6.776
High Speed Communication Circuits
Lecture 23

Design of Fractional-N Frequency Synthesizers and Bandwidth Extension Techniques

Michael Perrott

Massachusetts Institute of Technology

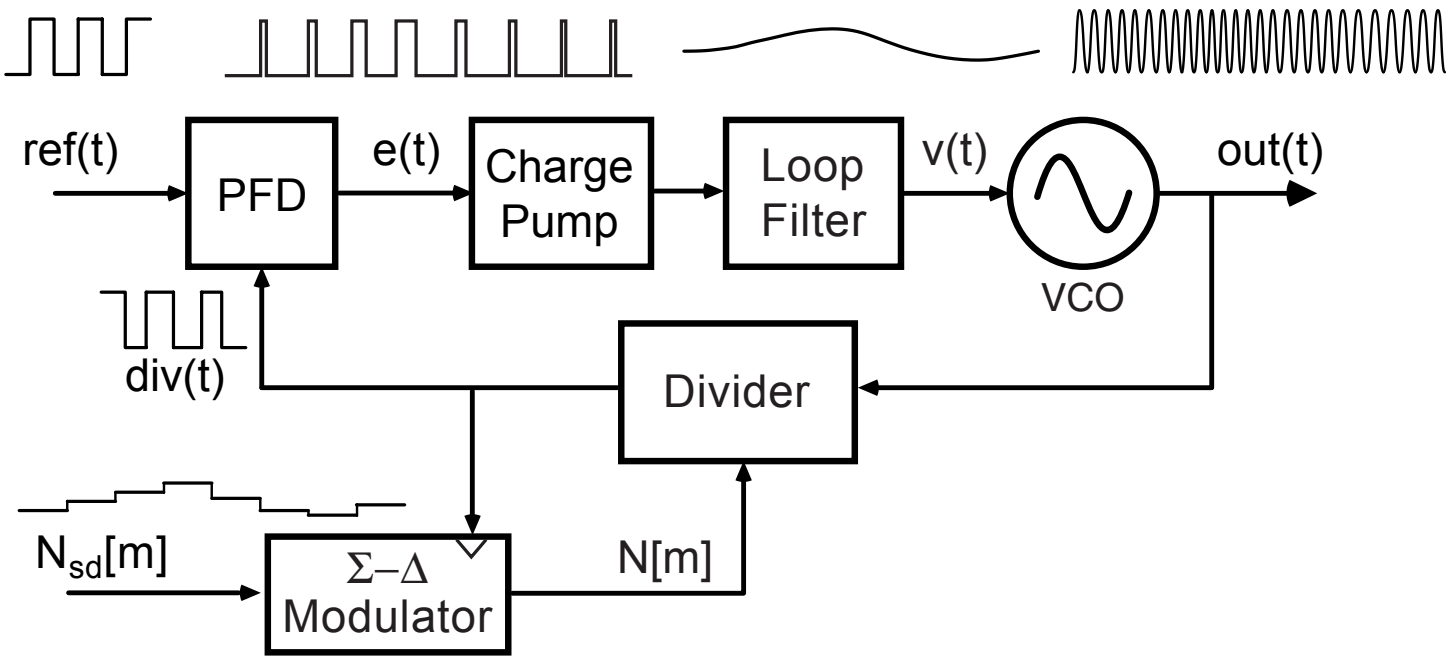
May 4, 2005

Outline of PLL Lectures

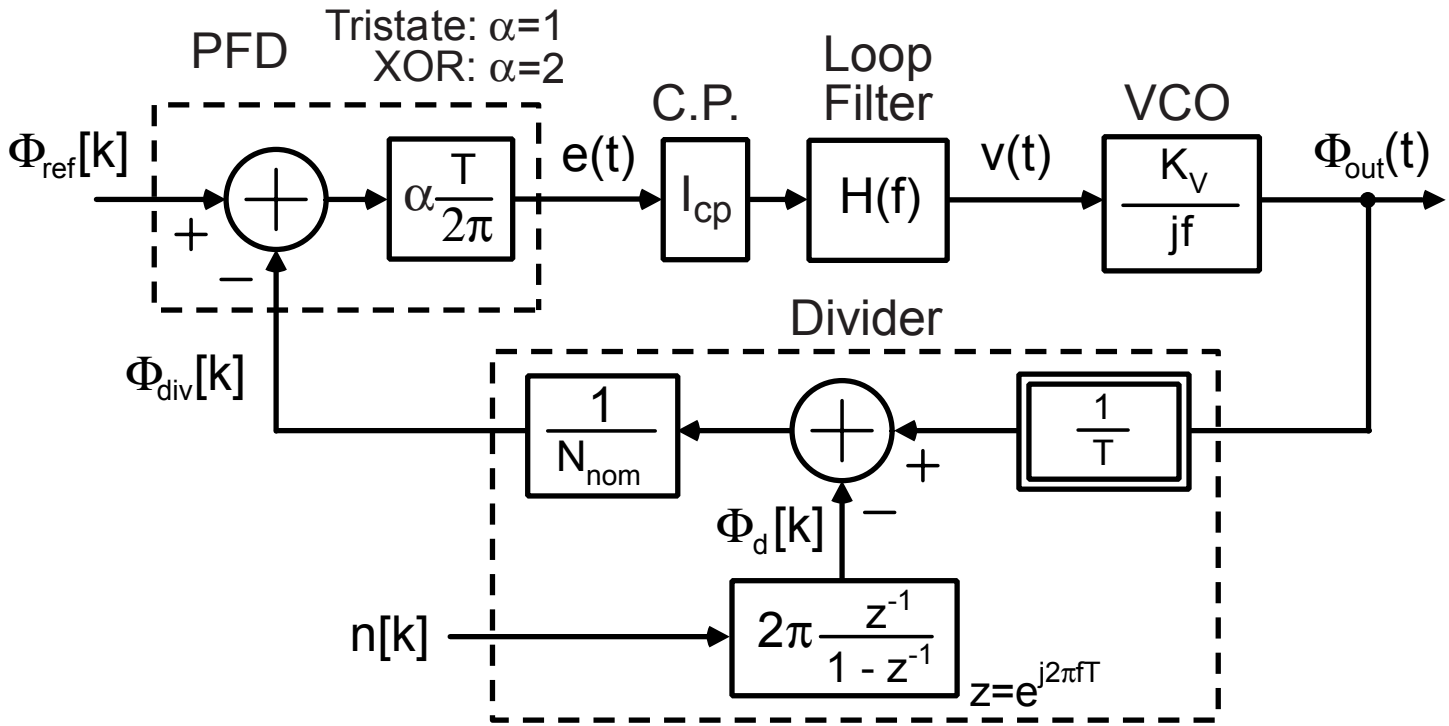
- **Integer-N Synthesizers**
  - **Basic blocks, modeling, and design**
  - **Frequency detection, PLL Type**
- **Noise in Integer-N and Fractional-N Synthesizers**
  - **Noise analysis of integer-N structure**
  - **Sigma-Delta modulators applied to fractional-N structures**
  - **Noise analysis of fractional-N structure**
- **Design of Fractional-N Frequency Synthesizers and Bandwidth Extension Techniques**
  - **PLL Design Assistant Software**
  - **Quantization noise reduction for improved bandwidth and noise**

