins when its input encounters a unit angular-frequency step. Settling times for 0.01% (or 0.05%) accuracy are tallied against corresponding phase margins. The phase margin for fastest settling speed along with the empirically optimal phase margin of 60° are used to calculate $r_{1,2}$ and $A_{1,2}$, as per Section 6.2.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Optimization of Phase Margin for Constituent Loops                       %%
%% Author: Xiongliang Lai                                                   %%
%% Date: Mar. 25th 2009                                                     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
syms r1 x;  % r1=wu_wz1;
syms zz1 t1;

% N1 has nothing to do with the settling time for percentage error
N1=300;
% wu1 = ClosedLoopBandwidth approximately and is set by the
% initial design
wu1=2*pi*30*10^3; % rad/s

i=1;
PM(1)=31; %%%
Increment=1; %%%%
while PM(i)<=60  % PM represents phase margin with unit in degree
    r_temp=solve(atan(x)-atan(1/x)-PM(i)/180*pi, x);
    % r_temp has two solutions; the positive one is correct
    r1(i)=double(r_temp(1,:));
    B1(i)=double((1+r1(i)^2)/(1+1/r1(i)^2));
    A1(i)=double(sqrt(wu1^4/B1(i)));

    Equ1=zz1^3+wu1*r1(i)*zz1^2+A1(i)*r1(i)^2*zz1+A1(i)*r1(i)*wu1;
    SS1=solve(Equ1, zz1);
    tt1=1e-6:1e-6:1e-4;
    for k=1:length(SS1)

sigma1(k,1)=SS1(k)*(wu1*r1(i)+SS1(k))/(2*SS1(k)*wu1*r1(i)+3*SS1(k)^2+A1
(i)*r1(i)^2)*exp(SS1(k)*t1);
      sigma1_num_temp(k,:)=subs(sigma1(k,1), t1, tt1);
    end
    sigma1_num=ones(1,length(sigma1))*sigma1_num_temp;
    Step_t1=N1-N1*sigma1_num;
    plot(tt1, abs(Step_t1), 'k'); grid; hold on;
```

```
    % 0.05 indicates 5% error within N1 * 5%
    t_settle(i)=tt1(max(find(abs((Step_t1-N1)/N1)>0.0001)));
    i=i+1;
    PM(i)=PM(i-1)+Increment;
end

pause;
hold off;
plot(PM(1:(length(PM)-1)), t_settle, 'k'); grid;

% Index of Minimum settling time and the corresponding
% Phase-Margin, r1, A1
[t_settle_min, index]=min(t_settle);


t_settle_min          %Mininum Settling Time
PhaseMargin=PM(index) %Phase Margin
r=double(r1(index))   %r1=wu_wz1
A=A1(index)

%Parameter when PhaseMargin = 60 degree
index=30;
% Phase Margin
PhaseMargin=PM(index)
% Settling time for 60 degree phase margin.
SettleTime_60=t_settle(index)
%r1=wu_wz1
r=double(r1(index))
A=A1(index)
```

A.3 Loop Filter Synthesis

The following Matlab script was used to synthesize each constituent loop's loop-filter components for given circuit parameters, preselected $\omega_{u1,2}$ and calculated $r_{1,2}$ and $A_{1,2}$, as per the design procedures in Section 6.3.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Synthesis for Loop-filter Components for Each Loop                       %%
%% Author: Xiongliang Lai                                                   %%
%% Date: Mar. 25th 2009                                                     %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initial Circuit Parameters for an Individual Loop
%Charge Pump Current in Unit of Ampere; Bigger Current -> Smaller R11;
I1=20e-6;
```

```
Kphi1=I1/2/pi;
%VCO Gain in Unit of radian/(sec.*volt.)
Kv1=2*pi*800e6;
N1=400;

% Parameters Obtained from "Optimization of Phase Margin for
Constituent
% Loops"
A1=9.5204e9;
wu1=2*pi*30*10^3;
r1=3.7321;

wp1=wu1*r1;
wz1=wu1/r1;

% Solve for Loop Filter Components: C11 C12 R11
syms C11 C12 R11;
[C11, C12, R11]=solve((C11+C12)/(R11*C11*C12)-wp1, 1/(R11*C11)-wz1,
Kphi1*Kv1/(N1*(C11+C12))-A1, C11, C12, R11);
C11=double(C11)
C12=double(C12)
R11=double(R11)
```

