NTNU – Trondheim
Norwegian University of
Science and Technology

Python Framework for Design and Simulation of Integer-N ADPLLs

Cole Nielsen

Electronic Systems Design, Specialization Project

Submission date:	December 2019
Supervisor:	Trond Ytterdal, IET
Co-supervisor:	Carsten Wulff, IET

Norwegian University of Science and Technology
Department of Electronic Systems

Abstract.

A pure-Python simulation and design automation framework for the design of integer-N all digital phase locked loop (ADPLL) frequency synthesizers is presented in this paper. The framework enables (a) automatic design of optimized discrete-time and quantized digital PLL loop filters, and (b) discrete-time simulation to verify filter and PLL designs for phase noise performance, lock-time and stability subject to variation using Monte-Carlo sampling. Simulation is performed with a discrete-event time domain simulator utilizing behavioral PLL component models, which permit accurate modeling effects of time-discretization and quantization in a ADPLL. Filter design automation is achieved via optimization of a prototype loop filter in a PLL for minimum total phase noise power and lock time, where lock time is maximally constrained. The optimizer utilizes continuous phase transfer function approximations of discrete PLL loop, allowing computationally fast estimates of phase noise and lock time. Thus continuous-to-discrete time conversion and coefficient quantization of the optimal filter is applied to yield a end loop-filter design. Second order optimization is utilized to minimize loop filter design error due to quantization effects.

Problem description.

To develop a standalone PLL simulation framework to aid and facilitate a later master's thesis project regarding the design of an all-digital, ultra-low power PLL (ULPPLL) frequency synthesizer. This framework is intended to address and ease all-digital PLL design and simulation challenges at a high level. Specifically it should allow for speedy PLL simulation defined with system and component- level specifications (e.g. desired frequency, gain of digitally contolled oscillator, phase detector resolution, divider modulus). This is to allow for development of component level specifications and validation of ideal PLL performance (phase noise, lock time, stability) before transistor level implementation. Furthermore, the framework should automate design of any pure-digital PLL components, namely loop-filter design, to accelerate overall time of PLL implementation.

Contents

1 Introduction **10**

2 Theory **11**
- 2.1 Continuous PLL Model . . . 11
  - 2.1.1 PLL Synthesizer architecture . . . 12
  - 2.1.2 Divider . . . 12
  - 2.1.3 Phase detector . . . 12
  - 2.1.4 Loop Filter . . . 12
  - 2.1.5 VCO . . . 13
  - 2.1.6 Continuous PLL Transfer function . . . 13
  - 2.1.7 PI-loop filter design . . . 14
  - 2.1.8 PI-controller peaking compensation . . . 15
  - 2.1.9 Regarding other PID controller permutations . . . 15
- 2.2 Digital, discretized PLL Model . . . 16
  - 2.2.1 Divider . . . 17
  - 2.2.2 TDC . . . 17
  - 2.2.3 Loop Filter . . . 17
  - 2.2.4 DCO . . . 19
  - 2.2.5 Discrete-time PLL transfer function . . . 19
- 2.3 PLL Noise . . . 20
  - 2.3.1 TDC noise . . . 20
  - 2.3.2 DCO noise . . . 21
  - 2.3.3 Divider noise . . . 22
  - 2.3.4 Loop filter noise . . . 23
  - 2.3.5 PLL output-referred noise . . . 23
  - 2.3.6 PLL noise sensitivity transfer functions . . . 23

3 Methods **24**
- 3.1 Behavioral, discrete event PLL simulation . . . 24
  - 3.1.1 Phase noise modeling . . . 24
  - 3.1.2 Measurement of phase noise, spurs . . . 24
  - 3.1.3 Monte-carlo sampling . . . 24
- 3.2 Loop filter optimization . . . 24
  - 3.2.1 Estimation of settling time . . . 24
  - 3.2.2 Estimation of PLL phase noise . . . 25
  - 3.2.3 Optimization algorithm . . . 25

4 Discussion **26**

5 Conclusion **26**

6 Block diagram **26**

7 Specifications **26**

8 Baseband phase noise **27**
- 8.1 Modulated Signal . . . 27
- 8.2 Local oscillator - noisy PLL . . . 27

8.3 Baseband phase noise 27

9 DCO tuning 28
9.1 Backgate tuning 28
9.2 Ring oscillator frequency derivation 28
9.2.1 Finding $\langle g_{ch} \rangle$ and C 29
9.2.2 Handling unequal NMOS/PMOS 30
9.2.3 Solving for oscillator frequency and power 30
9.3 Ring oscillator backgate tuning derivation 31
9.4 DCO Gain Uncertainty 32

A Appendix - Schedule 34

B Appendix - Code 34

C pres1 34
C.1 Motivation 34
C.2 State of the Art 35
C.3 Objectives - Synthesizer goals 35
C.4 Architecture - concepts 35
C.5 State of Art - Sub-1 mW PLLS 36
C.6 Physical limits 36

D pres2 36
D.1 Initial simulation/modeling approach 36
D.2 Implementation in Cadence 37

E pres4 37
E.1 Simulation approach 37
E.2 DCO performance criteria 37
E.3 DCO - reference spur requirements 38
E.4 DCO - steady state cap noise 38
E.5 DCO - linearity and accuracy 39
E.6 TDC - phase noise model 39

F pres5 39
F.1 This week 39
F.2 Next week 39
F.3 Loop filter original attempt 40
F.4 Loop filter new approach 40
F.5 Coarse frequency estimation 40
F.6 Loop filter Gain Coefficients 41
F.7 Coarse frequency calibration 41

List of Figures

1 Basic phase feedback network. 11
2 Basic PLL. 12
3 PI-controller PLL pole-zero locations. 14
4 Example PI-PLL responses with varied ζ. 15
5 Basic ADPLL. 16
6 TDC model. 17
7 Implementation of filter. 18
8 Discrete time PLL model. 19
10 Quantization as via additive error signal. 21
11 DCO additive noise model. 21
12 Thermal to phase noise conversion. 21
13 Model for ring oscillator. 29
14 Approximate model for ring oscillator inverter delay cell. 29
15 Backgate-tuned ring oscillator with coarse tuning capacitor bank. 31

List of Tables

1 System-level specifications 27
2 Component-level specifications. 27

Abbreviations.

BBPD	Bang-bang phase detector
BW	Bandwidth
CMOS	Complementaty metal oxide semiconductor
DAC	Digital to analog converter
DCO	Digitally controlled oscillator
DFT	Discrete Fourier transform
DSP	Digital signal processing
FDSOI	Fully depleted silicon on insulator
FET	Field effect transistor
FFT	Fast Fourier transform
FM	Frequency modulation
FRAC	Fractional
FSK	Frequency shift keying
IF	Intermediate frequency
INT	Integer
LF	Loop filter
LO	Local oscillator
MOSFET	Metal oxide semiconductor field effect transistor
NMOS	N-channel MOSFET
OTW	Oscillator tuning word
PD	Phase detector
PLL	Phase locked loop
PMOS	P-channel MOSFET
PN	Phase noise
PSD	Power spectral density
PSK	Phase shift keying
RC	Resistor-capacitor
RF	Radio frequency
RO	Ring oscillator
RX	Receiver
SNR	Signal to noise ratio
SSB	Single side band
TDC	Time to digital converter
TSPC	True single phase circuit
ULPPLL	Ultra-low power PLL
VCO	Voltage controlled oscillator
WURX	Wake up receiver

1 Introduction

Phase locked loops are extraordinarily useful frequency synthesizers that are vital to the operation of virtually all wired and wireless communication systems of today. The trend towards increasingly lower power wireless devices poses an acute need to reduce PLL power consumption. This is a challenge as PLLs typically rank among the highest power consuming components of a radio, and are necessarily so to limit phase noise. Reducing analog PLL power consumption can be a prohibitive challenge as the performance of analog loop filters degrade as a result of unavoidably lower charge pump current. However, recent CMOS process nodes with minimum gate lengths as small as 7nm allow for all-digital PLLs as an attractive alternative to analog designs due to low power consumption associated with the implementation of a digital loop filter. Digital loop-filters have a unique advantage where they can be scaled indefinitely as process nodes advance, while suffering no loss in performance.

All-digital PLLs introduce new challenges in the process of design in the ultra-low power domain. Low power design is complemented by low complexity design, which in a PLL transcribes to low resolution of phase detectors, digitally tunable oscillators, and loop filter digital data paths. In other words, effects of quantization are strong. Whereas analog designs and high power/resolution digital designs can be effectively analyzed (even by hand) with transfer functions in the Z-domain, time domain simulation is necessary to capture quantization effects in low power, low resolution ADPLLs. This, of course, presents a challenge in manual design of PLLs if simulation is an integral part to the most basic modeling, and thus motivates the creation of an automated design solution. Currently, no PLL design framework is known that automates PLL design for the needs of ultra low power PLLs as characterized here. Of the most prominent pre-existing frameworks, CppSim [add reference...] features a utility for design of continuous loop filters, however for this it does not provide any level of performance optimization, nor does it support discrete-time and quantized digital designs.

Thus in this paper, a new framework written in pure-Python is introduced, which uniquely addresses issues of ultra-low power ADPLL design. Specifically, design of integer-N type PLLs is focused on, as the impetus of this work is an integer-N PLL design project. Topics presented are (a) automatic design and optimization of ADPLL loop filters given target system and component level specs for the PLL, and (b) behavioral time domain PLL simulation for accurate analysis and verification of PLL performance, with an integrated Monte-Carlo sampling variation analysis engine.

A brief outline of the paper is as follows. An introduction to PLL theory is in section 2. Simulation and optimization methods are discussed in section 3. An example design exercise, a comparison to existing solutions, and general discussion considerations for using the framework are in section 4. Finally, section 5 concludes.

2 Theory

In its most basic form, a phase locked loop is a phase-based feedback system whose output tracks or maintains a fixed phase relationship to an input signal. Such a system is uniquely suited to the task of frequency synthesis, which is the process of generating derivative frequencies from some reference frequency. Given reference and output phase signals Φ_{ref} and Φ_{out}, a PLL can be modeled as in figure 1, with feedforward and feedback networks A(s) and B(s).

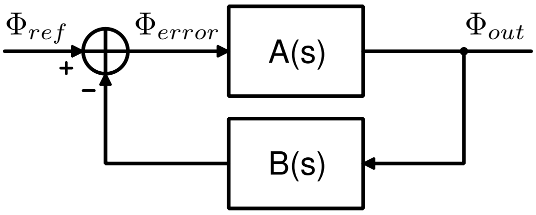


Figure 1: Basic phase feedback network.

The phase relationship for Φ_{ref} and Φ_{out} is therefore:

$$\frac{\Phi_{out}}{\Phi_{ref}} = \frac{A(s)}{1 + A(s)B(s)} \tag{1}$$

A particular case of interest is when B(s) = 1/N, where N is a constant, and the loop gain A(s)B(s) $>>$ 1:

$$\frac{\Phi_{out}}{\Phi_{ref}} \approx \frac{A(s)}{A(s)B(s)} = \frac{1}{B(s)} = N \tag{2}$$

We see that the phase through the PLL is multiplied by a factor of N. If the input phase signal is a sinusoid with frequency ω_{ref}, and likewise the output with ω_{out}, then $\phi_{ref}(t) = \omega_{ref}t$ and $\phi_{out}(t) = \omega_{out}t$. Thus:

$$\frac{\Phi_{out}}{\Phi_{ref}} = \frac{\omega_{out}t}{\omega_{ref}t} \approx N \quad \rightarrow \quad \omega_{out} \approx N\omega_{ref} \tag{3}$$

Therefore, it is observed that a PLL allows for the generation, i.e. synthesis, of a new frequency from a reference frequency signal. Given a feedback divider ration of 1/N, the PLL multiplies the reference frequency by a factor of N. In the following sections, more advanced models for PLL will be developed, extending the concept introduced here. Specifically, the theory of digital, discrete-time PLLs will be developed and extended from a continuous phase model of a basic PLL.

2.1 Continuous PLL Model

Although PLLs are practically limited to using discrete-time sampling in real-word hardware, continuous models can still be applied in their analysis and design. Thus a continuous PLL model is developed in this section to aid in the later discussed discretized PLL modeling.

2.1.1 PLL Synthesizer architecture

The traditional architecture for implementing a PLL frequency synthesizer [3] is shown in figure 2. This basic PLL is comprised of four components: (1) a phase detector, denoted by PD, (2) a loop filter, denoted by H$_{LF}$(s), (3) a voltage controlled oscillator, denoted by VCO, and (4) and phase divider, denoted by ”÷ N” in the figure. These components are explained in the following sections.

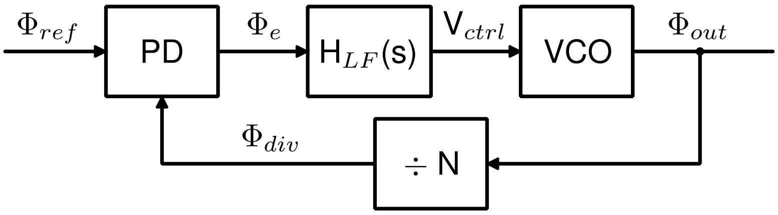


Figure 2: Basic PLL.