Design of Frequency Synthesizers

- **Focus on fractional-N architecture since it is essentially a "super set" of other PLL synthesizers**
  - **If we can design this structure, we can also design classical integer-N systems**



Frequency-domain Model



Perrott et. al. *JSSC*, Aug. 2002

- **Closed loop dynamics parameterized by**

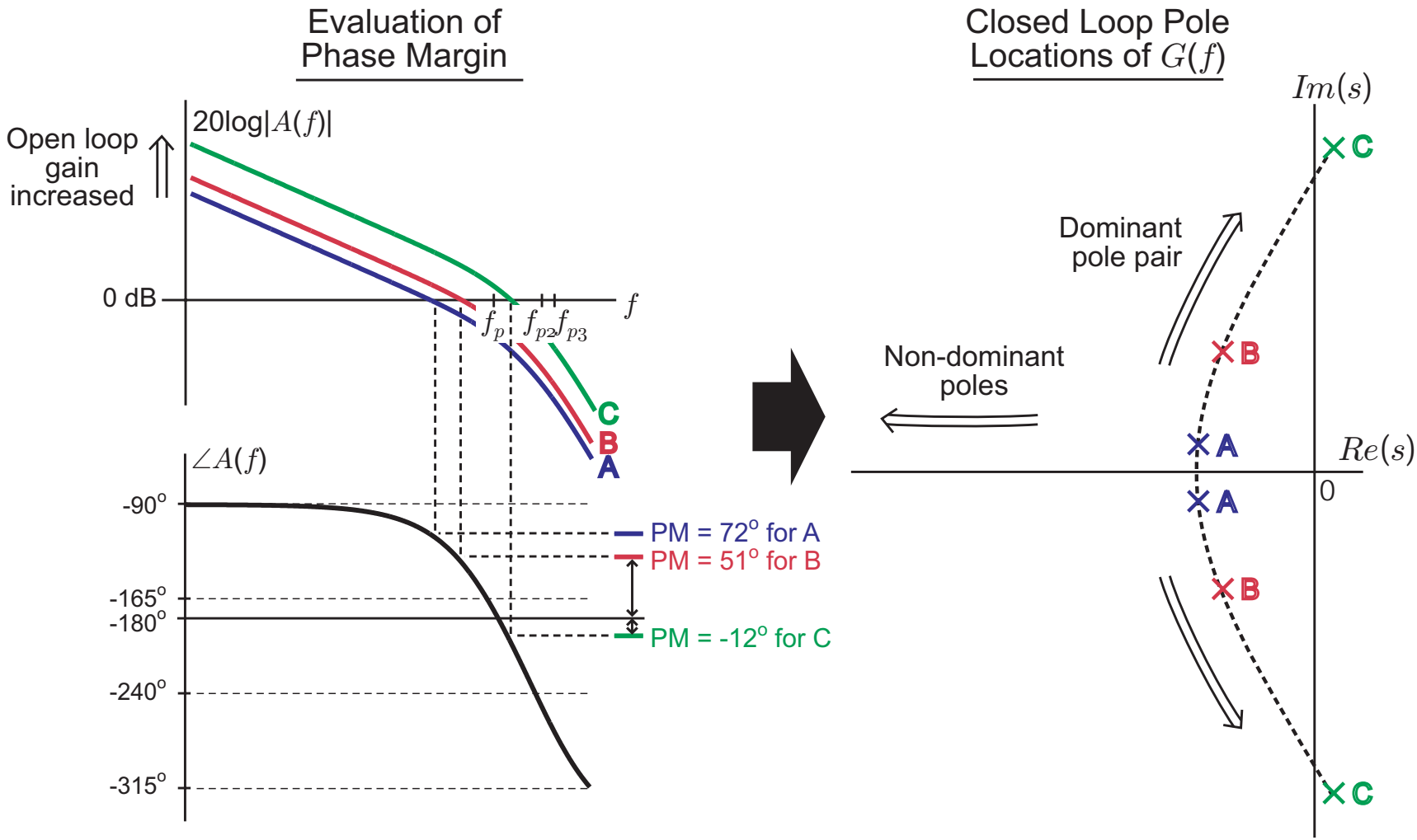
$$G(f) = \frac{A(f)}{1 + A(f)} \quad \text{where} \quad A(f) = \frac{\alpha I_{cp} H(f) K_V}{N_{nom} 2\pi j f}$$