A.4 Architecture Analysis

The following Matlab script was used to analyze and simulate various properties and performances of each of the constituent loops and the overall architecture for the proposed frequency synthesizer. The properties and performances of a single constituent loop include open-loop zeros, poles and bode plot, closed-loop bandwidth, linear frequency-switching settling behavior and the estimated settling time, and linear frequency-switching phase capturing behavior and the estimated capture range. The properties and performances of the overall architecture include the total bandwidths and the total frequency-switching settling behaviors and the estimated settling times. Bandwidths have been shown in Figure 7.1 and settling behaviors have been illustrated in Figure 7.4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Analyses for Constituent Loops and the Cascaded Architecture:          %%
%%  1) Closed-loop Bandwidth                                              %%
%%  2) Closed-loop Poles & Zeros                                          %%
%%  3) Open-loop Bode Plot                                                %%
%%  4) Open-loop Poles & Zeros                                            %%
%%  5) Linear Frequency Switching Settling Behaviors                      %%
%%  6) Lock-in Capture Range and Transients                               %%
%% Author: Xiongliang Lai                                                 %%
%% Date: Mar. 25th 2009                                                   %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Analyses for the 1st Loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A1 r1 wu1 N1;
syms t w1;

NN1=400;    %%%
w1=2*pi*(1e3:10:1e7); %%%

% Open-loop Transfer Function
GH1=A1/s^2*(1+s*r1/wu1)/(1+s/wu1/r1);  % r1=wu_wz1

% Closed-Loop Transfer Function
%Closed-Loop Impulse Response in s Domain
ClosedLoop_s1=simple(N1*GH1/(1+GH1));
%Closed-Loop Impulse Response in t Domain
ClosedLoop_t1=simple(ilaplace(ClosedLoop_s1, s, t));
%Closed-Loop Step Response in s Domain
Step_s1=simple(1/s*ClosedLoop_s1);
%Closed-Loop Step Response in t Domain
Step_t1=simple(ilaplace(Step_s1, s, t));
%r=wu_wz1
Step1_tt01=subs(Step_t1, {A1, N1, wu1, r1}, {9.5204e+009, 400,
2*pi*30*10^3, 3.7321});  %%%

% Linear Frequency Switching Settling Behavior for a Unit Input
% Frequency Step
tt=1e-6:1e-6:1e-3/2;
Step1_tt02=subs(Step1_tt01, t, tt);
Step1_tt03=double(Step1_tt02);
plot(tt, Step1_tt03, 'k--'); hold on; grid;
% 0.05 indicates 5% error within N1 * 5%; 1.3333e-006
t1_settle=tt(max(find(abs((Step1_tt03-NN1)/NN1)>.01e-2)));

% Open-Loop Bode Plot
open1_freq01=subs(GH1, {A1, N1, wu1, r1}, {9.5204e+009, 400,
2*pi*30*10^3, 3.7321}); %%%
open1_freq02=subs(open1_freq01, {s}, {j*w1});
plot(w1, 20*log10(abs(open1_freq02))); hold on;
  wu_1=2*pi*30*10^3; r_1=3.7321; wp1=wu_1*r_1; wz1=wu_1/r_1; %%%
  zero1_freq=subs(open1_freq01, {s}, {j*wz1});
```

```
  unity1_freq=subs(open1_freq01, {s}, {j*wu_1});
  pole1_freq=subs(open1_freq01, {s}, {j*wp1});
  plot(wz1, 20*log10(abs(zero1_freq)), 'o'); hold on;
  plot(wu_1, 20*log10(abs(unity1_freq)), 'd'); hold on;
  plot(wp1, 20*log10(abs(pole1_freq)), '*'); hold off; grid;
plot(w1, angle(open1_freq02)/pi*180); hold on;
  plot(wz1, angle(zero1_freq)/pi*180, 'o'); hold on;
  plot(wu_1, angle(unity1_freq)/pi*180, 'd'); hold on;
  plot(wp1, angle(pole1_freq)/pi*180, '*'); hold off; grid