2.1.2 Divider

The phase divider is used as the feedback path in the PLL, where the division modulus N controls the frequency multiplication of the PLL. The tranfer function of the divider is:

$$\frac{\Phi_{div}}{\Phi_{out}} = \frac{1}{\mathrm{N}} \tag{4}$$

2.1.3 Phase detector

The phase detector is used to measure the phase of the feedback signal (i.e. divider output) in relation to the reference phase, in order to establish a phase error signal Φ_e used to control the tuning of the PLL.

$$\Phi_e = \Phi_{ref} - \Phi_{div} \tag{5}$$

2.1.4 Loop Filter

The PLL loop filter is used to control the phase-frequency response of PLL, which affects transient PLL behavior, as well as phase noise performance (see section 2.3). As will later be seen, low pass response is desired in a PLL, so the loop filter must be designed accordingly. Generally, this can be designed arbitrarily to have N poles and M zeros, as such:

$$\mathrm{H}_{LF}(s) = \frac{\Sigma_0^M b_m s^m}{\Sigma_0^N a_n s^n} \tag{6}$$

Rational choice of the loop filter will ensure that the PLL will be stable and minimize steady state phase error. In order to achieve zero steady state error, the loop filter must contain a pole at zero, in other words an integrator. A PLL containing such a pole is classified as a Type II

PLL, and a PLL omitting the pole is considered Type I. A logical approach to loop filter design for zero-phase error is to treat it as a PID controller, where:

$$\mathrm{H}_{LF}(s) = sK_d + K_p + \frac{K_i}{s} = \frac{K_d}{s}\left(s^2 + s\frac{K_p}{K_d} + \frac{K_i}{K_d}\right) \tag{7}$$

Such a loop filter contains two zeros and one integrator pole at zero. The gain parameters K_p, K_i, K_d can be tuned to achieve the desired bandwidth and stability for the PLL. The impacts of loop filter design will be further considered in sections 2.1.6-2.1.9.

2.1.5 VCO

The voltage controlled oscillator is an oscillator with frequency controlled by an input signal V_{ctrl}. The VCO is characterized by its gain $K_{DCO} = \partial f / \partial \mathrm{V}_{ctrl}$, and the nominal oscillation frequency f_0. Analyzed in terms of phase, an oscillator can be seen as a time-phase integrator:

$$\Phi_{VCO} = \Phi_{out} = \int 2\pi(K_{DCO}\mathrm{V}_{ctrl}(t) + f_0)\mathrm{dt} \tag{8}$$

In s-domain, where frequency offsets will be represented via initial conditions for modeling purposes, the VCO transfer function is therefore:

$$\frac{\Phi_{VCO}}{\mathrm{V}_{ctrl}} = \frac{\Phi_{out}}{\mathrm{V}_{ctrl}} = \frac{2\pi K_{DCO}}{s} \tag{9}$$

2.1.6 Continuous PLL Transfer function

Now that the continuous PLL synthesizer is understood at a component level, the closed loop dynamics of the PLL can be analyzed. First the PLL loop gain is determined:

$$\mathrm{OL}(s) = \mathrm{H}_{LF}(s)\mathrm{H}_{VCO}(s)\mathrm{H}_{DIV}(s) = \frac{2\pi K_{VCO}K_d}{\mathrm{N}}\frac{1}{s^2}\left(s^2 + s\frac{K_p}{K_d} + \frac{K_i}{K_d}\right) \tag{10}$$

With the phase detector as the feedback summation point, the closed loop response of the PLL from reference to output is:

$$\frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = \frac{\mathrm{H}_{\mathrm{DIV}}^{-1}\mathrm{OL}(s)}{1+\mathrm{OL}(s)} = \frac{2\pi K_{VCO}\left(s^2K_d + sK_p + K_i\right)}{s^2\left(1 + \frac{2\pi K_{VCO}K_d}{\mathrm{N}}\right) + \frac{2\pi K_{VCO}}{\mathrm{N}}\left(sK_p + K_i\right)} \tag{11}$$

It should be noted that in the closed loop configuration, this PLL phase transfer function contains two poles and two zeros. This is not a low pass response as desired for a satisfactory PLL phase noise power spectrum, as will later be discussed. In order achieve low pass operation, the derivative term K_d must be set to zero, yielding a PI controller for the loop filter (with one zero and two poles):

$$\frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = \frac{\mathrm{H}_{\mathrm{DIV}}^{-1}\mathrm{OL}(s)}{1+\mathrm{OL}(s)} = \frac{2\pi K_{VCO}\left(sK_p + K_i\right)}{s^2 + \frac{2\pi K_{VCO}}{\mathrm{N}}\left(sK_p + K_i\right)} \tag{12}$$

Steady state zero phase error can be verified by setting s=0 in the PI-controller PLL transfer function:

$$e_\Phi(s)\big|_{s=0} = \left(\Phi_{ref}(s) - \frac{\Phi_{out}(s)}{\mathrm{N}}\right)\bigg|_{s=0} = \Phi_{ref}(s)\left(1 - \frac{\Phi_{out}(s)}{\mathrm{N}\Phi_{ref}(s)}\right)\bigg|_{s=0} = \Phi_{ref}(s)\left(1 - \frac{\mathrm{N}}{\mathrm{N}}\right) = 0 \tag{13}$$

2.1.7 PI-loop filter design

Given a PI-controller loop filter, which can be optionally represented using a pole at zero and a zero with $\omega_z = K_i/K_p$:

$$\mathrm{H}_{LF}(s) = K_p + \frac{K_i}{s} = \frac{K_i}{s}\left(\frac{s}{\omega_z} + 1\right) \tag{14}$$

Selection of (not-necessarily optimal) PI controller gains can be easily derived from overall PLL settling time requirements. Suppose that settling time t_s is defined such that the PLL settles within $\pm\delta$ of the final value for a step input. If the initial and final PLL output frequencies are f_i and Nf_{ref}, and settling with $\pm f_{tol}$ is desired, $\delta = f_{tol}/|f_i - Nf_{ref}|$. Setting time is therefore:

$$t_s = -\tau\ln(\delta) \tag{15}$$

Thus, to find settling time, a value for the PLL time constant τ must be derived. Rewriting equation 12 with substitutions $\omega_z = K_i/K_p$ and $\mathrm{K} = 2\pi K_{VCO}K_i/\mathrm{N}$:

$$\frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = \mathrm{N}\cdot\frac{s\frac{K}{\omega_z} + K}{s^2 + s\frac{K}{\omega_z} + K} \tag{16}$$

If the second order denominator can be redefined in terms of a natural frequency ω_n and damping ζ, such that:

$$s^2 + s\frac{K}{\omega_z} + K = s^2 + s2\zeta\omega_n + \omega_n^2 \tag{17}$$

It is then found that $\omega_n = \sqrt{K}$, and $\omega_z = \sqrt{K}/2\zeta$. The poles of equation 16 are then located at s = $\zeta\sqrt{K} \pm \sqrt{K}\sqrt{1-\zeta^2}$. The settling time of the PLL will be determined by the real portion of dominant pole of equation 16, specifically $\tau = 1/|\min(\Re(\{s_{p1}, s_{p2}\}))|$. Based on the pole-zero plot of figure 3, it can be observed that the dominant pole location is maximized with $\zeta = 1$. The pole-zero loci orientations are based on increasing ζ values. According to Razavi [2], ζ is usually "chosen to be $> \sqrt{2}/2$ or even 1 to avoid excessive ringing."

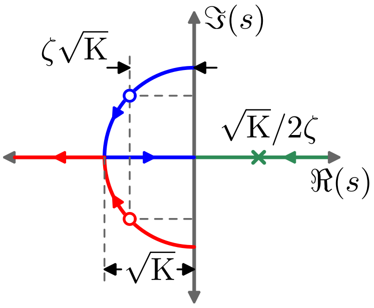


Figure 3: PI-controller PLL pole-zero locations.

To illustrate the effect of damping, figure 4 illustrates the example frequency and step responses of a PI-controlled PLL with N=1. Notice excessive peaking and ringing for $\zeta < \sqrt{2}/2$. The peaking observed in the frequency response is unavoidable with the PI-PLL due to the inherent zero in the transfer function. Its effect can be reduced with large ζ, however this will increase PLL settling time.

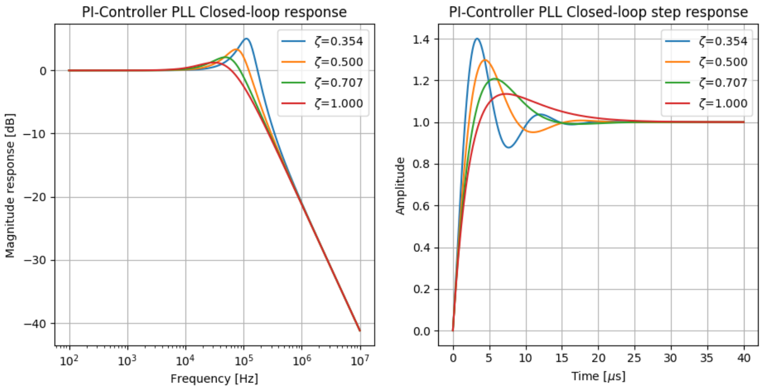


Figure 4: Example PI-PLL responses with varied ζ.

If ζ is constrained to ≤ 1:

$$\tau = \frac{1}{|\min(\Re(\{s_{p1}, s_{p2}\}))|} = \frac{1}{\zeta\sqrt{K}} \tag{18}$$

Thus:

$$t_s = \frac{-\ln(\delta)}{\zeta\sqrt{K}} = \frac{-\ln\left(\frac{f_{tol}}{|f_i - Nf_{ref}|}\right)}{\zeta\sqrt{K}} \tag{19}$$

Based on specification for settling time and damping ζ, the values for K and ω_z can be determined. If K_{VCO} and N are also specified, the PI gain coefficients can be solved using $\omega_z = K_i/K_p$ and K $= 2\pi K_{VCO}K_i/$N.

$$\omega_z = \frac{-\ln(\delta)}{2t_s} = \frac{-\ln\left(\frac{f_{tol}}{|f_i - Nf_{ref}|}\right)}{2t_s} \tag{20}$$

$$K = \frac{\ln^2(\delta)}{\zeta^2 t_s^2} = \frac{\ln^2\left(\frac{f_{tol}}{|f_i - Nf_{ref}|}\right)}{\zeta^2 t_s^2} \tag{21}$$

2.1.8 PI-controller peaking compensation

To compensate for closed loop peaking, the original PI-controller loop filter of equation 14 can be modified with the addition of a single tunable pole at ω_p. The closed loop response becomes third order, which complicates direct analysis and design of the loop filter. However, utilizing the automated filter optimization approach described later in this paper resolved issues regarding filter design in this case.

$$\mathrm{H}_{LF}(s) = \frac{K_i}{s}\frac{\left(\frac{s}{\omega_z} + 1\right)}{\left(\frac{s}{\omega_p} + 1\right)} \tag{22}$$

2.1.9 Regarding other PID controller permutations

If individual terms within the PID-controller are dropped, different controller permutations (PD, ID, PI, P, I, D) can be achieved. As mentioned before, inclusion of an integral term is needed to ensure the desired zero steady state error for a PLL. This leaves ID and I-controllers as possible alternative solutions to PI for the loop filter. However, it can be easily found that neither of these controllers result in a stable PLL, which leaves a PI-controller as the only viable PID implementation.

I-only controller

Setting the K_p and K_d terms of equation 11 to zero yields:

$$\frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = \frac{2\pi K_{VCO} K_i}{s^2 + \frac{2\pi K_{VCO}}{\mathrm{N}} K_i} \tag{23}$$

This closed loop transfer function results in a pair of poles at $\pm\sqrt{2\pi K_{VCO} K_i / \mathrm{N}}$. This will never be stable, as it can only be manifested as (1) a pair of poles on the imaginary axis, which is an oscillator, or (2) a real pole in the right-half plane and a real pole in the left-half plane, the former of which is not causally stable.

ID-controller

Setting the K_p term of equation 11 to zero yields:

$$\frac{\Phi_{out}(s)}{\Phi_{ref}(s)} = \frac{2\pi K_{VCO}\left(s^2 K_d + K_i\right)}{s^2\left(1 + \frac{2\pi K_{VCO} K_d}{\mathrm{N}}\right) + \frac{2\pi K_{VCO}}{\mathrm{N}} K_i} \tag{24}$$

The poles of this transfer have the same form as the I-only controller, and this PLL-controller configuration is unstable for the same reasons as the I-controller PLL.

2.2 Digital, discretized PLL Model

Based on the continuous PLL theory, a model for digital, discretized PLLs (i.e. ADPLLs) can be adapted. The general approach here is to utilize the bilinear transformation between continuous s-domain models to the discrete z-domain models. As commonly cited in PLL literature from a seminal paper by Gardner [1], if the sampling frequency $f_s > 10 \cdot \mathrm{BW}_{loop}$, where BW_{loop} is the PLL loop bandwidth, the effects of time sampling are easily ignored for purposes of analysis. Thus the design methods established in this paper are predicated on $f_s > 10 \cdot \mathrm{BW}_{loop}$.