Review of Classical Design Approach

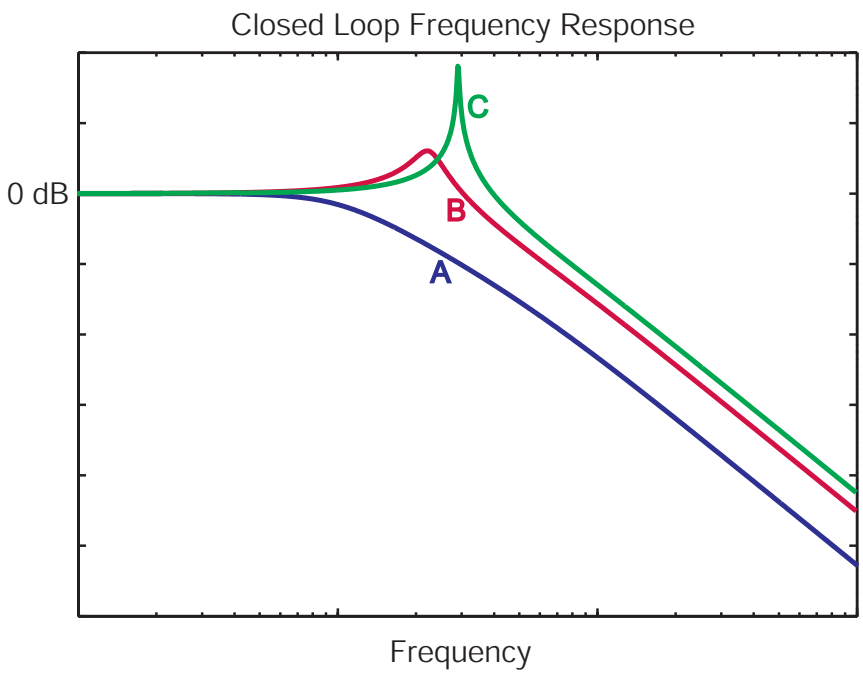
Given the desired closed-loop bandwidth, order, and system type:

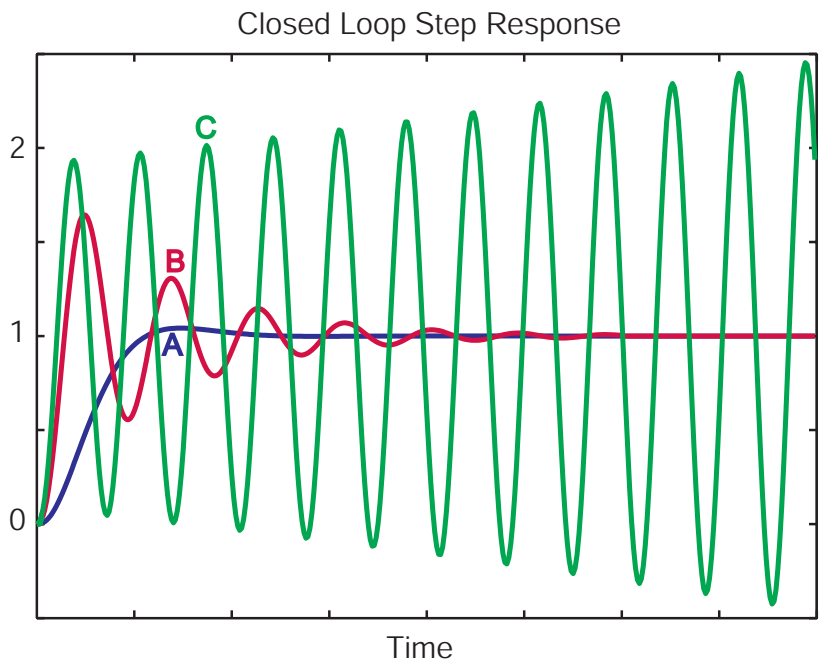
1. **Choose an appropriate topology for $H(f)$**
   - **Depends on order, type**
2. **Choose pole/zero values for $H(f)$ as appropriate for the required bandwidth**
3. **Adjust the open-loop gain to achieve the required bandwidth while maintaining stability**
   - **Plot gain and phase bode plots of $A(f)$**
   - **Use phase (or gain) margin criterion to infer stability**

Example: First Order, Type I with Parasitic Poles



First Order, Type I: Frequency and Step Responses





Limitations of Open Loop Design Approach

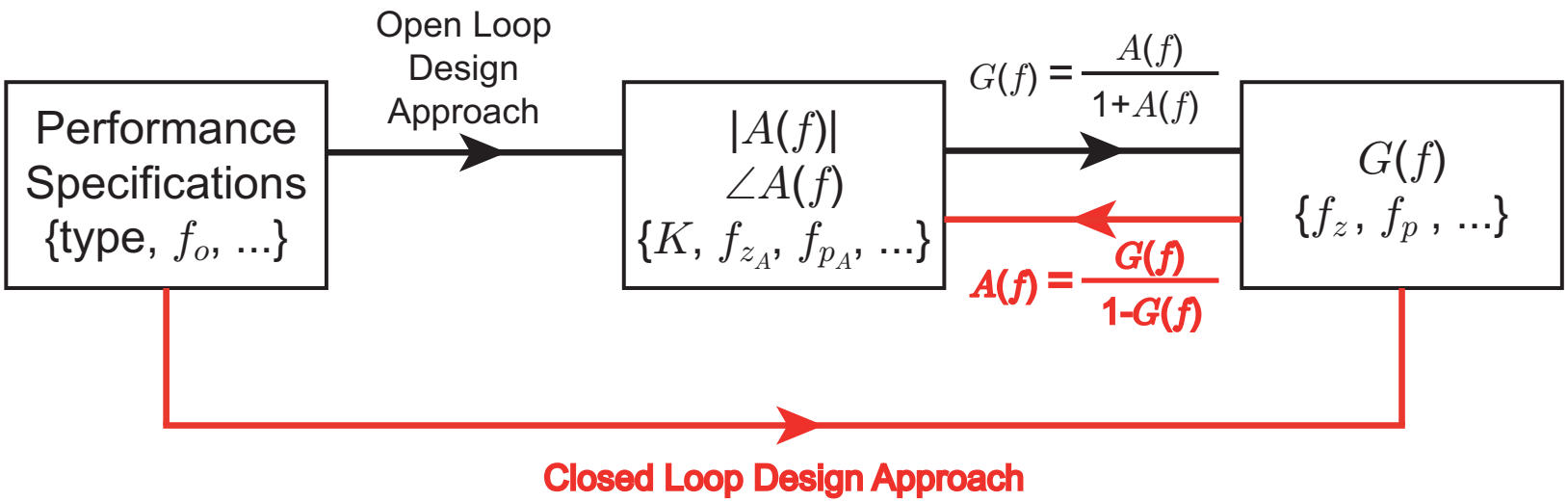
- **Constrained for applications which require precise filter response**
- **Complicated once parasitic poles are taken into account**
- **Poor control over filter shape**
- **Inadequate for systems with third order rolloff**
  - **Phase margin criterion based on second order systems**

Closed loop design approach:

Directly design G(f) by specifying dominant pole and zero locations on the s-plane (pole-zero diagram)