% Closed-Loop Bode Plot (Bandwidth)
closed1_freq01=subs(ClosedLoop_s1, {A1, N1, wu1, r1}, {9.5204e+009,
 400, 2*pi*30*10^3, 3.7321}); %%%
closed1_freq02=subs(closed1_freq01, {s}, {j*w1});
plot(w1/2/pi, 20*log10(abs(closed1_freq02)), 'k--'); hold on; grid;
%Find 3dB Frequency in Closed-loop Magnitude Transfer Function
f1_3dB=w1(max(find(abs(closed1_freq02)>NN1*sqrt(2)/2)))/2/pi;

% Closed-Loop Poles and Zeros
syms n1 d1;
[n1 d1]=numden(simple(closed1_freq01));
poles1=double(solve(d1, s));
zeros1=double(solve(n1, s));

% Lock-In Transfer Function
LockIn_s1=simple(1/s^2*(1-GH1/(1+GH1)));
LockIn_t1=simple(ilaplace(LockIn_s1, s, t));
LockIn1_tt01=subs(LockIn_t1, {A1, N1, wu1, r1}, {9.5204e+009,
 400, 2*pi*30*10^3, 3.7321}); %%%%%%%%%%%%%%%%%%%

% Lock-in Transients with Phase Variation and Calculation of
% Lock-In Range in Hz
LockIn1_tt02=subs(LockIn1_tt01, t, tt/2);
LockIn1_tt03=double(LockIn1_tt02);
plot(tt, LockIn1_tt03); grid; hold on;
LockInRange1=2*pi/max(LockIn1_tt03)/(2*pi); % Unit in Hz



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Analyses for the 2nd Loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A2 r2 wu2 N2;
syms t w2;

NN2=250;    %%%
w2=2*pi*(1e3:10:1e7);  %%%

% Open-loop Transfer Function
GH2=A2/s^2*(1+s*r2/wu2)/(1+s/wu2/r2);  % r=wu_wz2

% Closed-Loop Transfer Function
%Closed-Loop Impulse Response in s Domain
ClosedLoop_s2=simple(N2*GH2/(1+GH2));
%Closed-Loop Impulse Response in t Domain
ClosedLoop_t2=simple(ilaplace(ClosedLoop_s2, s, t));
```

```
%Closed-Loop Step Response in s Domain
Step_s2=simple(1/s*ClosedLoop_s2);
%Closed-Loop Step Response in t Domain
Step_t2=simple(ilaplace(Step_s2, s, t));
%r=wu_wz2
Step2_tt01=subs(Step_t2, {A2, N2, wu2, r2}, {1.2958e+010, 250,
2*pi*35*10^3, 3.7321});  %%%

% Linear Frequency Switching Settling Behavior for a Unit Input
% Frequency Step
tt=1e-6:1e-6:1e-3/2;
Step2_tt02=subs(Step2_tt01, t, tt);
Step2_tt03=double(Step2_tt02);
plot(tt, Step2_tt03, 'k-.'); hold on; grid;
% 0.05 indicates 5% error within N1 * 5%
t2_settle=tt(max(find(abs((Step2_tt03-NN2)/NN2)>.01e-2)));

% Open-Loop Bode Plot
open2_freq01=subs(GH2, {A2, N2, wu2, r2}, {1.2958e+010, 250,
2*pi*35*10^3, 3.7321}); %%%
open2_freq02=subs(open2_freq01, {s}, {j*w2});
plot(w2, 20*log10(abs(open2_freq02))); hold on;
  wu_2=2*pi*35*10^3; r_2=3.7321; wp2=wu_2*r_2; wz2=wu_2/r_2; %%%
  zero2_freq=subs(open2_freq01, {s}, {j*wz2});
  unity2_freq=subs(open2_freq01, {s}, {j*wu_2});
  pole2_freq=subs(open2_freq01, {s}, {j*wp2});
  plot(wz2, 20*log10(abs(zero2_freq)), 'o'); hold on;
  plot(wu_2, 20*log10(abs(unity2_freq)), 'd'); hold on;
  plot(wp2, 20*log10(abs(pole2_freq)), '*'); hold off; grid;
plot(w2, angle(open2_freq02)/pi*180); hold on;
  plot(wz2, angle(zero2_freq)/pi*180, 'o'); hold on;
  plot(wu_2, angle(unity2_freq)/pi*180, 'd'); hold on;
  plot(wp2, angle(pole2_freq)/pi*180, '*'); hold off; grid