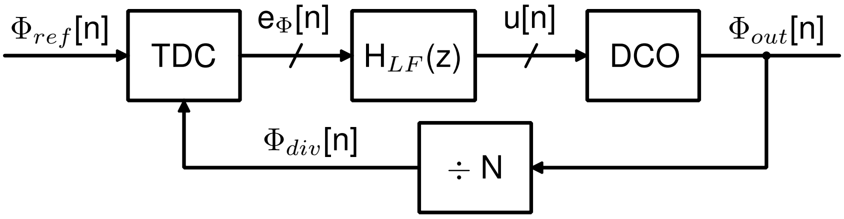


Figure 5: Basic ADPLL.

The basic architecture of an ADPLL is shown in figure 5. Here, compared to the continuous PLL, the phase detector has been replaced with a time to digital converter (TDC), the loop filter $\mathrm{H}_{LF}(s)$ with a discrete loop filter $\mathrm{H}_{LF}(z)$, and the VCO with a digitally controlled oscillator (DCO). In this architecture, the TDC, loop filter, and DCO are digital.

2.2.1 Divider

A digital divider behaves nearly identical to the continuous case:

$$\Phi_{div}[n] = \frac{\Phi_{out}[n]}{\mathrm{N}} \tag{25}$$

Application of the z-transform:

$$\frac{\Phi_{div}(z)}{\Phi_{out}(z)} = \frac{1}{\mathrm{N}} \tag{26}$$

2.2.2 TDC

The TDC is a digital, quantized representation of the the phase detector. It takes input phase signals Φ_{div}[n] and Φ_{ref}[n], and outputs a digital phase error measurement word $e_\Phi[n]$. Figure 6 shows the basic TDC model architecture. A TDC will have limited resolution in phase, equivalent to M steps per reference cycle. This is a minimum step size in time of Δt_{step} = $1/Mf_{ref}$. Since the output of the TDC is digital, the model applies a scale factor $\mathrm{M}/2\pi$ and floor rounding, so 1 LSB of $e_\Phi[n]$ equates to Δt_{step} timing error between Φ_{div}[n] and Φ_{ref}[n].

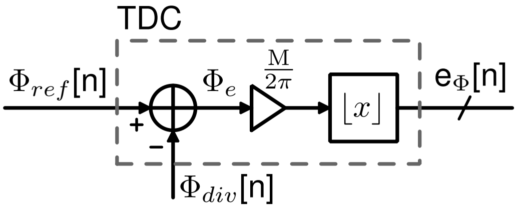


Figure 6: TDC model.

$$e_\Phi[n] = \left\lfloor \frac{\mathrm{M}}{2\pi}(\Phi_{ref}[n] - \Phi_{div}[n]) \right\rfloor \tag{27}$$

For purposes of PLL transfer function calculation, the z-domain representation is as follows. Effects of quantization will be handled in section 2.3.

$$e_\Phi(z) = \frac{\mathrm{M}}{2\pi}(\Phi_{ref}(z) - \Phi_{div}(z)) \tag{28}$$

2.2.3 Loop Filter

The loop filter design will be derived from the continuous PI-controller loop filter with optional peaking compensation (equation 22) via application of the bilinear transform. The bilinear transform specifically allows for the conversion of a continuous transfer function to discrete representation, and vice versa. This, however is conditioned on satisfaction of Nyquist sampling criteria, and in the case of PLLs it is recommended that $f_s > 10 \cdot \mathrm{BW}_{loop}$ to ensure transformation accuracy [1]. A high level of oversampling allows for the following definition of the bilinear transform, where 1/T=f_{ref}:

$$\begin{aligned} z^{-1} &= e^{-sT} && \text{(definition of z-space)} \\ &= \sum_{k=0}^{\infty} \frac{(-sT)^k}{k!} && \text{(exponential Taylor series)} \\ &\approx 1 - sT && (\text{if } |sT| = 2\pi \mathrm{BW}_{loop} \cdot T << 1) \end{aligned}$$

Thus the bilinear transform identities are:

$$z^{-1} = 1 - sT \tag{29}$$

$$s = \frac{1}{T}(1 - z^{-1}) \tag{30}$$

Applying 30 to equation 22 yields the z-domain loop filter:

$$\mathrm{H}_{LF}(z) = \mathrm{H}_{LF}(s)\Big|_{s=\frac{1}{T}(1-z^{-1})} = \frac{K_i}{s}\frac{\left(\frac{s}{\omega_z}+1\right)}{\left(\frac{s}{\omega_p}+1\right)}\Bigg|_{s=\frac{1}{T}(1-z^{-1})} \tag{31}$$

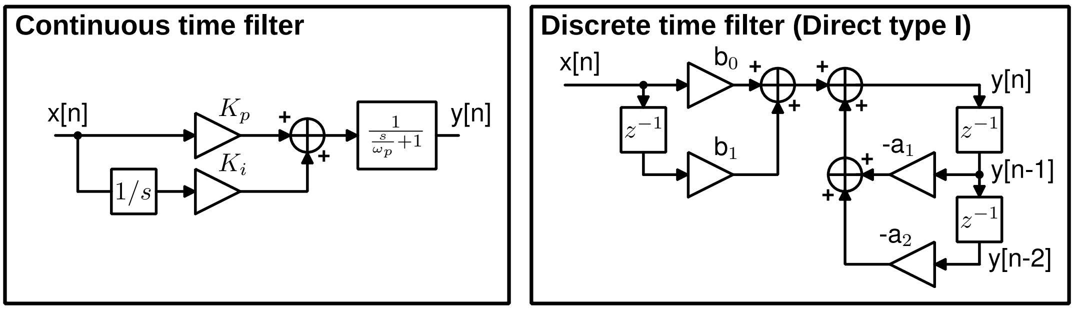
$$= \frac{\omega_p}{\omega_z}\frac{k_i T}{(1-z^{-1})}\frac{(1+\omega_z T) - z^{-1}}{(1+\omega_p T) - z^{-1}(2+\omega_p T) + z^{-2}} \tag{32}$$

Equation 43 is converted to a digitally implementable representation via converting into the canonical representation of 33, which determines the tap coefficients for the sampled-time difference equation 34.

$$\mathrm{H}_{LF}(z) = \frac{\sum_{j=0}^{M} b_j z^{-j}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{33}$$

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{j=0}^{M} b_j x[n-j] \tag{34}$$

Equation 34 is directly implementable in digital hardware with a direct type 1 IIR filter shown in figure 7, with the filter coefficients given by equations 35-38. The filter coefficients must be quantized into finite resolution fixed point words for a complete digital implementation. Effects of quantization will be discussed in section 2.3.


Continuous time filter

Discrete time filter (Direct type I)

Figure 7: Implementation of filter.

$$a_1 = -\frac{2+\omega_p T}{1+\omega_p T} \tag{35}$$

$$a_2 = \frac{1}{1+\omega_p T} \tag{36}$$

$$b_0 = \frac{K_i \omega_p T}{\omega_z} \frac{1+\omega_z T}{1+\omega_p T} \tag{37}$$

$$b_1 = \frac{K_i \omega_p T}{\omega_z} \frac{1}{1+\omega_p T} \tag{38}$$

2.2.4 DCO

The DCO is modeled in discrete time as a recursive phase integrator, dependent on the DCO gain K_{DCO}, which provides the frequency tuning per LSB of the oscillator, the input oscillator tuning word u[n], and the sampling period T.

$$\Phi_{out}[n] = \Phi_{out}[n-1] + 2\pi K_{DCO} u[n] T \tag{39}$$

Application of the z-transform yields:

$$\frac{\Phi_{out}(z)}{u(z)} = \frac{2\pi K_{DCO} T}{1-z^{-1}} \tag{40}$$

Application of the bilinear transform to the DCO transfer function yields:

$$\frac{\Phi_{out}(s)}{u(s)} = \frac{2\pi K_{DCO} T}{1-(1-sT)} = \frac{2\pi K_{DCO}}{s} \tag{41}$$

2.2.5 Discrete-time PLL transfer function

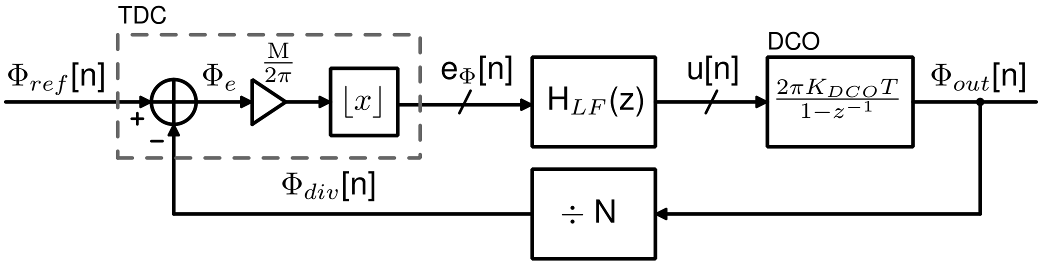


Figure 8: Discrete time PLL model.

The transfer function for the discrete-time PLL can be computed in the z-domain, and also approximated continuously. The open loop z-domain transfer function is:

$$\mathrm{OL}(z) = \mathrm{H}_{TDC}(z)\mathrm{H}_{LF}(z)\mathrm{H}_{DCO}(z)\mathrm{H}_{DIV}(z) = \tag{42}$$

$$2\pi K_{DCO} K_i T^2 \frac{\mathrm{M}}{\mathrm{N}} \frac{\omega_p}{\omega_z} \frac{(1+\omega_z T) - z^{-1}}{(1+\omega_p T) - z^{-1}(3+2\omega_p T) + z^{-2}(3+\omega_p T) - z^{-3}} \tag{43}$$

The closed loop z-domain PLL phase transfer function is:

$$\frac{\Phi_{out}(z)}{\Phi_{ref}(z)} = \frac{\mathrm{H}_{DIV}(z)^{-1}\mathrm{OL}(z)}{1+\mathrm{OL}(z)} = \tag{44}$$

$$\frac{2\pi K_{DCO}K_iT^2\mathrm{M}\frac{\omega_p}{\omega_z}(1+\omega_zT) - z^{-1}}{(1+\omega_pT+2\pi K_{DCO}K_iT^2\frac{\mathrm{M}}{\mathrm{N}}\frac{\omega_p}{\omega_z}(1+\omega_zT)) - z^{-1}(3+2\omega_pT+2\pi K_{DCO}K_iT^2\frac{\mathrm{M}}{\mathrm{N}}\frac{\omega_p}{\omega_z}) + z^{-2}(3+\omega_pT) - z^{-3}} \tag{45}$$

The s-domain approximation of the transfer function is:

$$\mathrm{OL}(s) = \mathrm{H}_{TDC}(s)\mathrm{H}_{LF}(s)\mathrm{H}_{DCO}(s)\mathrm{H}_{DIV}(s) = \frac{\mathrm{M}}{\mathrm{N}}\frac{K_{DCO}K_i}{s^2}\frac{\left(\frac{s}{\omega_z}+1\right)}{\left(\frac{s}{\omega_p}+1\right)} \tag{46}$$

And in closed loop configuration the PLL phase transfer function is:

$$\frac{\Phi_{out}(z)}{\Phi_{ref}(z)} = \frac{\mathrm{H}_{DIV}(z)^{-1}\mathrm{OL}(z)}{1+\mathrm{OL}(z)} = \frac{\mathrm{M}K_{DCO}K_i\left(\frac{s}{\omega_z}+1\right)}{s^3\frac{1}{\omega_z}+s^2+\frac{\mathrm{M}}{\mathrm{N}}K_{DCO}K_i\left(\frac{s}{\omega_z}+1\right)} \tag{47}$$

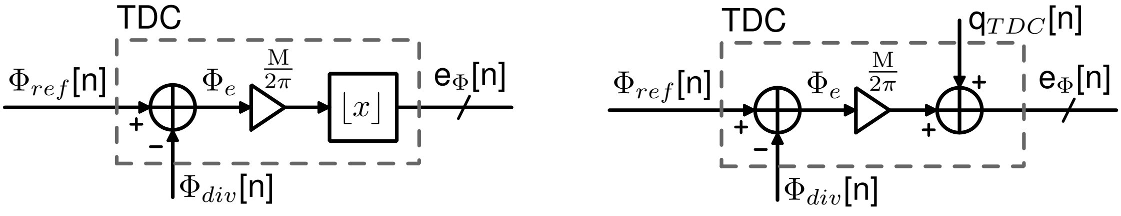
Incidentally, the s-domain approximation is significantly simpler and will is preferred in this paper for purposes of analysis.

2.3 PLL Noise

The predominant sources of noise in the discrete-time ADPLL are chiefly quantization (in the TDC, loop filter, and DCO), along with thermal noise (in the DCO, divider and TDC). The noise generated by these quantization sources will be discussed in the following sections.

2.3.1 TDC noise

The predominant phase noise source in the TDC is due to quantization. A straightforward approach to model quantization noise is to utilize the model of figure 9b to represent quantization.



(a) TDC Model.

(b) TDC additive noise model.