Closed Loop Design Approach: Overview

- $G(f)$ completely describes the closed loop dynamics
  - Design of this function is the ultimate goal

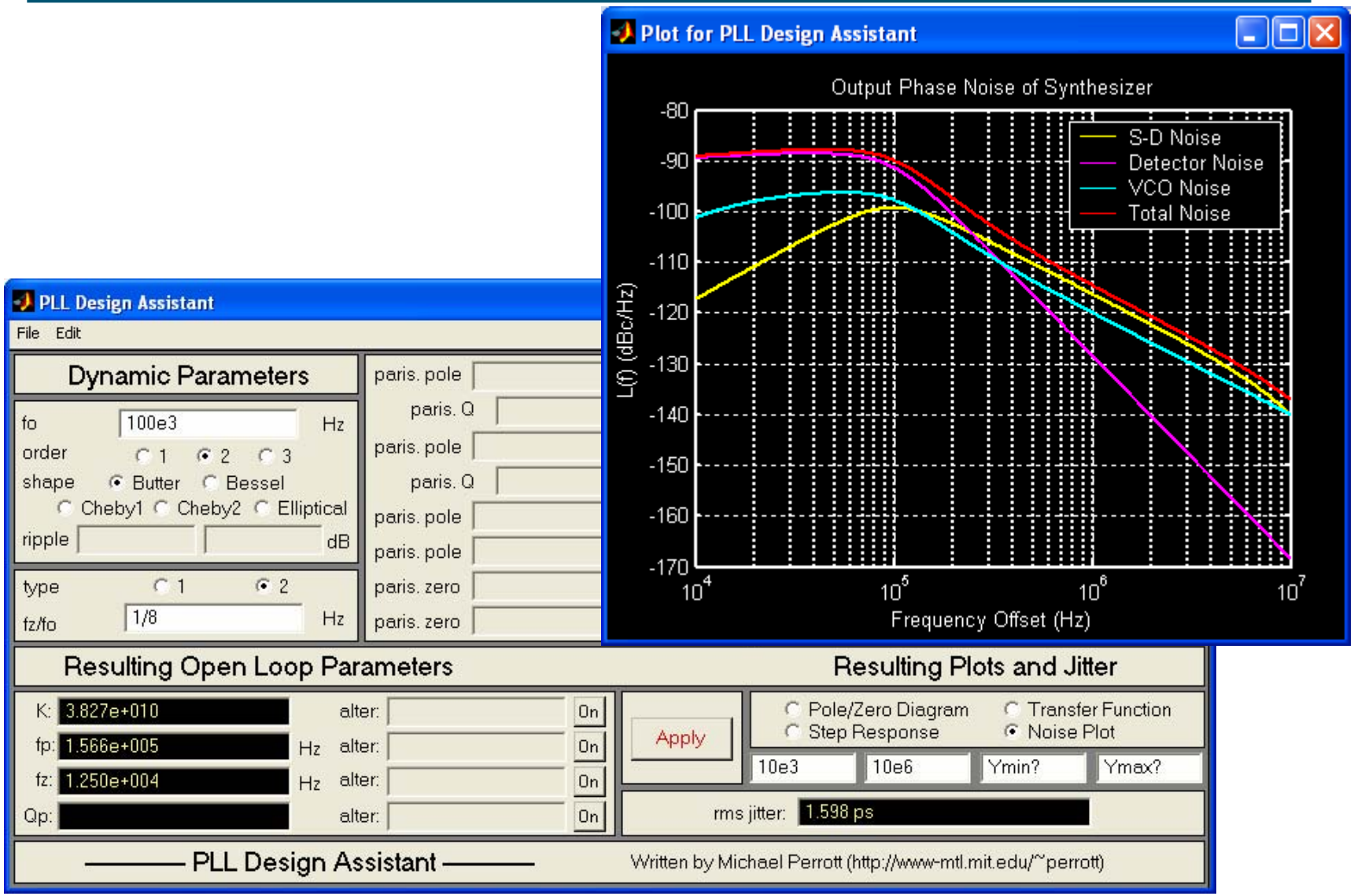


- Instead of indirectly designing $G(f)$ using plots of $A(f)$, solve for $G(f)$ directly as a function of specification parameters
- Solve for $A(f)$ that will achieve desired $G(f)$
- Account for the impact of parasitic poles/zeros

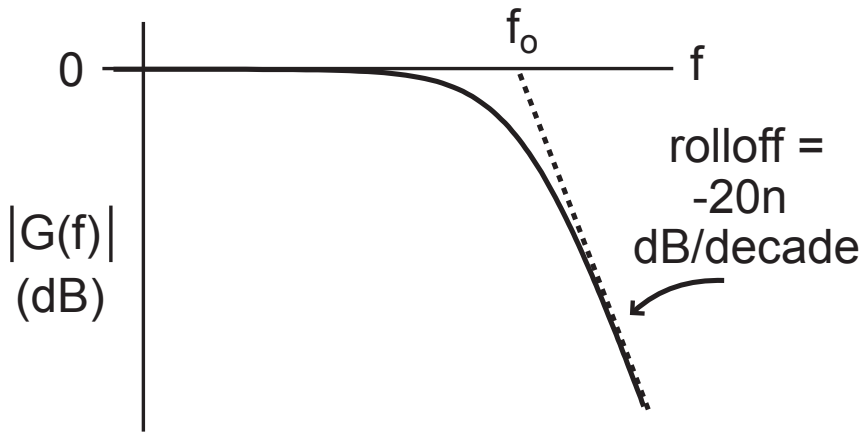
Closed Loop Design Approach: Implementation

- **Download PLL Design Assistant Software at http://www-mtl.mit.edu/research/perrottgroup/tools.html**
- **Read accompanying manual**
- **Algorithm described by C.Y. Lau et. al. in "Fractional-N Frequency Synthesizer Design at the Transfer Function Level Using a Direct Closed Loop Realization Algorithm", Design Automation Conference, 2003**