% Closed-Loop Bode Plot (Bandwidth)
closed2_freq01=subs(ClosedLoop_s2, {A2, N2, wu2, r2}, {1.2958e+010,
 250, 2*pi*35*10^3, 3.7321}); %%%
closed2_freq02=subs(closed2_freq01, {s}, {j*w2});
plot(w2/2/pi, 20*log10(abs(closed2_freq02)), 'k-.'); hold on; grid;
% 3dB Frequency in Closed-loop Magnitude Transfer Function
f2_3dB=w2(max(find(abs(closed2_freq02)>NN2*sqrt(2)/2)))/2/pi;

% Lock-In Transfer Function
LockIn_s2=simple(1/s^2*(1-GH2/(1+GH2)));
LockIn_t2=simple(ilaplace(LockIn_s2, s, t));
LockIn2_tt01=subs(LockIn_t2, {A2, N2, wu2, r2}, {1.2958e+010, 250,
2*pi*35*10^3, 3.7321}); %%%

% Lock-in Transients with Phase Variation and Calculation of
% Lock-In Range in Hz
LockIn2_tt02=subs(LockIn2_tt01, t, tt/2);
LockIn2_tt03=double(LockIn2_tt02);
plot(tt, LockIn2_tt03, 'm'); grid; hold on;
LockInRange2=2*pi/max(LockIn2_tt03)/(2*pi); % Unit in Hz
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Analyses for the Cascaded Architectrue
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=1000;
N_Cas=NN1*NN2/M;    %%%
% A1 N1 wu1 r1 & A2 N2 wu2 r2 replaced by numerical values
CasLoop=closed1_freq01*closed2_freq01/M;
% Cascaded Loop Step Response in s Domain
CasLoopStep_s=simple(1/s*CasLoop);
% Cascaded Loop Step Response in t Domain
CasLoopStep_t=simple(ilaplace(CasLoopStep_s, s, t));

% Linear Frequency Switching Settling Behavior for a Unit Input
% Frequency Step
tt=1e-6:1e-6:1e-3/2;
CasLoopStep_tt01=subs(CasLoopStep_t, t, tt);
CasLoopStep_tt02=double(CasLoopStep_tt01);
plot(tt, CasLoopStep_tt02, 'k'); hold off; grid;
%Calculate settling time within 5% of the final frequency
%0.05 indicates 5% error within N1 * 5%
tCas_settle=tt(max(find(abs((CasLoopStep_tt02-N_Cas)/N_Cas)>.01e-2)));