Using this model, the quantized signal e_Φ[n] is the sum of the its unquantized representation $\Phi_e\frac{\mathrm{M}}{2\pi}$ with a quantization error signal $\mathrm{q}_{TDC}[n]$. Figure 12 illustrates this process.

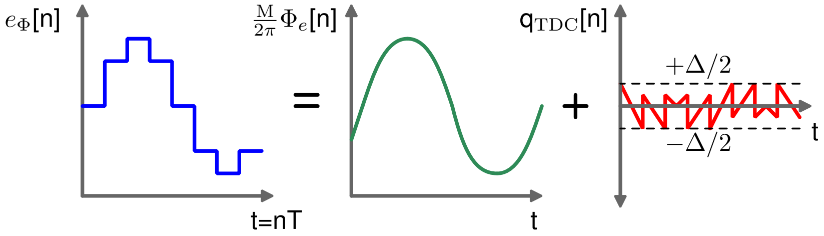

e_Φ[n]
t=nT
$\frac{M}{2\pi}\Phi_e$[n]
t
=
+
q_{TDC}[n]
$+\Delta/2$
$-\Delta/2$
t

Figure 10: Quantization as via additive error signal.

The quantization noise signal has the statistical property that it is uniformly distributed in the range $[-\Delta/2, \Delta/2]$, i.e. $P_q(Q = q) = \mathrm{U}(-\Delta/2, \Delta/2)$ if Δ is the quantization step size. The power of the TDC quantization noise signal is:

$$\sigma^2_{q_{TDC}} = \int_{-\infty}^{\infty} q^2 P_q(Q = q)dq = \int_{-\Delta/2}^{\Delta/2} \frac{q^2}{\Delta}dq = \frac{\Delta^2}{12} \tag{48}$$

Since e_Φ[n] is digital signal, the minimum step size is Δ=1 LSB. The TDC quantization noise power is therefore $\sigma^2_{q_{TDC}} = 1/12\,\mathrm{LSB}^2$. The po If the quantization noise is assumed to be white, and the TDC is sampled at f_{ref}, the quantization PSD is:

$$S_{q_{TDC}} = \frac{P_{q_{TDC}}}{\Delta f} = \frac{\sigma^2_{q_{TDC}}}{f_{ref}} = \frac{\Delta^2}{12 f_{ref}} = \frac{1}{12 f_{ref}} \quad \frac{[\mathrm{LSB}]^2}{[\mathrm{Hz}]} \tag{49}$$

2.3.2 DCO noise

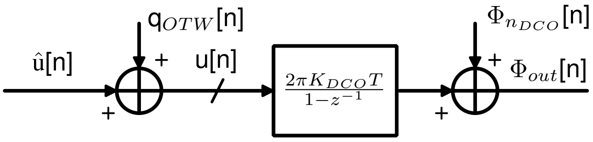

û[n]
q_{OTW}[n]
+
+
u[n]
$\frac{2\pi K_{DCO}T}{1-z^{-1}}$
$\Phi_{n_{DCO}}$[n]
+
+
Φ_{out}[n]

Figure 11: DCO additive noise model.

Noise in the DCO is from quantization of the oscillator tuning word (OTW), and from thermal and stochastic sources. In the digital PLL, the OTW is quantized in loop filter, so this source of noise will be analyzed in the loop filter section. Thus oscillator thermal noise will be considered.

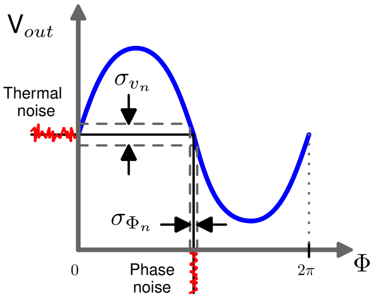

V$_{out}$
Thermal noise
σ_{v_n}
σ_{Φ_n}
0
Phase noise
2π
Φ

Figure 12: Thermal to phase noise conversion.

Noise from circuit and supply noise can be realized as an additive voltage component to the oscillator waveform $V_{osc}(t)$, with variance $\sigma^2_{V_n}$. As $V_{osc}(t)$ will have a finite derivative, a small additive noise signal $v_n(t)$ will be coupled into the time domain as a phase noise disturbance $\delta\Phi_n$ in the oscillator phase. At any point in time, assuming Gaussian white noise, $\delta\Phi_n$ is sampled from the following distribution:

$$\delta\Phi_n \sim \text{Norm}\left(\mu = 0, \sigma = \left(\frac{dV(\Phi)}{d\Phi}\right)^{-1} \sigma_{v_n}\right)\Bigg|_{\Phi=\omega_{osc}t} \tag{50}$$

The phase disturbance $\delta\Phi_n(t)$ at t is independent of all other points in time. If the phase of the oscillator is analyzed with discrete step T:

$$\Phi_{out}[n] = \Phi_{out}[n-1] + \omega_{osc}T + \delta\Phi_n[\Phi = n\omega_{osc}T] \tag{51}$$

The contribution only due to phase disturbance, i.e. phase noise Φ_n[n]:

$$\Phi_n[n] = \Phi_{out}[n] - \omega_{osc}T = \Phi_{out}[n-1] + \delta\Phi_n[n|\Phi = n\omega_{osc}T] \tag{52}$$

Notice this takes the form of a random walk signal. Computing the z-transform

$$\Phi_n(z) = \frac{\delta\Phi_n(z)}{1 - z^{-1}} \tag{53}$$

Application of the bilinear transform can be used to approximate the continuous noise spectrum:

$$\Phi_n(z)|_{z=1-sT} = \frac{\delta\Phi_n(z)}{1 - z^{-1}}\bigg|_{z=1-sT} = \frac{\delta\Phi_n(z = 1 - sT)}{sT} \tag{54}$$

The phase noise PSD is therefore:

$$S_{\Phi n_{DCO}}(\omega) = \left|\frac{\delta\Phi_n(z = 1 - j\omega T)}{j\omega T}\right|^2 = \frac{1}{\omega^2}\left|\frac{\delta\Phi_n(z = 1 - j\omega T)}{T}\right|^2 \tag{55}$$

Following that the phase disturbance signal $\delta\Phi_n(t)$ is white spectrum, a value for its power $S_{0\Phi n_{DCO}}$ can be defined as such:

$$S_{0\Phi n_{DCO}} = \lim_{T\to 0} \text{Var}\left(\left|\frac{\delta\Phi_n(z = 1 - j\omega T)}{T}\right|^2\right) \tag{56}$$

The value for $S_{0\Phi n_{DCO}}$ is highly dependent on implementation and is best extracted via simulation or physical measurement by fitting the following:

$$S_{\Phi n_{DCO}}(\omega) = \frac{S_{0\Phi n_{DCO}}}{\omega^2} \tag{57}$$

2.3.3 Divider noise

Divider noise is manifested as jitter on the the output signal, due to thermal noise coupling from voltage to phase. If the divider is a digital circuit, with edge rate dV/dt, and subject to thermal noise in the form of a voltage v_n, with noise power of $\sigma^2_{v_n}$, the phase noise (jitter) conferred is:

$$\sigma_{\Phi n_{DIV}} = \omega_{ref}\sigma_{tn_{DIV}} = \omega_{ref}\left(\frac{dV}{dt}\right)^{-1} \sigma_{v_n} \tag{58}$$

Thus the power is:

$$S_{\Phi n_{DIV}} = \omega^2_{ref}\sigma^2_{tn_{DIV}} = \omega^2_{ref}\left(\frac{dV}{dt}\right)^{-2} \sigma^2_{v_n} \tag{59}$$

2.3.4 Loop filter noise

Loop filter noise is principally from quantization in the digital case. In the digital loop filter, the mathematical operations are carried out with finite precision, and thus rounding errors occur. Here, an architecture with B bits in each fixed point word throughout the loop filter is assumed for analysis.

2.3.5 PLL output-referred noise

In terms of analysis, PLL noise referred to the PLL output is of most interest. In this case, the noise is defined in terms of phase noise Φ_n, or an unwanted added component to the oscillator phase signal $\Phi_{osc} = \omega_{osc}t$. The PLL output phase signal Φ_{out} is therefore:

$$\Phi_{out}(t) = \Phi_{osc}(t) + \Phi_n(t) = \omega_{osc}t + \Phi_n(t) \tag{60}$$

In analyzing phase noise, the noise power spectral density of the PLL output is of most interest. To compute this, an output voltage waveform of the PLL must defined. Given an amplitude A_0:

$$V_{out} = \Re\left(A_0 e^{j\Phi_{out}(t)}\right) = \Re\left(A_0 e^{j\omega_{osc}t} e^{j\Phi_n(t)}\right) \tag{61}$$

Assuming the phase noise signal is zero mean, $\mathbb{E}[\Phi_n(t)] = 0$, and the power of phase noise signal is small, $\mathrm{Var}[\Phi_n(t)] << 1$, then the approximation $e^{j\Phi_n(t)} = 1 + j\Phi_n(t)$ can be applied.

$$\begin{aligned} V_{out} = \Re\left(A_0 e^{j\omega_{osc}t} e^{j\Phi_n(t)}\right) &= \Re\left(A_0 e^{j\omega_{osc}t} + j\Phi_n(t) A_0 e^{j\omega_{osc}t}\right) & (62)\\ &= A_0 \cos(\omega_{osc}t) - \Phi_n(t) A_0 \sin(\omega_{osc}t) & (63) \end{aligned}$$

The result is a carrier cosine signal, and an orthogonal sine signal modulated by the phase noise Φ_n. From this, the spectral density of the phase noise relative to the carrier can be estimated. Power spectral density is computed as such, assuming the carrier and the phase noise signal components are uncorrelated:

$$S_{V_{out}} = |\mathcal{F}\{V_{out}(t)\}|^2 = A_0^2 |\mathcal{F}\{\cos(\omega_{osc}t)\}|^2 + A_0^2 |\Phi_n(\omega) * \mathcal{F}\{\sin(\omega_{osc}t)\}|^2 \tag{64}$$

If single side band (SSB) noise power spectral density $\mathcal{L}(\Delta f)$ is defined as the phase noise PSD at offset Δf from the carrier frequency f_{osc}, normalized to the carrier power:

$$\mathcal{L}(\Delta f) = |\Phi_n(2\pi\Delta f)|^2 = |\mathcal{F}\{\Phi_n(t)\}|^2 = S_{\Phi_n}(\Delta f) \tag{65}$$

Thus, the PLL output PSD noise relative to the carrier can be computed as the PSD of the phase noise signal $\Phi_n(t)$.

2.3.6 PLL noise sensitivity transfer functions

Having models for noise of generated by each PLL component, noise sensitivity transfer functions must be computer to refer each noise source to the PLL output in terms of phase. After which, the output noise PSD can be inferred based on the output phase signal, following the theory of section 2.3.5. An important assumption is made, in that every noise source within the PLL is independent, so that their individual noise powers can be summed at the PLL output to compute the overall PLL output noise.

and a method to calculate noise power spectral density from the output phase signal

All noise uncorrelated.

Plot of BW vs fasf Generic theory of PLL Discrete PLL Noise theory

3 Methods

Talk about how simulator is implemented: Discrete simulation models of phase noise, dco etc Filter optimization -phase noise and lock time estimate in frequency domain

Design recommendations: High sampling rate Minimum choice of TDC tesolution, constraints for divider jitter,

3.1 Behavioral, discrete event PLL simulation

3.1.1 Phase noise modeling

3.1.2 Measurement of phase noise, spurs

3.1.3 Monte-carlo sampling

Used to verify stability

3.2 Loop filter optimization

Reference phase noise does not matter, is always fixed. DCO and TDC phase noise should be highest. Will simulate with other noise sources, but framework will optimize loop filter design and provide recommendations for other parameters (max divider jitter) so DCO and TDC phase noise are dominant.