PLL Design Assistant



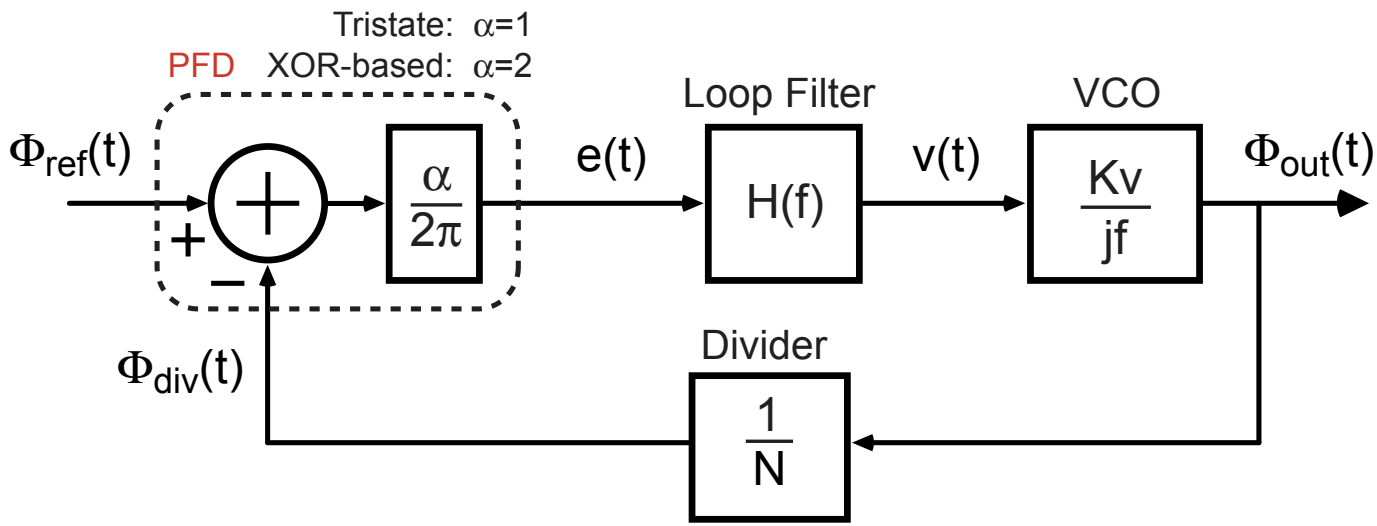
Definition of Bandwidth, Order, and Shape for G(f)



- **Bandwidth – f_o**
  - **Defined in asymptotic manner as shown**
- **Order – n**
  - **Defined according to the rolloff characteristic of G(f)**
- **Shape**
  - **Defined according to standard filter design methodologies**
    - Butterworth, Bessel, Chebyshev, etc.

Definition of Type

- **Type I: one integrator in PLL open loop transfer function**
  - **VCO adds on integrator**
  - **Loop filter, H(f), has no integrators**
- **Type II: two integrators in PLL open loop transfer function**
  - **Loop filter, H(f), has one integrator**



Loop Filter Transfer Function Vs Type and Order of G(f)

H(s) Topology For Different Type and Orders of G(f)

	Type I	Type II
Order 1	K_{LP}	$K_{LP}\frac{1+s/w_z}{s}$
Order 2	$\frac{K_{LP}}{1+s/w_p}$	$K_{LP}\frac{1+s/w_z}{s(1+s/w_p)}$
Order 3	$\frac{K_{LP}}{1+s/(w_pQ_p)+(s/w_p)^2}$	$\frac{K_{LP}(1+s/w_z)}{s(1+s/(w_pQ_p)+(s/w_p)^2)}$

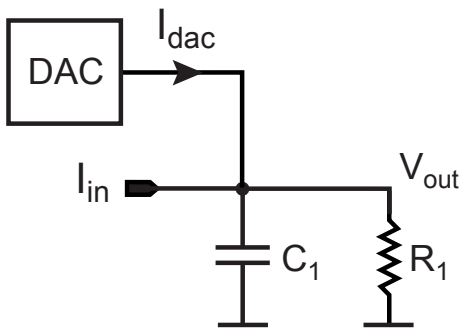
where $K_{LP} = K\frac{N_{nom}}{K_v I_{cp}\alpha}$

Calculated from software

- **Practical PLL implementations limited to above**
  - **Prohibitive analog complexity for higher order, type**
- **Open loop gain, K, will be calculated by algorithm**
  - **Loop filter gain related to open loop gain as shown above**

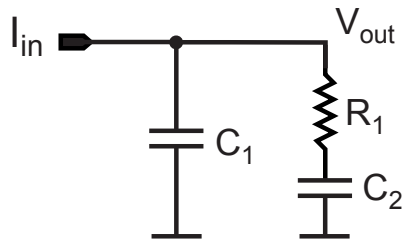
Passive Topologies to Realize a Second Order PLL

Type I, Order 2



$$\frac{V_{out}}{I_{in}} = \frac{R_1}{1+sR_1C_1}$$

Type II, Order 2

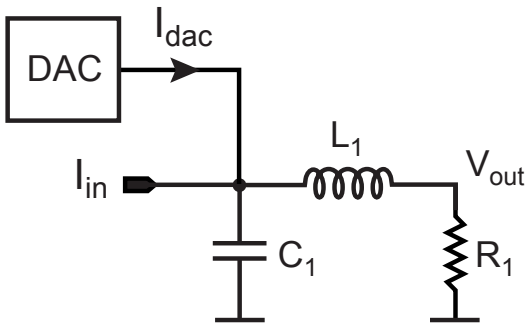


$$\frac{V_{out}}{I_{in}} = \frac{1}{s(C_1+C_2)} \frac{1+sR_1C_2}{1+sR_1C_{||}}$$

- **DAC is used for Type I implementation to coarsely tune VCO**
  - **Allows full range of VCO to be achieved**

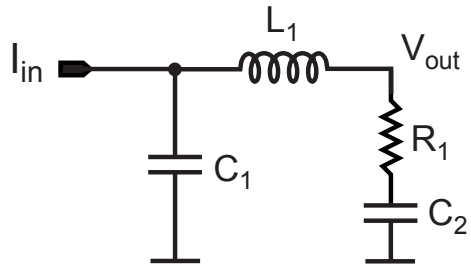
Passive Topologies to Realize a Third Order PLL