% Cascaded-Loop Bode Plot (Bandwidth)
wCas=2*pi*(1e3:10:1e7); %%%
CasLoop_freq01=subs(CasLoop, {s}, {j*wCas});
plot(wCas/2/pi, 20*log10(abs(CasLoop_freq01)), 'k'); grid; hold on;
% 3dB Frequency in Closed-loop Magnitude Transfer Function
fCas_3dB=wCas(max(find(abs(CasLoop_freq01)>N_Cas*sqrt(2)/2)))/2/pi;
```

A.5 Continuous Phase-noise Analysis

The following Matlab script was used to simulate continuous close-in phase-noise variations in the components of the proposed architecture. Its application to the 1st-loop VCO continuous phase noise was explained in Section 6.5 as filtering by the constituent loops of the architecture and the simulation results was illustrated in Figure 7.3.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Analysis of 1st-loop VCO Continuous Phase Noise              %%
%% Author: Xiongliang Lai                                       %%
%% Date: Mar. 25th 2009                                         %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
```

```
% Leeson's Phase-Noise Polynomial
fm=1e3:5e3:1e9; %(Hz)  %%%
wm=2*pi*fm;
nVCO=9.8e-15+21.7./fm.^2+7.6e6./fm.^3;
plot(fm, 10*log10(nVCO), '--k'); hold on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  1st-loop Continuous Output Phase-Noise Spectrum
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A1 r1 wu1;

N1=400;  %%%
w1=wm;  %%%

% 1st-loop Open-Loop Transfer Function
GH1=A1/s^2*(1+s*r1/wu1)/(1+s/wu1/r1);  % r1=wu_wz1

% 1st-loop H_L Function in s Domain
H_L_s1=simple(GH1/(1+GH1));
H_L_s1_freq01=subs(H_L_s1, {A1, wu1, r1}, {9.5204e9, 2*pi*30*10^3,
3.7321}); %%%
H_L_s1_freq02=subs(H_L_s1_freq01, {s}, {j*w1});

% plot(fm, 10*log10(abs(H_L_s1_freq02))); hold on;
% plot(fm, 10*log10(abs(1-H_L_s1_freq02))); grid;

% 1st-loop Output Phase-Noise Spectrum
n1=abs(1-H_L_s1_freq02).^2.*nVCO;

plot(fm, 10*log10(n1), '-.k');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Bridging-divider Continuous Output Phase-Noise Spectrum
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=1000;
n_bridge=(1/M)^2.*n1;
plot(fm, 10*log10(n_bridge), ':k');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  2nd-loop Continuous Output Phase-Noise Spectrum
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A2 r2 wu2;

% Pick N2 maximum for worst case
N2=250;  %%%
w2=wm;  %%%

% 2nd-loop Open-Loop Transfer Function
GH2=A2/s^2*(1+s*r2/wu2)/(1+s/wu2/r2);  % r2=wu_wz2

% 2nd-loop H_L Function in s Domain
H_L_s2=simple(GH2/(1+GH2));%Closed-Loop Impulse Response in s domain
H_L_s2_freq01=subs(H_L_s2, {A2, wu2, r2}, {1.2958e10, 2*pi*35*10^3,
3.7321}); %%%
```

```
H_L_s2_freq02=subs(H_L_s2_freq01, {s}, {j*w2});

% 2nd-loop Output Phase-Noise Spectrum
n2=abs(N2.*H_L_s2_freq02).^2.*n_bridge;
plot(fm, 10*log10(n2), 'k'); grid;
```

A.6 Discrete Phase-spur Analysis

The following Matlab script was used to analyze phase-spur variations when passing through the components in the proposed architecture. The spur generation mechanism in fractional-N PLLs was explained in Section 3.2; the filtering effects applied to the spurs by the constituent loops of the architecture and the measuring principles of the spur strengths were discussed in Section 6.4; and the resulted performances were illustrated in Figure 7.2.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Analysis of Discrete Phase Spurs from Fractional-N Mechanism in %%
%%  Mini-denominator Mode                                          %%
%% Author: Xiongliang Lai                                         %%
%% Date: Mar. 25th 2009                                           %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
% Circuit Parameter Initialization
A=1;
Q=12;
N1=380;  %%%
N1_star=N1+A/Q;
I=20e-6; % amp; Charge Pump Current
Kd=I/(2*pi); % amp/rad; Phase Detector Gain

% Fractional-N Compensation & Overflow Realization
Compen=zeros(1,Q);
overflow=zeros(1,Q);
for cycle=2:Q
    Compen(cycle)=rem(Compen(cycle-1)+A,Q);
    if Compen(cycle-1)>Compen(cycle)
        overflow(cycle-1)=1;
    end
end
```

```
PhaseError=-Compen;  % Phase Error
n_DET=1/N1_star*2*pi/Q*Kd*PhaseError;  % amp; Phase Detector Noise