3.2.1 Estimation of settling time

Based on a continuous model of the PLL dynamics, an overall closed loop phase noise transfer funvtion G(f) is defined in the following form, where M < N. The transfer function is defined as a rational function of two polynomial functions of s.

$$G(f) = \frac{\Sigma_0^M b_m s^m}{\Sigma_0^N a_n s^n} \tag{66}$$

An estimate of the settling time of G(f) can be made from the eigenvalues of its representation in state space. This can be easily solved computation with available signal processing packages such as `scipy.signal`. Suppose the state transition matrix corresponding to G(f) is $\mathbf{\Phi}$, the set of k eigenvalues $\{\lambda_1, ..., \lambda_N\}$ for the system is found by first puting the system in the following state space representation form:

$$s\mathbf{Y}(s) = \mathbf{A}\mathbf{Y}(s) + \mathbf{B}X(s) \tag{67}$$

Then the state transition matrix is computed, and the eigenvalues of the aforementioned are found.

$$\mathbf{\Phi} = (s\mathbf{I} - A)^{-1} \tag{68}$$

$$|\mathbf{\Phi} - \lambda\mathbf{I}| = 0 \tag{69}$$

The step response of this system will take the form as a sum of complex exponentials, weighted by the eigenvectors $\{\mathbf{v}_1, ..., \mathbf{v}_N\}$ of $\mathbf{A}$.

$$\mathbf{y}(t) = \mathbf{v_1}e^{\lambda_1 t} + ... + \mathbf{v_k}e^{\lambda_k t}, \quad \mathbf{y(t)} = [y(t)\ \ y^{'}(t)\ \ ...\ \ y^{(k)}(t)]^T \tag{70}$$

The dynamics of the step response are governed by the exponential components of y(t). If $\{\lambda_1, ..., \lambda_N\} \in \mathbb{C}$ where $\lambda_k = \sigma_k + j\omega_K$, the real portion of each λ_k will describe the transient behavior. The long term settling of y(t) will be dominated by the λ_k with the smallest valued real component. This value is approximately the time constant for the system. Settling time t_s can be considered as the interval required for the signal to drop within a tolerance band $\pm\delta_{tol}$y(∞) about the final value y(∞).

$$t_s = \tau\ln(\delta_{tol}) = \frac{\ln(\delta_{tol})}{\min(|\Re(\{\lambda_1, ..., \lambda_k\})|)} \tag{71}$$

3.2.2 Estimation of PLL phase noise

It is assumed that the dominant output-referred phase noise contributions are due to the DCO thermal noise and the TDC quantization. If such is the case, total output integrated noise power is at a minimimum when the TDC and DCO contributions are approximately equal. Thus S_{TDC} and S_{DCO} are the PLL output-referred noise PSD respectively for the TDC and DCO noise sources. The total PLL output noise PSD $S_\Sigma(f)$ is (N is the PLL divider modulus):

$$S_\Sigma(f) = f_{clk} \cdot |2\pi N \cdot G(f)|^2 S_{TDC} + |1 - G(f)|^2 S_{DCO} \tag{72}$$

Given a bandwidth of interest Δf (i.e. baseband bandwidth for radio applications), the total integrated phase noise power is:

$$P_{\phi noise} = 2\int_0^{\Delta f} S_\Sigma(f)df \tag{73}$$

This can be computation solved for a grid of K values in the interval Δf, where each point represents a frequency bin f_{bin} = Δf/K. Therefore this estimate is implemented as such:

$$\hat{P}_{\phi noise} = 2\sum_{k=0}^{K-1} S_\Sigma(kf_{bin})f_{bin} \tag{74}$$

3.2.3 Optimization algorithm

Utilize BFGS optimization with constraints on settling time

4 Discussion

Discuss choices for loop filter optimizer (loop filter type) how was loop filter transfer function prototype arrived at? ie what is the best loop filter design

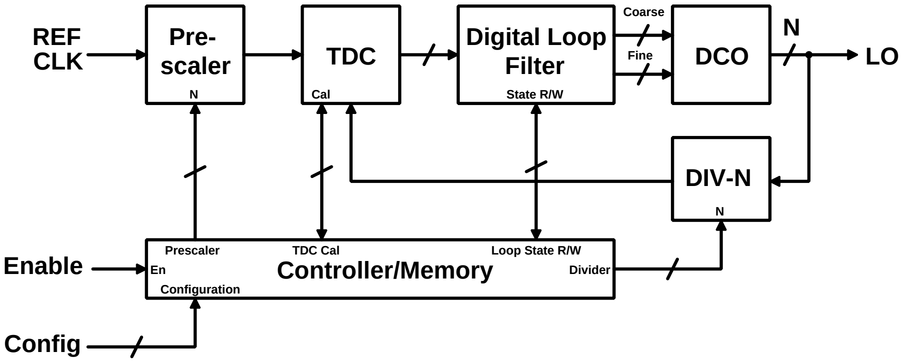
Compare to state of art (perrot) Perrot's pre-existing work: general purpuse simulation architecture, doesn't directly handle optimization for integer-N

Talk about why DCO flicker noise doesn't matter (i.e. Reference flicker is much greater) Also, we don't optimize for reference flicker, it will be constant no matter what, reference is separate from PLL and PLL reponse is flat N multiplier from ref to output.

5 Conclusion

Extend to fractional-N. Loop filter design will be the same, difference is divider model.

6 Block diagram


REF CLK
Pre-scaler
N
TDC
Cal
Digital Loop Filter
State R/W
Coarse
Fine
DCO
N
LO
DIV-N
N
Enable
Prescaler
TDC Cal
Loop State R/W
En
Controller/Memory
Divider
Configuration
Config

DCO	TDC	Divider	Other	SUM
70 μW	20 μW	10 μW	$<<1$ μW	100 μW

7 Specifications

Additionally: PLL output should support IQ sampling at LO frequency.

Parameter	Value	Unit	Notes
Frequency	2.4-2.4835	GHz	2.4G ISM Band
Ref. frequency	16	MHz	Yields 6 channels
Power	≤ 100	μW	
Residual FM	≤ 107	kHz$_{RMS}$	BER $\leq$ 1e-2, f_{dev}=±250 KHz
Initial Lock Time	≤ 50	μs	Upon cold start
Re-lock Time	≤ 5	μs	Coming out of standby
Bandwidth	100	kHz	(nominally), tunable

Table 1: System-level specifications

Parameter	Value	Unit	Notes
DCO LSB Resolution	≤ 50	kHz	Determined from quantization noise.
DCO DNL	< 1	LSB	Ensures monotonicity
TDC Resolution	≤ 3.8	ns	
TDC Resolution (bits)	≥ 4.03	bits	

Table 2: Component-level specifications.

8 Baseband phase noise

8.1 Modulated Signal

A generalized FSK/PSK modulated signal $x_s(t)$ can be written in the following with phase trajectory $\phi_s(t)$. $\phi_s(t)$ is composed of a modulation component $\phi_m(t)$ and carrier component $\omega_c t$.

$$x_s(t) = A_s \cos(\phi_s(t)) = A_s \cos(\phi_m(t) + \omega_c t) \tag{75}$$

8.2 Local oscillator - noisy PLL

A noisy PLL can be realized as $x_{lo}(t)$ with phase trajectory $\phi_{lo}(t)$. $\phi_{lo}(t)$ is comprised of a phase noise component $\phi_n(t)$ and an oscillation $\omega_{lo} t$.

$$x_{lo}(t) = A_{lo} \cos(\phi_{lo}(t)) = A_{lo} \cos(\phi_n(t) + \omega_{lo} t) \tag{76}$$

8.3 Baseband phase noise

The signal in the baseband, $x_{bb}(t)$, is given by mixing $x_s(t)$ with the LO $x_{lo}(t)$. For analysis of the baseband phase noise, zero-IF ($\omega_c = \omega_{lo}$) is considered. Thus:

$$x_{bb}(t) = A_s \cos(\phi_m(t) + \omega_c t) \cdot A_{lo} \cos(\phi_n(t) + \omega_{lo} t) = \tag{77}$$

$$\frac{1}{2} A_s A_{lo} \left[\cos(2\omega_{lo} t + \phi_m(t) + \phi_n(t)) + \cos(\phi_m(t) - \phi_n(t))\right] \tag{78}$$

The $2\omega_{lo}t$ component is assumed to be rejected due to limited mixer bandwidth. Thus the baseband signal is now:

$$x_{bb}(t) = A_s \cos(\phi_m(t) + \omega_c t) \cdot A_{lo} \cos(\phi_n(t) + \omega_{lo} t) = \tag{79}$$

$$\frac{1}{2} A_s A_{lo} \left[\cos(\phi_m(t) - \phi_n(t))\right] = \frac{1}{4} A_s A_{lo} \left[e^{j\phi_m(t)} e^{-j\phi_n(t)} + e^{-j\phi_m(t)} e^{j\phi_n(t)}\right] \tag{80}$$

The phase noise is presumed to be comprised of random phase and frequency walk, such that $\langle \phi_n(t) \rangle = 0$. Also, the phase noise is presumed to be small in amplitude, such that $\phi_n(t) << 1$, so a Taylor polynomial-based approximation of $e^{-j\phi_n(t)} = 1 - j\phi_n(t)$ can be made. Accordingly :

$$x_{bb}(t) \approx \frac{1}{4} A_s A_{lo} \left[e^{j\phi_m(t)}(1 - j\phi_n(t)) + e^{-j\phi_m(t)}(1 + j\phi_n(t))\right] \tag{81}$$

$$\frac{1}{4} A_s A_{lo} \left[\cos(\phi_m(t)) + \phi_n(t) \sin(\phi_m(t))\right] \tag{82}$$

The above can be treated as a signal component $m(t) = \cos(\phi_m(t))$, and a noise component $n(t) = \phi_n(t) \sin(\phi_m(t))$. The Fourier transform of the noise component can be considered:

$$N(f) = \mathcal{F}\{\phi_n(t) \sin(\phi_m(t))\} = \Phi_n(f) * \mathcal{F}\{-\cos(\phi_m(t) + \pi/2)\} \tag{83}$$

$$= -j\Phi_n(f) * \mathcal{F}\{\cos(\phi_m(t))\} \tag{84}$$

The signal and noise power spectral densities are therefore approximately:

$$S_{MM} = |M(f)|^2 = |\mathcal{F}\{\cos(\phi_m(t))\}|^2 \tag{85}$$

$$S_{NN} = |N(f)|^2 = |\Phi_n(f) * \mathcal{F}\{\cos(\phi_m(t))\}|^2 \tag{86}$$

The inband noise and signal power of these components can be easily computed via integration:

$$P_x = \int_{-\Delta f/2}^{+\Delta f/2} S_{XX} df \tag{87}$$

SNR is therefore the ratio of P_M/P_N within a bandwidth of interest Δf, which is straightforward to determine computationally for arbitrary PLL phase noise and modulation spectral densities.

9 DCO tuning

9.1 Backgate tuning

Tuning of a ring oscillator DCO through backgate terminal control will be considered. A general analysis of ring oscillator frequency will be made first to begin.

9.2 Ring oscillator frequency derivation

To analyze the oscillation frequency of a CMOS ring oscillator, an approximate model for a CMOS inverter will first be considered. A common model for delay in digital circuits [elmore delay model] is an RC circuit, where the MOSFET channels are approximated with an average conductance value $\langle g_{ch} \rangle$, and the output node is approximated to have a capacitance of C. With such a model, a ring oscillator would be assumed to have waveforms as decaying exponential, with time constant $\tau = \langle g_{ch} \rangle^{-1} C$, such as in Figure 13.

To calculate oscillation frequency ring oscillator from the RC model, several inferences are made:

- The switching point V_M of the inverters is $V_{DD}/2$, based on the assumption that the NMOS and PMOS are of equal strength.

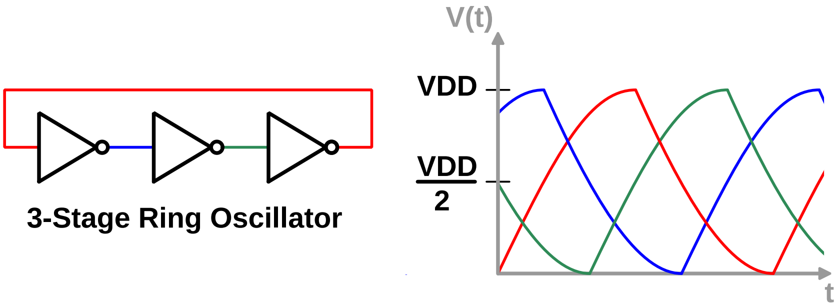
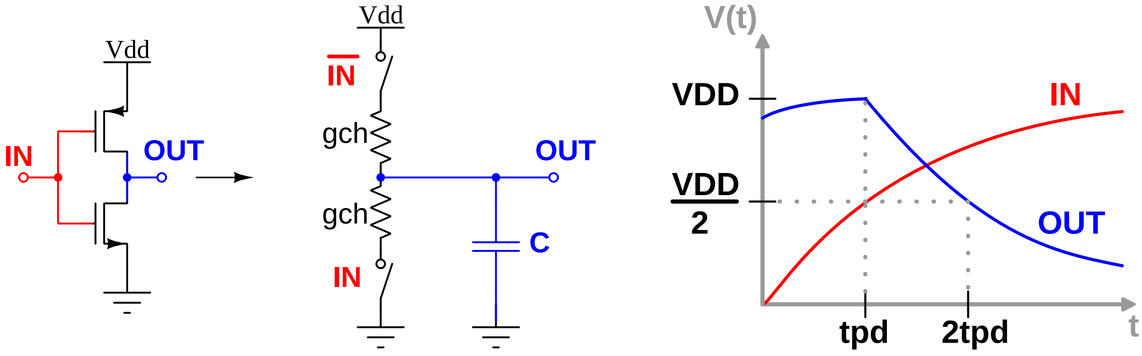


Figure 13: Model for ring oscillator.



(a) Inverter approximate model. **(b)** Inverter waveforms in ring oscillator.

Figure 14: Approximate model for ring oscillator inverter delay cell.

- The output of an inverter will have a decaying exponential which starts coincident with the passing of V_M at the input.
- The propagation delay t_{pd} for an inverter will be the time differential between the V_M crossing points on the input and output.
- The oscillator frequency will be $f_{osc} = 1/2Nt_{pd}$, where N is the number of stages (i.e. defined by 2N propagation delays).