Type I, Order 3



$$\frac{V_{out}}{I_{in}} = \frac{R_1}{1+sR_1C_1+s^2L_1C_1}$$

Type II, Order 3



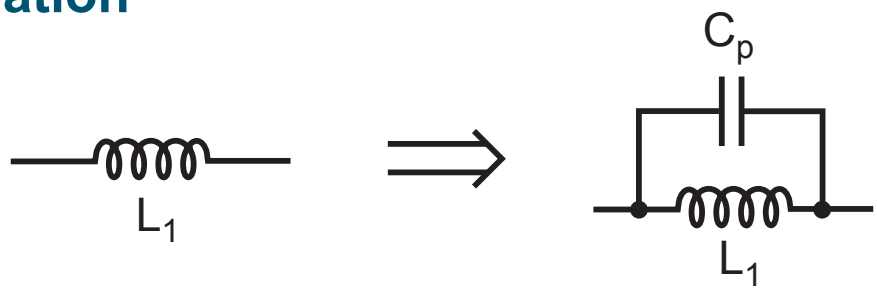
$$\frac{V_{out}}{I_{in}} = \frac{1}{s(C_1+C_2)} \cdot \frac{1+sR_1C_2}{1+sR_1C_{||}+s^2L_1C_{||}}$$

where $C_{||} = C_1C_2/(C_1+C_2)$

- **Inductor is necessary to create a complex pole pair**
  - **Must be implemented off-chip due to its large value**

Problem with Passive Loop Filter Implementations

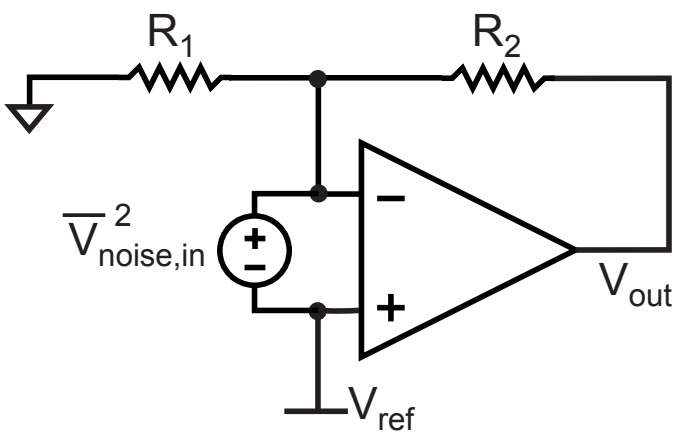
- Large voltage swing required at charge pump output
  - Must support full range of VCO input
- Non-ideal behavior of inductors (for third order G(f) implementations)
  - Hard to realize large inductor values
  - Self resonance of inductor reduces high frequency attenuation



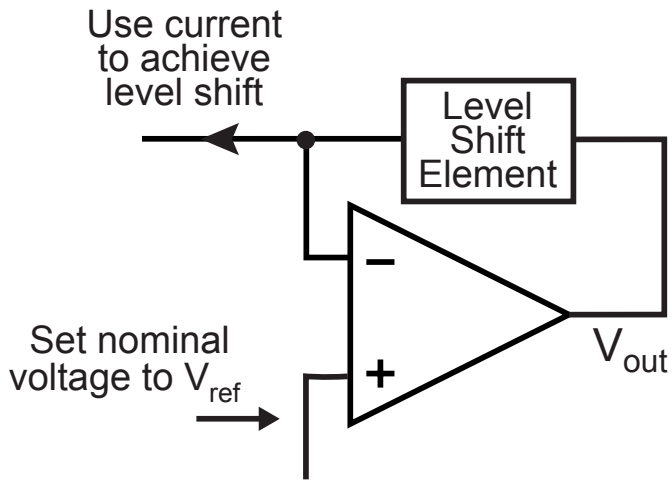
Alternative: active loop filter implementation

Guidelines for Active Loop Filter Design

- Use topologies with unity gain feedback in the opamp
  - Minimizes influence of opamp noise

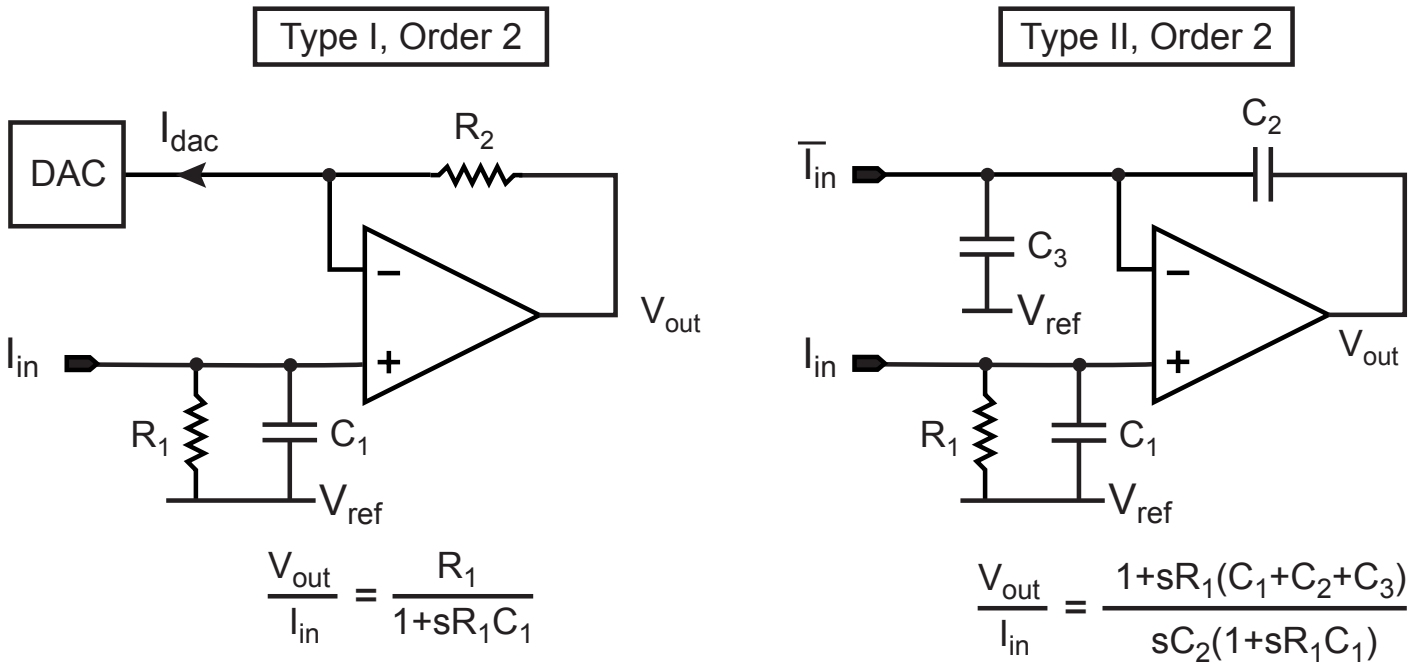


- Perform level shifting in feedback of opamp
  - Fixes voltage at charge pump output