% Sample the continuous-time phase detector noise.
fr=10e6; % Hz; Reference Frequency
Tr=1/fr;
N=100;   % Number of Samples in One Tr
tau=Tr/N;
pointer=1;

for cycle=1:Q % Q*Tr is the period of the phase detector noise,.
    x(pointer:pointer+N-1)=n_DET(cycle);
    pointer=N+pointer;
end

% Define the number of repetitions in the piecewise continuous time
signal.
Repetition=1024;
Rep=log2(Repetition);

for l=1:Rep
    x=[x x];
end

% Discrete Time Fourier Transform
N=length(x);
% !!! Multiplying tau to recover the original non-sampled signal
continuous
% time fourier transform.
X=1/N*fft(x);

% Swap the order in X.
X=fftshift(X);

% Corresponding Frequencies of Discrete Time Fourier Transform
k=0:1:N-1;
W=(k*(2*pi/N)-pi)/tau; % Frequency in radian/s
F=W/2/pi; % Frequency in /s

% Plot the power spectrum (CTFT).
plot(F,abs(X).^2);
grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Output Discrete Phase Spurs from the 1st-loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A1 r1 wu1;

w1=W;

% 1st-loop Open-Loop Transfer Function
GH1=A1/s^2*(1+s*r1/wu1)/(1+s/wu1/r1);  % r1=wu_wz1

% 1st-loop H_L Function in s Domain
```

```
H_L_s1=simple(GH1/(1+GH1));
H_L_s1_freq01=subs(H_L_s1, {A1, wu1, r1}, {9.5204e9, 2*pi*30*10^3,
3.7321}); %%%
H_L_s1_freq02=subs(H_L_s1_freq01, {s}, {j*w1});

% 1st-loop Output Phase Spurs
theta_n_stage1=abs(N1_star/Kd*H_L_s1_freq02).^2.*abs(X).^2+1e-12;

% Plot Spurs in dB(or equivalently dBc).
plot(F, 10*log10(theta_n_stage1),'k');
grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Output Discrete Phase Spurs from the Bridging Divider
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=1000;  %%%
theta_n_bridge=(1/M)^2*theta_n_stage1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Output Discrete Phase Spurs from the 2nd Loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms s A2 r2 wu2;

w2=W;
N2=250; % Pick N2 maximum for worst case %%%

% 2nd-loop Open-Loop Transfer Function
GH2=A2/s^2*(1+s*r2/wu2)/(1+s/wu2/r2);  % r2=wu_wz2

% 2nd-loop H_L Function in s Domain
H_L_s2=simple(GH2/(1+GH2));
H_L_s2_freq01=subs(H_L_s2, {A2, wu2, r2}, {1.2958e10, 2*pi*35*10^3,
3.7321}); %%%

H_L_s2_freq02=subs(H_L_s2_freq01, {s}, {j*w2});

% 2nd-loop Output Phase Spurs
theta_n_stage2=abs(N2*H_L_s2_freq02).^2.*theta_n_bridge;