Following the definition of V_M, it is trivial to find that $t_{pd} = \tau \ln 2$. It is therefore known that:

$$f_{osc}^{-1} = 2Nt_{pd} = \frac{2\ln(2)NC}{\langle g_{ch} \rangle} \tag{88}$$

9.2.1 Finding $\langle g_{ch} \rangle$ and C

The node capacitance C is trivial to find based on the inverter gate capacitance and a lumped load capacitance term C_L:

$$C = C_{ox}\left(W_N L_N + W_P L_N\right) + C_L \tag{89}$$

The average channel conductance $\langle g_{ch} \rangle$ is more involved to find. To do so, several assumptions are made:

- L >> L$_{min}$, so no velocity saturation, and therefore square law is applicable.
- NMOS and PMOS have equal V_t and transconducance.
- Output transition occur with the active FET in saturation during t_{pd}. This requires:
  - $V_{DD}/4 < V_t < V_{DD}/2$

Following those assumptions, $\langle g_{ch} \rangle$ can be computed via integral within the period t_{pd}:

$$\langle g_{ch} \rangle = \frac{1}{t_{pd}} \int_0^{t_{pd}} \frac{I_{out}(t)}{V_{out}(t)} dt \tag{90}$$

I_{out} is computed using the saturated MOSFET square law model an exponential waveforms assumptions. An I_{short} term is included to account for output current reduction from short-circuit conduction.

$$I_{out}(t) = \frac{k_n}{2}\left(\frac{W}{L}\right)_n \left[(V_{in}(t) - V_t)^2\right] - I_{short} = \frac{k_n}{2}\left(\frac{W}{L}\right)_n \left[\left(V_{DD}\left(1 - e^{-t/\tau}\right) - V_t\right)^2 - \left(\frac{V_{DD}}{2} - V_t\right)^2\right] \tag{91}$$

$k_n = \mu_n C_{ox}$, with the equal PMOS/NMOS assumption, $k_n\left(\frac{W}{L}\right)_n = k_p\left(\frac{W}{L}\right)_p$. V_{out} is simply a decaying exponential with a delay t_pd versus the input:

$$V_{out} = V_{DD} e^{-(t - t_{pd})/\tau} \tag{92}$$

Now, computing the integral for $\langle g_{ch} \rangle$ yields:

$$\langle g_{ch} \rangle = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_n \left[V_{DD}\left(\frac{7}{8\ln 2} - 1\right) - V_t\left(\frac{1}{\ln 2} - 1\right)\right] \tag{93}$$

As a simplification, α is defined as:

$$\alpha = \left[V_{DD}\left(\frac{7}{8\ln 2} - 1\right) - V_t\left(\frac{1}{\ln 2} - 1\right)\right] \tag{94}$$

9.2.2 Handling unequal NMOS/PMOS

In the case of different threshold voltages for NMOS and PMOS:

$$f_{osc}^{-1} = N(t_{pdn} + t_{pdp}) = \ln(2)NC\left(\frac{1}{\langle g_{ch} \rangle_n} + \frac{1}{\langle g_{ch} \rangle_p}\right) = \frac{2\ln(2)NC}{\langle g_{ch} \rangle'} \tag{95}$$

A modified $\langle g_{ch} \rangle'$ is defined:

$$\langle g_{ch} \rangle' = 2\left(\frac{1}{\langle g_{ch} \rangle_n} + \frac{1}{\langle g_{ch} \rangle_p}\right)^{-1} = 2\frac{\langle g_{ch} \rangle_n \langle g_{ch} \rangle_p}{\langle g_{ch} \rangle_n + \langle g_{ch} \rangle_p} = 2\frac{\frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_n \alpha_n \frac{1}{2}\mu_p C_{ox}\left(\frac{W}{L}\right)_p \alpha_p}{\frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_n \alpha_n + \frac{1}{2}\mu_p C_{ox}\left(\frac{W}{L}\right)_p \alpha_p} \tag{96}$$

This is somewhat unmanagable, hovever enforcing $\mu_n C_{ox}\left(\frac{W}{L}\right)_n = \mu_p C_{ox}\left(\frac{W}{L}\right)_p$ for V_M to equal $V_{DD}/2$ gives:

$$\langle g_{ch} \rangle' = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_n \frac{2\alpha_n \alpha_p}{\alpha_n + \alpha_p} = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_n \alpha' \tag{97}$$

Thus α_n and α_p are found for the according threshold voltages and then $\langle g_{ch} \rangle$ can be found.

$$\alpha' = \frac{2\alpha_n \alpha_p}{\alpha_n + \alpha_p} \tag{98}$$

9.2.3 Solving for oscillator frequency and power

Solving for oscillator frequency:

$$f_{osc} = \frac{\mu_n C_{ox}}{4\ln 2NC}\left(\frac{W}{L}\right)_n \left[V_{DD}\left(\frac{7}{8\ln 2} - 1\right) - V_t\left(\frac{1}{\ln 2} - 1\right)\right] \tag{99}$$

If gate capacitance is the dominant load component, and PMOS/NMOS are equal sized such that $C = 2WLC_{ox}$:

$$f_{osc} = \frac{\mu_n}{8\ln 2N} \cdot \frac{1}{L^2}\left[V_{DD}\left(\frac{7}{8\ln 2} - 1\right) - V_t\left(\frac{1}{\ln 2} - 1\right)\right] \tag{100}$$

Power can also be calculated, knowing in digital circuits $P = fC_{\Sigma}V_{DD}^2$, where C_{Σ} is the total active capacitance. Thus:

$$P_{osc} = Nf_{osc}CV_{DD}^2 = \frac{\mu_n C_{ox}}{4\ln 2}\left(\frac{W}{L}\right)_n \left[V_{DD}\left(\frac{7}{8\ln 2}-1\right) - V_t\left(\frac{1}{\ln 2}-1\right)\right] \tag{101}$$

It should be noted that the power consumption is proportional to FET aspect ratio (W/L).

9.3 Ring oscillator backgate tuning derivation

Using the basic expressions for ring oscillator frequency, the operature under backgate biasing can be found. In UTBB-FDSOI processes, the threshold voltage of a FET varies with linear dependence on the applied back gate bias V_{BG} (relative to source). Given the body effect coefficient of a process, γ, V_t is:

$$V_t = V_{t0} - \gamma V_{BG} \tag{102}$$

Using this in the ring oscillator frequency equation:

$$f_{osc} = \frac{\mu_n C_{ox}}{4\ln 2NC}\left(\frac{W}{L}\right)_n \left[V_{DD}\left(\frac{7}{8\ln 2}-1\right) - V_{t0}\left(\frac{1}{\ln 2}-1\right) + \gamma V_{BG}\left(\frac{1}{\ln 2}-1\right)\right] \tag{103}$$

Equivalently, $f_{osc} = f_{0,osc} + \Delta f_{osc}(V_{BG})$, where:

$$\Delta f_{osc}(V_{BG}) = \gamma V_{BG}\frac{\mu_n C_{ox}}{4\ln 2NC}\left(\frac{W}{L}\right)_n \left[\frac{1}{\ln 2}-1\right] \tag{104}$$

And $f_{0,osc}$ is the frequency with no backgate bias. If the backgate is swept from 0 to V_{DD}, and the node capacitance is increasingly varied (C0 to C3), Figure 15 is observed. Note that the change in frequency is linear with to backgate bias.

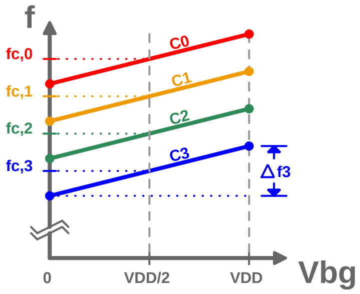


Figure 15: Backgate-tuned ring oscillator with coarse tuning capacitor bank.

If the backgate voltage is constrained in the range [0, V_{DD}], the center frequency f_c in the tuning range of the oscillator is then:

$$f_c = \frac{\mu_n C_{ox}}{4\ln 2NC}\left(\frac{W}{L}\right)_n \left[V_{DD}\left(\frac{7}{8\ln 2}-1+\frac{\gamma}{2\ln 2}-\frac{\gamma}{2}\right) - V_{t0}\left(\frac{1}{\ln 2}-1\right)\right] \tag{105}$$

The tuning range is also therefore:

$$\Delta f = \frac{\gamma V_{DD}}{2}\frac{\mu_n C_{ox}}{4\ln 2NC}\left(\frac{W}{L}\right)_n \left[\frac{1}{\ln 2}-1\right] \tag{106}$$

The fractional tuning range of the oscillator is:

$$\frac{\Delta f}{f_c} = \frac{1}{2}\cdot\frac{\gamma V_{DD}\left(1-\ln 2\right)}{V_{DD}\left(\frac{7}{8}-\ln 2+\frac{\gamma}{2}-\frac{\gamma}{2}\ln 2\right)-V_{t0}\left(1-\ln 2\right)} \tag{107}$$

If a N-bit DAC is used to control the oscillator, the resulting DCO gain is therefore:

$$K_{DCO} = \frac{\Delta f}{2^{N_{DAC}}} = \frac{f_c}{2^{N_{DAC}+1}}\cdot\frac{\gamma V_{DD}\left(1-\ln 2\right)}{V_{DD}\left(\frac{7}{8}-\ln 2+\frac{\gamma}{2}-\frac{\gamma}{2}\ln 2\right)-V_{t0}\left(1-\ln 2\right)} \tag{108}$$

9.4 DCO Gain Uncertainty

The DCO gain K_{DCO} is used in setting the loop filter coefficients, so the uncertainty of the DCO gain is of interest to allow for statistical analysis of the PLL across process variation. The uncertainty of K_{DCO} (normalized with nominal K_{DCO} value) as a function of V_{DD}, V_{t0} and γ is:

$$\sigma_{KDCO} = \sqrt{\left(\frac{\partial K_{DCO}}{\partial V_{DD}} \cdot \frac{\sigma_{VDD}}{K_{DCO}}\right)^2 + \left(\frac{\partial K_{DCO}}{\partial V_{t0}} \cdot \frac{\sigma_{Vt0}}{K_{DCO}}\right)^2 + \left(\frac{\partial K_{DCO}}{\partial \gamma} \cdot \frac{\sigma_{\gamma}}{K_{DCO}}\right)^2} \tag{109}$$

$$\frac{\partial K_{DCO}}{\partial V_{DD}} = \frac{f_c}{2^{N_{DAC}+1}} \cdot \frac{-\gamma V_{t0}(1-\ln 2)^2}{\left[V_{DD}\left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2}\ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]^2} \tag{110}$$

$$\frac{\partial K_{DCO}}{\partial V_{t0}} = \frac{f_c}{2^{N_{DAC}+1}} \cdot \frac{\gamma V_{DD}(1-\ln 2)^2}{\left[V_{DD}\left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2}\ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]^2} \tag{111}$$

$$\frac{\partial K_{DCO}}{\partial \gamma} = \frac{f_c}{2^{N_{DAC}+1}} \cdot \frac{V_{DD} \cdot (1-\ln 2)\left[V_{DD}\left(\frac{7}{8} - \ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]}{\left[V_{DD}\left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2}\ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]^2} \tag{112}$$

Simplified:

$$\sigma_{KDCO} = \frac{1}{\gamma V_{DD}\left[V_{DD}\left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2}\ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]} \cdot \sqrt{(\gamma V_{t0}(1-\ln 2)\sigma_{VDD})^2 + (\gamma V_{DD}(1-\ln 2)\sigma_{Vt0})^2 + \left(V_{DD}\left[V_{DD}\left(\frac{7}{8} - \ln 2\right) - V_{t0}\left(1 - \ln 2\right)\right]\sigma_{\gamma}\right)^2} \tag{113}$$

TODO - extract γ, V_{t0} variance for FETs in process kit.

References

[1] F. Gardner. “Charge-Pump Phase-Lock Loops”. In: *IEEE Transactions on Communications* 28.11 (Nov. 1980), pp. 1849–1858. DOI: `10.1109/tcom.1980.1094619`.

[2] Behzad Razavi. *Design of analog CMOS integrated circuits*. McGraw-Hill Education, 2017.

[3] Behzad Razavi. “Design of monolithic phase-locked loops and clock recovery circuitsDa tutorial”. In: 1996.