% Plot Spurs in dB(or equivalently dBc).
plot(F, 10*log10(theta_n_stage2),'k');
grid;
```

References

[1] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Hoboken, NJ: Wiley-Interscience, 2005.

[2] T. K. K. Kan, G. C. T. Leung and H. C. Luong, "A 2-V 1.8-GHz Fully Integrated CMOS Dual-Loop Frequency Synthesizer," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1012-1020, Aug. 2002.

[3] W. S. T. Yan and H. C. Luong, "A 2-V 900-MHz Monolithic CMOS Dual-Loop Frequency Synthesizer for GSM Receivers," *IEEE J. Solid-State Circuits*, vol. 36, pp. 204-216, Feb. 2001.

[4] P. P. Sotiriadis, "Principles of Cascaded Diophantine Frequency Synthesis," in *2008 IEEE Int. Frequency Control Symp.*, pp. 181.

[5] B. Razavi, *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 1998.

[6] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd ed. New York: Cambridge Univ. Press, 2003.

[7] G. Bianchi, *Phase-Locked Loop Synthesizer Simulation*. Blacklick, OH: McGraw-Hill, 2005.

[8] S. E. Meninger and M. H. Perrott, "A Fractional-N Frequency Synthesizer Architecture Utilizing a Mismatch Compensated PFD/DAC Structure for Reduced Quantization-Induced Phase Noise," *IEEE Trans. Circuits Syst. II*, vol. 50, pp. 839-849, Nov. 2003.

[9] I. Galton, "Delta-Sigma Fractional-N Phase-Locked Loops," in *Phase-Locking in High-Performance Systems: From Devices to Architectures*, B. Razavi, Ed. Wiley-IEEE Press, 2003, pp. 23-32.

[10] J. Craninchx and M. S. J. Steyaert, "A Fully Integrated CMOS DCS-1800 Frequency Synthesizer," *IEEE J. Solid-State Circuits*, vol. 33, pp. 2054-2065, Dec. 1998.

[11] Z. Shu, K. L. Lee and B. H. Leung, "A 2.4-GHz Ring-Oscillator-Based CMOS Frequency Synthesizer With a Fractional Divider Dual-PLL Architecture," *IEEE J. Solid-State Circuits*, vol. 39, pp. 452-462, Mar. 2004.

[12] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Boston, MA: McGraw-Hill, 2001.

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

[14] W. B. Davenport and W. L. Root, *An Introduction to the Theory of Random Signals and Noise*. New York: McGraw-Hill, 1958.

[15] D. T. Comer, "Clock Generation and Synchronization for High-Speed Serial Data Systems," Pro*ceedings of the 7th European Frequency and Time Forum*, March, 1993, Neuchatel Switzerland, pp. 251-254.

[16] D. T. Comer, "A CMOS Phase Detector for Mixed Signal ASIC Application," *Proceedings of the International ASIC Conference*, Rochester, NY, August, 1993, pp. 232-234.

[17] C. Barrett, "Fractional/Integer-N PLL Basics," *Texas Instruments Technical Brief SWRA029*, August, 1999.

[18] N. Tan, *Switched-current Design and Implementation of Oversampling A/D Converters*. Norwell, MA: Kluwer Academic Publishers, 1997.

[19] K. Shu and E. Sanchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*. New York, NY: Springer, 2005.

[20] S. Norsworthy, R. Schreier and G. Temes, Eds., *Delta-Sigma Data Converters: Theory, Design and Simulation*. New York, NY: IEEE Press, 1997.

[21] T. Kenny, T. Riley, N. Filiol and M. Copeland, "Design and Realization of a Digital $\Delta\sum$ Modulator for Fractional-N Frequency Synthesis," *IEEE Trans. on Vehicular Technology*, vol. 48, pp. 510-521, Mar. 1999.

[22] L. Sun, T. Lepley, F. Nozahic, A. Bellissant, T. Kwasniewski and B. Heim, "Reduced Complexity High Performance Digital Delta-Sigma Modulator for Fractional-N Frequency Synthesis," *IEEE Proc. ISCAS*, vol. 2, Orlando, FL, July 1999, pp. 152-155.

[23] D. Mijuskovic, M. Bayer, T. Chomicz, N. Garg, F. James, P. McEntarfer and J. Porter, "Cell-based Fully Integrated CMOS Frequency Synthesizers," *IEEE J. Solid-State Circuits*, vol. 29, pp. 271-279, Mar. 1994.

[24] Y. Koo, H. Huh, Y. Cho, J. Lee, J. Park, K. Lee, D. Jeong and W. Kim, "A Fully Integrated CMOS Frequency Synthesizer with Charge-averaging Charge Pump and Dual-path Loop Filter for PCS- and Cellular-CDMA Wireless Systems," *IEEE J. Solid-State Circuits*, vol. 27, pp. 536-542, May 2002.

[25] J. Silva-Martinez and A. Vazquez-Gonzalez, "Impedance Scalers for IC Active Filters," in *IEEE Proc. ISCAS'98*, vol. 1, June 1998, pp. 151-154.

[26] J. Craninckx and M. Steyaert, "A 1.75-GHz/3-V Dual-Modulus Divide-by-128/129 Prescaler in 0.7-um CMOS," *IEEE J. Solid-State Circuits*, vol. 31, pp. 890-897, July 1996.