Week Number	Dates	Tasks	Outcomes
36	2.9 - 8.9	Review PLL Design	Refreshed Knowledge
37	9.9 - 15.9	Modeling/simulation (set up)	–
38	16.9 - 22.9	Modeling/simulation	TDC/DCO Requirements
39	23.9 - 29.9	Modeling/simulation	Loop Filter/Digital Algorithms
40	30.9 - 6.10	Modeling/simulation	**Loop filter,** Ideal implementation in Cadence
41	7.10 - 13.10	Circuit Research	DCO/Divider topologies
42	14.10 - 20.10	Circuit Research	TDC/other topologies
43	21.10 - 27.10	Circuit Implementation	Digital logic (schematic)
44	28.10 - 3.11	Circuit Implementation	DCO (schematic)
45	4.11 - 10.11	Circuit Implementation	Divider/other (schematic)
46	11.11 - 17.11	Circuit Implementation (TDC)	
47	18.11 - 24.11	Circuit Implementation (TDC)	TDC (schematic)
48	25.11 - 1.12	Full Circuit testing	Testbenches, find bugs, design fixes
49	2.12 - 8.12	Full Circuit testing	Design Fixes/iteration
50	9.12 - 15.12	–	–

Legend: **Done** **Current** **Revised**

A Appendix - Schedule

B Appendix - Code

```
######################################################################
# Simulation loop
######################################################################

t0 = time.clock()
for n in range(SAMPLES)[1:]:
    clk_out[n] = clk.update()
    tdc_out[n] = TDC_SCALE*((TDC_OFFSET+tdc.update(clk=clk_out[n-1], xin=div_out[n-1]))%TDC_STEPS)
    lf_out[n]  = lf.update(xin=tdc_out[n-1], clk=clk_out[n-1])
    osc_out[n] = dco.update(lf_out[n-1])
    div_out[n] = div.update(osc_out[n-1], DIV_N)
tdelta = time.clock() - t0
print("\nSimulation completed in%f s"%tdelta)
```

Listing 1: excode

C pres1

C.1 Motivation

- WSNs require ultra-low power circuits. A sensor should last for many years (>5) on a coin cell battery.
- With the below P_{avg} values, a CR2032 cell with 0.6 Wh capacity will last:
    - 1 μW → 70 years

  - 10 μW → 7 years
  - 100 μW → 0.7 years

- To save power, run devices with low duty cycle; activate remotely using wake up receiver (WuRx).
  - For 5 years of life, using 10% of battery energy for the WuRx, $P_{\mathrm{WURX}} < \mathbf{1.4}\ \mu\mathbf{W}$ on average is required.
- Main challenge for receiver is low power LO synthesis.
  - Synthesizer-free approaches (OOK receiver) lack robustness.
  - Advanced modulation schemes are too demanding with phase noise (PN).
  - Best option is FSK, using low power synthesizer with loose PN requirements to meet power target.

C.2 State of the Art

Performance averages based on sampling of 15 works published in IEEE:

Group	Count	Power [μW]	Freq [MHz]	RF BW [MHz]	Bitrate [kbps]	BER	Sens. [dBm]
All	15	96	1400	1.2	52	3e-3	-66
800/900M	6	138	887	0.95	15	4e-3	-68
All 2.4G	6	81	2400	2	60	2.8e-3	-67
2.4G FSK	2	190	2400	1.5	98	1e-3	-69
2.4G OOK	4	27	2400	2.5	41.5	4e-3	-66

- State of art of 2.4G WuRx:
  - Active Power $\leq 100\ \mu$W
  - RF BW $\approx$ 1 MHz
  - Sensitivity = -70 dBm (BER 1e-3)
  - Data Rate = 100 kbps

C.3 Objectives - Synthesizer goals

- Design ultra-low-power frequency synthesizer to meet requirements for *practical* wake up receivers.
  - **$\leq$1 μW average consumption**
  - Targeting **1% duty cycle** for **$\leq$100 μW active power**.
    - Fast locking to reduce on time/energy consumption.
- Synthesis range within **2.4 GHz ISM band**.
- Enable **wake up call detection within 1s** with 1e-3 miss rate.
  - Achievable with 250 kbps data, BER $\leq$ 1e-2, 1% duty, 20% wake up call Tx density.
  - Assuming 32-symbol wake up call used for false-alarm robustness.
  - High BER inherent due to high PN, expect many misses before sucess.
  - Use large FSK modulation index (m) to ease PN and power requirements.
    - m=2 → 2π phase shift per symbol or ±250 kHz frequency deviation.
    - RMS Residual frequency modulation (RFM) of PLL should be $<<$ symbol frequency deviation to achieve desired BER.
    - **RFM is derived from phase noise integration; use to constrain PLL PN.**

C.4 Architecture - concepts

- Utilize **Digital PLL**.
  - Inherent feedback helps with PVT variation (yield).
  - Calibration easy in digital design, agains helps with PVT variation.
  - Can store state when PLL idle, allowing for faster lock times coming out of standby.
- Utilize low duty cycle to achieve power target.

  - Can exploit semi-frequent calibration to improve performance.
  - Possibly can run PLL open loop when lock is achieved to save power.
- Utilize oscillator running at 1/N subharmonic of target frequency.
  - Use 2N phases in oscillator to achieve equivalent IQ sampling.
  - Avoids having to run oscillator at 2x target frequency as done typically.
  - 1/3 subharmonic operation should allow for dual mode 2.4G and 915M operation.
- Employ subsampling to reduce divider noise and TDC power?

C.5 State of Art - Sub-1 mW PLLS

Application is niche, so comparable PLLs hard to find. *Initial lock time and relock time

Type	P$_{PLL}$ [μW]	P$_{osc}$ [μW]	Freq [MHz]	PN@Δf [dBc/Hz]	t$_{lock}$* [μs]	Osc.	Ref Freq
Dig. Frac-N	650	304	2400	-110@0.5M	15/4	LC	26M
Ana. Int-N	680	510	2400	-110@1M	130/70	LC	1M
Ana. Int-N	128		500	-94@1M		Ring	31.25M
Ana. Int-N	570		800	-92.6@0.1M	200	LC	0.2M
Dig. Frac-N	250	173	2448		22/1	Ring	9M
Ana. Int-N	950		5500	-106@1M		IL-LC	

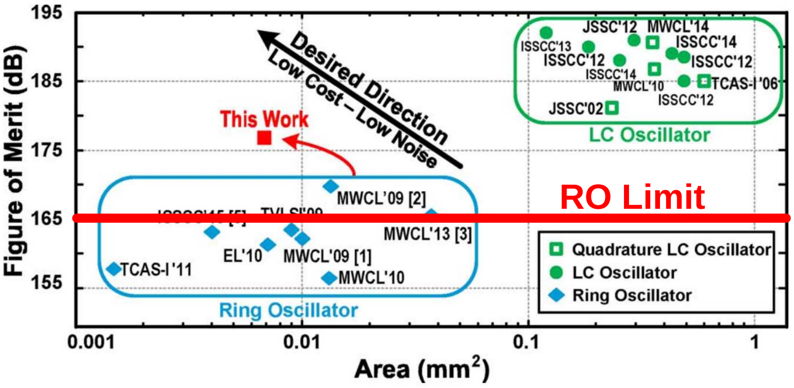
- Power limited by oscillator type. Scaling of LC-oscillator limited by gain requirements for self-starting. LC does not scale to low enough power.
- Current state of art for minimum power will be with ring oscillator, on the order of 200 μW for total PLL consumption.

C.6 Physical limits

Ring oscillator phase noise limit from ”Minimum Achievable Phase Noise of RC Oscillators”, Navid et al. 2005:

$$PN_{min}(\Delta f) = 10 \log 10 \left(\frac{7.33 k_B T}{P} \left(\frac{f_0}{\Delta f} \right)^2 \right) \tag{114}$$

If f_0 = 2.4 GHz, P = 50 μW, Δf= 1 MHz, T = 293 K, → **PN$_{min}$ = -84.7 dBc/Hz**
– This limit applied to the below FOM comparison (FOM PN=165 dB):



D pres2

D.1 Initial simulation/modeling approach

- Started writing Python code to simulate digital PLL.
  - Most familiar for me (I have modelled CDRs in Python before).
  - Simple to implement digital filters with Python Control Systems Library.

        - Also - Convenient stability analysis.
- Decided starting point:
    - Model DCO based on physical phase-noise limit of ring-oscillators.
    - Model loop filter as digital IIR second-order LPF (approximate second-order type-II analog PLL).
    - Ideal divider and TDC.
- Investigate (to determine component requirements):
    - Loop-BW to achieve PN requirements. Closed-loop sensitivity analysis:
        - TDC phase noise significant at f¡BW, determine limits for this.

- Investigate (continued):
    - Quantization, linearity impact of TDC & DCO on phase noise.
        - Determine number of bits required for each, and non-linearity limits.
    - Loop filter implementations (IIR vs FIR), order & type.
        - Evaluate stability, lock time, operating region to meet requirements.
    - Digital data path requirements to implement filter.
    - Algorithm/state machine for handling continuity of fine/coarse DCO ranging.
        - Previous two lead to Verilog description.
- **Final Outcomes:**
    - Full design constraints for PLL components, sufficient to allow schematic-level implementation.

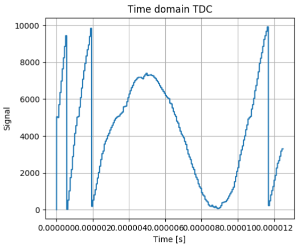
D.2 Implementation in Cadence

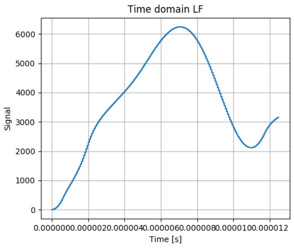
- When initial modeling/simulation satisfactory, translate design into a setup within Cadence.
    - Implement using Verilog-A, ahdlLib components?
- Will act as known-good test bench for testing transistor-level implementations of components in future design stages.
    - Replace element by element until fully functioning at transistor level.

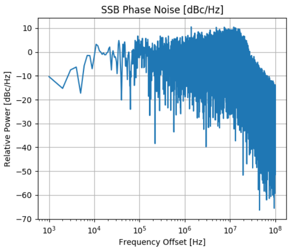
E pres4

E.1 Simulation approach

- My current full-PLL simulations are not generally stable.
    - Due to arbitrary choices for loop filter parameters not yielding a stable loop.
    - Next week plan is simulation / analysis / requirements definition of loop filter.
    - Will have to address issue of phase wrapping, fixed point resolution.





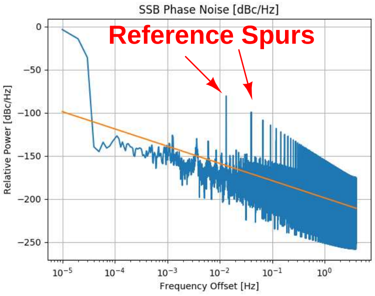


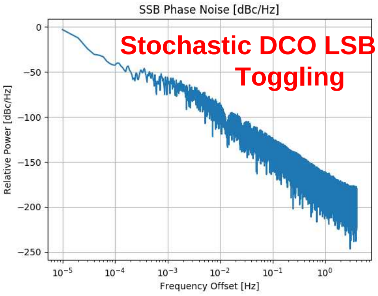
- Opted for mathematical-modelling of TDC, separate simulation of DCO.
    - Found straightforward model for TDC phase noise from Michael Perrott of MIT.
    - DCO simulation estimates quantization noise.
    - Will resume with full PLL simulation for loop filter and remainder of modeling.

E.2 DCO performance criteria

- Resolution set by frequency accuracy requirements, quantization noise.

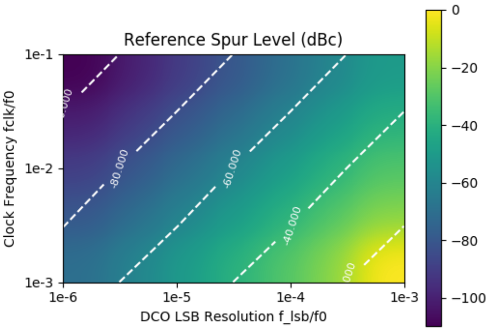
- Quantization noise is manifested here as:
  - Reference spurs resulting from deterministic components of signal.
  - A quasi-random noise signal when lock is achieved.
    - Results from stochastic toggling of $\sim$ 1 LSB of DCO tuning word to track low frequency variations.
    - Rolloff of -20 dB/decade at low frequency (same as ring oscillator), -40 dB/decade at high frequencies.

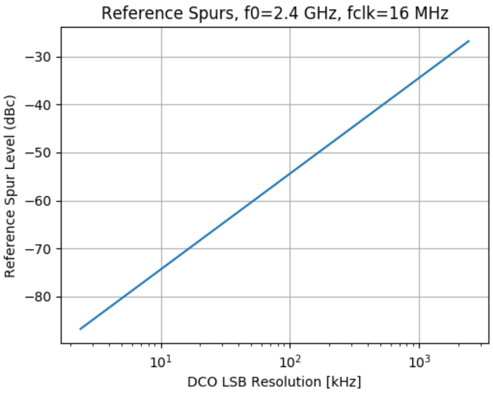




E.3 DCO - reference spur requirements

- Worst case reference spur level.
  - DCO tuning word toggling up/down 1 LSB every reference cycle.
- With f_0 = 2.4 GHz, f_{clk} = 16 MHz:
  - 52 kHz per LSB (i.e. K_{DCO}) is needed for a maximum -60 dBc reference spur level.



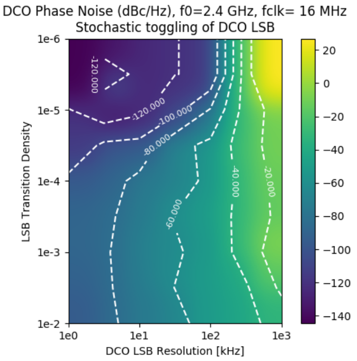


E.4 DCO - steady state cap noise

- In steady state, DCO tuning word will vary occassionaly by $\sim$ 1 LSB to track stochastic frequency changes.
  - This generates noise and should be less than the thermal phase noise of DCO.
  - Very pessimistic estimate here (assumes abrupt frequency change). Abrupt frequency steps contribute $1/\Delta f$ dependent component to phase noise.
- Theoretical ring oscillator phase noise limit from [2]:

$$PN_{min}(\Delta f) = 10 \log 10 \left(\frac{7.33 k_B T}{P} \left(\frac{f_0}{\Delta f} \right)^2 \right) \tag{115}$$

- If f_0 = 2.4 GHz, P = 50 μW, Δf= 1 MHz, T = 293K, $\rightarrow$ **PN ¡ -84.7 dBc/Hz** from this noise process.
  - LSB resolution of **50 kHz** seems feasible. Will have to verify with full PLL sim to account for loop dynamics.



E.5 DCO - linearity and accuracy

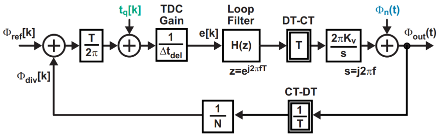
- Frequency accuracy.
  - Indeterminate IF assumed for wake up receivers, so accuracy not so critical.
  - If RF bandwidth of receiver ¿ PLL bandwidth, reasonable assumption is maximum frequency offset (accuracy) should be ¡ PLL bandwidth.
  - Current PLL bandwidth spec is 100 kHz, suggested 50 kHz LSB step from quantization noise analysis is sufficient?
    - This is assuming accuracy is tied to DCO resolution.
- Linearity:
  - Integral non-linearity over the tuning DCO range is not important, so no spec for INL is suggested. Should be locally linear (constrains DNL).
  - Monotonicity is essential, so must strictly have DNL ¡ 1 LSB.

E.6 TDC - phase noise model

- Based on a phase-domain model for PLL phase noise from Michael Perrot [1].

$$S_{\phi,out}(f) = f_{clk} \cdot |2\pi N G(f)|^2 \cdot \frac{\Delta t_{del}^2}{12} \tag{116}$$

$$G(f) = \frac{A(f)}{1 + A(f)} \tag{117}$$



- $S_{\phi,out}(f)$ is the TDC phase noise component, N is the divider modulus, G(f) is the closed loop PLL transfer function, A(f) is the open loop transfer function, Δt_{del} is the TDC time resolution.
- G(0) = 1, and G(f) ≈ 1 for f ∈ [0, f_{CBW}], where f_{CBW} is the closed loop bandwidth.

- Naive estimate for TDC time resolution:
  - Phase noise flat within closed-loop bandwidth. This component dominates power of the integrated phase noise a PLL with low TDC-resolution.
  - Use Residual frequency modulation equation and equation 2 (TDC phase noise) to estimate Δt_{del}. Integrate in f ∈ [0, f_{CBW}]

$$\Delta f_{RFM} = 2\int_{f_a}^{f_b} f^2 * PN(f)df \tag{118}$$

- With f_{clk} = 16 MHz, N = 150 (for 2.4 GHz synthesis), Δf_{RFM} ¡ 107 kHz to meet BER requirement, and f_{CBW} = 100 kHz. (PLL presumed to have very low TDC resolution)
  - Δt_{del} ¡ 3.8 ns
  - Phase noise of TDC below f_{CBW} is -47.7 dBc/Hz
  - Equates to minimum of 16.4 quantization steps for TDC (4.03 bits).
  - Assumptions of high phase noise and low resolution appear valid.

F pres5

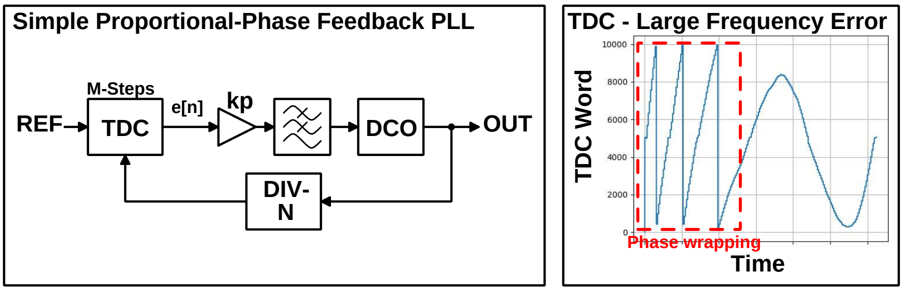
F.1 This week

- **Primary:** Loop analysis, requirements definition.
  - Design open loop filter design to meet closed loop requirements.
    - Dependent on DCO properties (k_{DCO}, f_0, tuning range).
    - Difference equation implementing discrete time 2nd order IIR filter.
  - Datapath requirements (fixed-point resolution)

F.2 Next week

- **Primary:** Ideal component PLL implementation in Cadence, continue loop filter work.
  - Ideal component PLL implementation is not a lot of work.
  - Loop filter very critical, spend more time on this.
    - Need to make Verilog description of loop filter for simulation in Cadence.

F.3 Loop filter original attempt

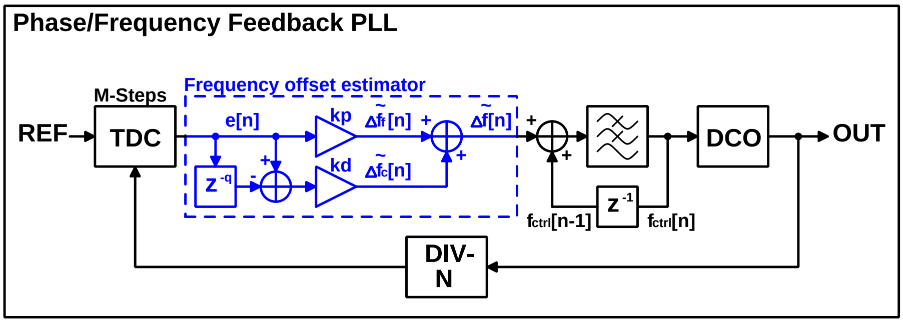

Simple Proportional-Phase Feedback PLL
M-Steps
REF
TDC
e[n]
kp
DCO
OUT
DIV-N
TDC - Large Frequency Error
TDC Word
10000
8000
6000
4000
2000
0
Phase wrapping
Time

- Started with simple PLL loop with proportional-phase fed into loop filter (LF).
  - Not stable at large frequency offset, due to frequency wrapping. At the TDC:

$$\Delta\phi = \frac{2\pi\Delta f T}{N} \rightarrow T_{wrap} = \frac{N}{\Delta f} \tag{119}$$

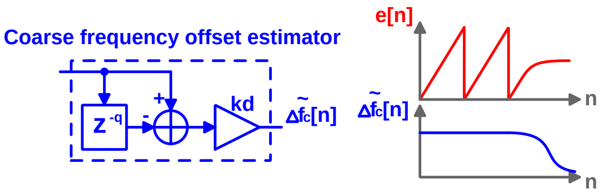
  - Upon cold start, Δf is expected to be up to 100 MHz, N=150 $\rightarrow$ T_{wrap} = 1.5 μs
    - $f_{wrap} \sim$ 600 kHz, this is unstable with a loop bandwidth of 100 kHz.
  - Must opt for alternate loop structure.

F.4 Loop filter new approach


Phase/Frequency Feedback PLL
M-Steps
Frequency offset estimator
REF
TDC
e[n]
kp
$\Delta\tilde{f}_f[n]$
+
$\Delta\tilde{f}[n]$
+
+
+
z^{-q}
-
+
kd
$\Delta\tilde{f}_c[n]$
+
DCO
OUT
z^{-1}
$f_{ctrl}[n-1]$
$f_{ctrl}[n]$
DIV-N

- Two-fold approach: utilize propotional-phase and coarse frequency offset estimation in feedback.
  - Coarse frequency estimator to handle high frequency offset (e.g. cold start-up).
  - Proportional-phase for near steady state. Acts like fine frequency estimator.
- Offset estimates are summed with previous oscillator tuning word (OTW, also f_{ctrl} here), then low passed filtered to yield new OTW.
  - Low pass filter loop keeps steady state.
  - Frequency estimator updates if any changes detected.

F.5 Coarse frequency estimation


Coarse frequency offset estimator
z^{-q}
+
-
kd
$\Delta\tilde{f}_c[n]$
e[n]
n
$\Delta\tilde{f}_c[n]$
n

- **Coarse frequency estimation:** Given M-step TDC, outputting phase error signal e$_\phi$[n], and a divider modulus N

$$\Delta\phi_{DCO}[n;q] = N \cdot \Delta\phi_{REF}[n] = 2\pi\frac{N}{M}\left(e_\phi[n] - e_\phi[n-q]\right), \quad \Delta\phi_{DCO}[n;q] = \Delta\omega_{DCO}[n]qT_{ref} = 2\pi q\frac{\Delta\tilde{f}_{DCO}}{f_{ref}} \tag{120}$$

$$\Delta\tilde{f}_c = \Delta\tilde{f}_{DCO} = \frac{f_{ref}}{q}\frac{N}{M}\left(e_\phi[n] - e_\phi[n-q]\right) \tag{121}$$

- Is a discrete differentiator, with gain coeficient to convert $d\phi/dt$ to frequency.

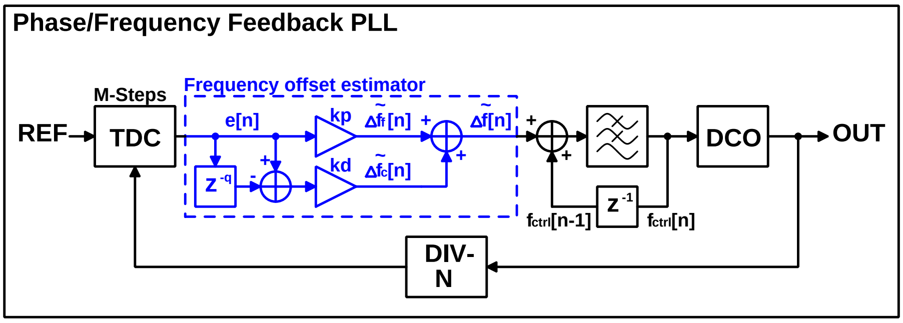
  - Design logic to handle phase wrapping.
  - Useful in coarse frequency range calibration. Can detect fast if frequency offset too large.
  - Delay q is used to increase frequency resolution.

- Given DCO gain K_{DCO}, the required gain K_d of the filter is:

$$K_d = \frac{f_{ref}}{qK_{DCO}} \frac{N}{M} \left(e_\phi[n] - e_\phi[n-q] \right) \tag{122}$$

- Disable the coarse estimator if $e_\phi[n] - e_\phi[n-q]$ ¡ some threshold:
  - Offset small enough, allow to run as classical phase-detector mode.

F.6 Loop filter Gain Coefficients



- When running in proportional-only feedback mode, the open loop gain $A(f)$ is:

$$A(f) = K_p \frac{M}{N} \frac{K_{LPF}}{s + \omega_{LPF} - K_{LPF}} \frac{2\pi K_{DCO}}{s} \tag{123}$$

- The closed loop gain is therefore (continuous):

$$G(f) = \frac{A(f)}{1 + A(f)} = \frac{2\pi M K_p K_{LPF} K_{DCO}/N}{s^2 + s(\omega_{LPF} - K_{LPF})/N + 2\pi M K_p K_{LPF} K_{DCO}/N} \tag{124}$$

- The form of a second order low pass filter is, with natural frequency ω_n and damping coefficient ζ:

$$H_{LPF}(f) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{125}$$

- Setting $H_{LPF}(f) = G(f)$, equivalencies for ω_n and ζ are found:

$$\omega_n = \sqrt{2\pi M K_p K_{LPF} K_{DCO}/N} \tag{126}$$

$$\zeta = \frac{\omega_{LPF} - K_{LPF}}{2\sqrt{2\pi M N K_p K_{LPF} K_{DCO}}} \tag{127}$$

- The Butterworth closed loop response with 100 kHz bandwidth, $\omega_n \sim 2\pi \cdot 100Khz, \zeta = 0.707$.
- Coefficients K_{LPF}, K_p, ω_n can be solved computationally.
- Need to reformulate in Z-domain.

F.7 Coarse frequency calibration

- Ring oscillator DCOs will have large variation in frequency due to PVT variation.
- Use bank of coarse capacitance values to correct range of oscillator.
- Coarse frequency estimator used in this calibration. DCO is has a fine range of Δf_{fine}.

Coarse frequency tuning state machine (as pseudo-code)

```
1. $C_{tune}$ = $C_{opt}$ = 0;    LE = $\Delta f_{fine}$
2. Reset PLL
3. Estimate Frequency offset → $\tilde{f}_{offset}$
4. If abs($\tilde{f}_{offset}$ - $\Delta f_{fine}/2$) ¡ LE:    // Centers fine tuning range around target frequency
     • LE = abs($\tilde{f}_{offset}$ - $\Delta f_{fine}/2$)
     • $C_{opt}$ = $C_{tune}$
5. If $C_{tune}$ == $C_{max}$: Goto 8
6. $C_{tune}$ += 1
7. Goto 2
8. $C_{tune}$ = $C_{opt}$; End
```