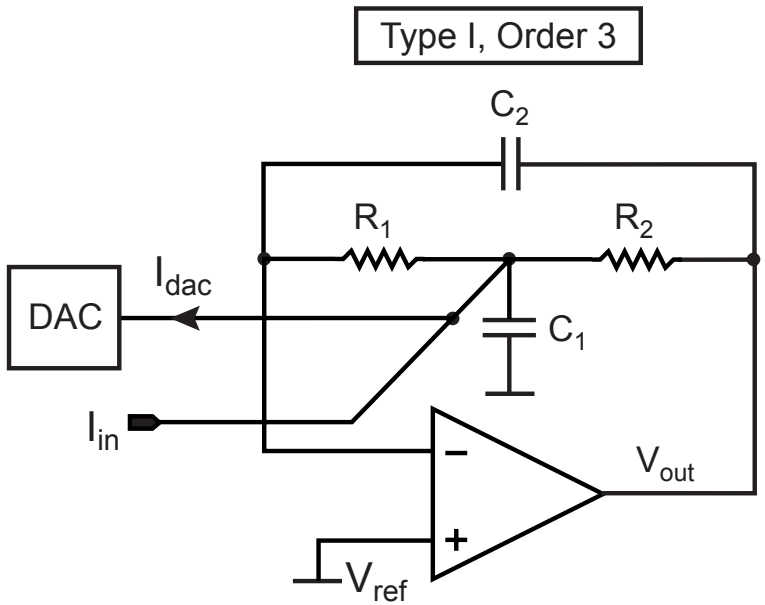
- Prevent fast edges from directly reaching opamp inputs
  - Will otherwise cause opamp to be driven into nonlinear region of operation

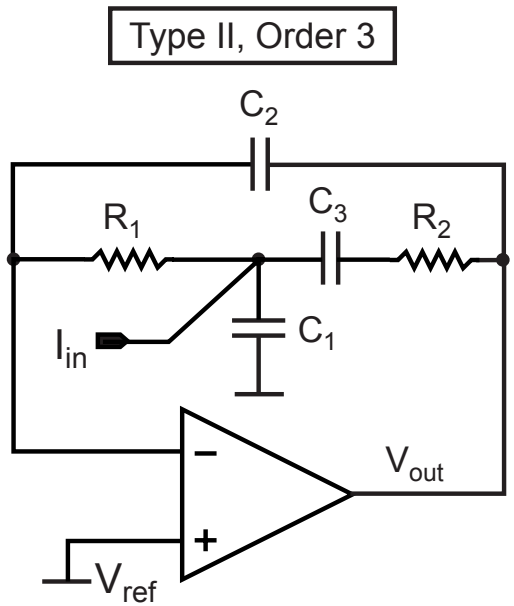
Active Topologies To Realize a Second Order PLL



- Follows guidelines from previous slide
- Charge pump output is terminated directly with a high Q capacitor
  - Smooths fast edges from charge pump before they reach the opamp input(s)

Active Topologies To Realize a Third Order PLL





$$\frac{V_{out}}{I_{in}} = \frac{-R_2}{1+s(R_1+R_2)C_2+s^2R_1R_2C_1C_2}$$

$$\frac{V_{out}}{I_{in}} = \frac{-1}{s(C_1+C_2)} \cdot \frac{1+sR_2C_3}{1+sC_{||}(R_1(1+C_1/C_3)+R_2)+s^2R_1R_2C_1C_{||}}$$

where $C_{||} = C_2C_3/(C_2+C_3)$

- **Follows active implementation guidelines from a few slides ago**

Example Design

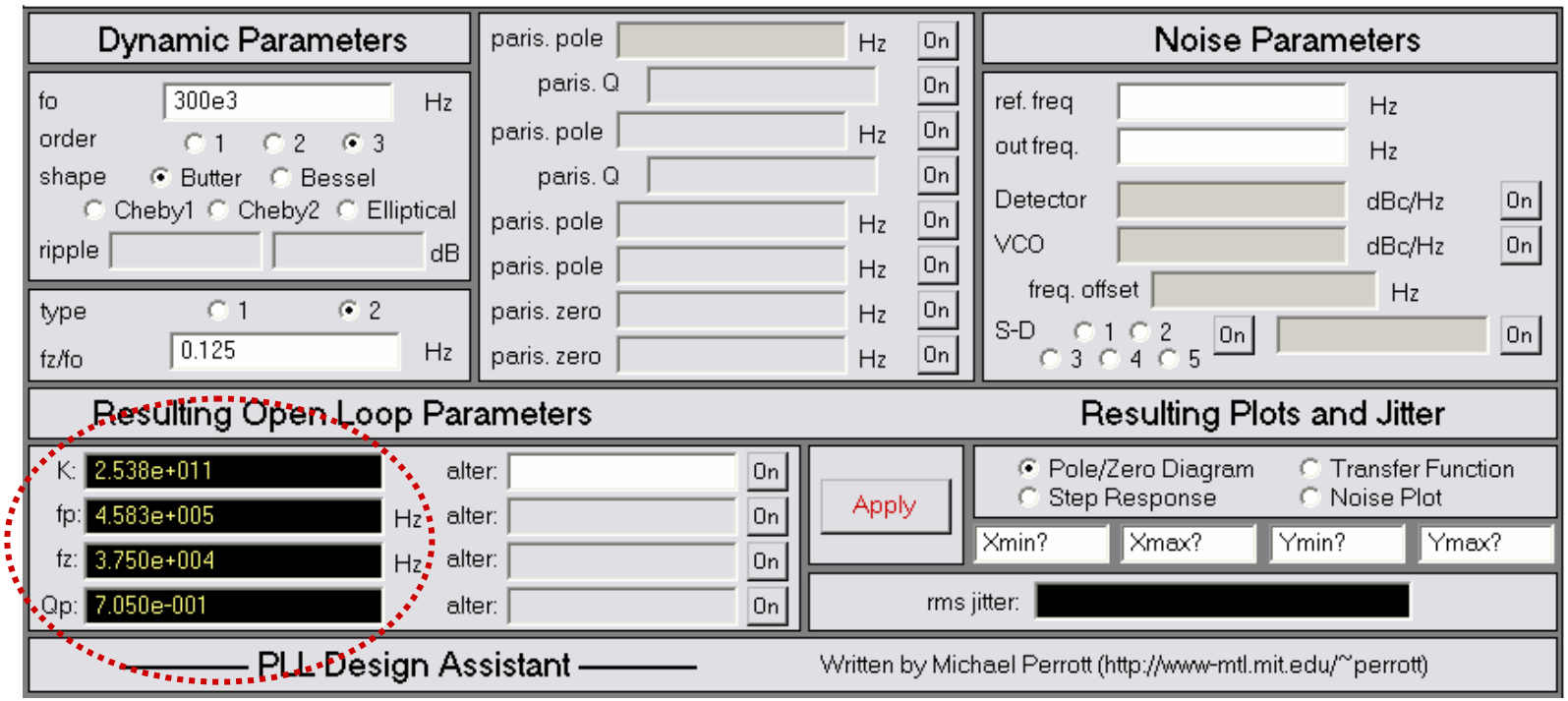
- Type II, 3rd order, Butterworth, f_o = 300kHz, f_z/f_o = 0.125
  - No parasitic poles
- Required loop filter transfer function can be found from table:

$$\Rightarrow H(s) = \frac{K_{LF}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_p Q_p}+\left(\frac{s}{w_p}\right)^2\right)} \quad \text{where}$$

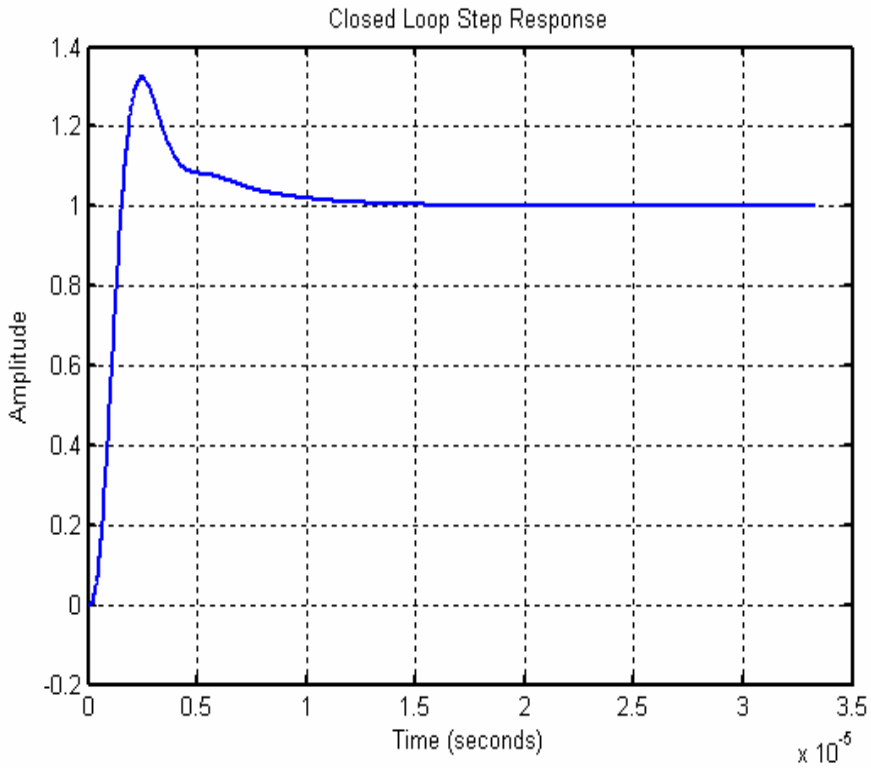
$$K_{LF} = \frac{N_{nom} K}{\alpha I_{cp} K_v}$$

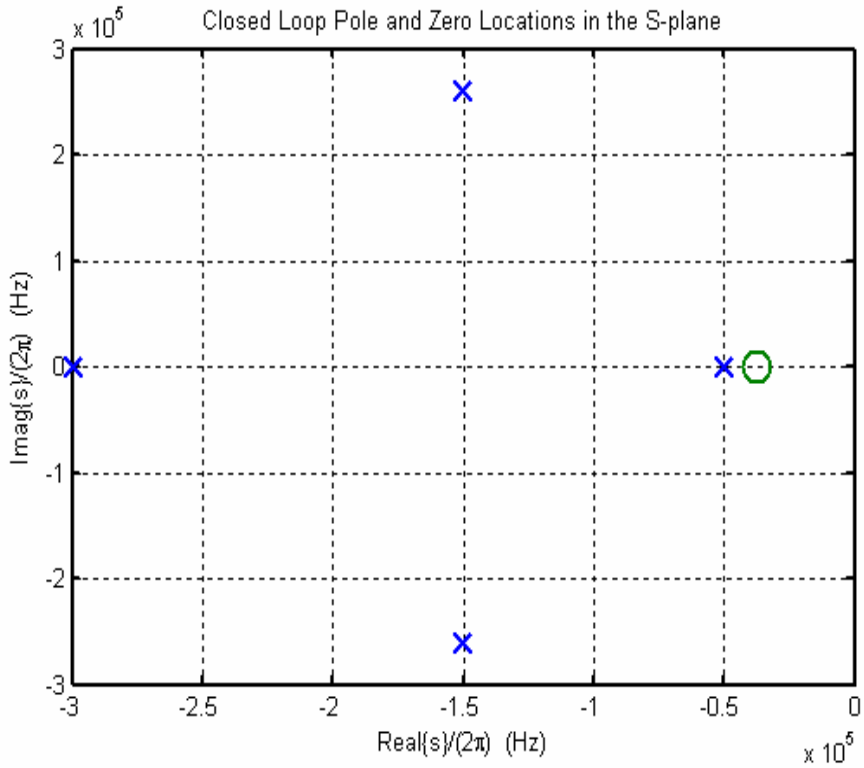
Use PLL Design Assistant to Calculate Parameters

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

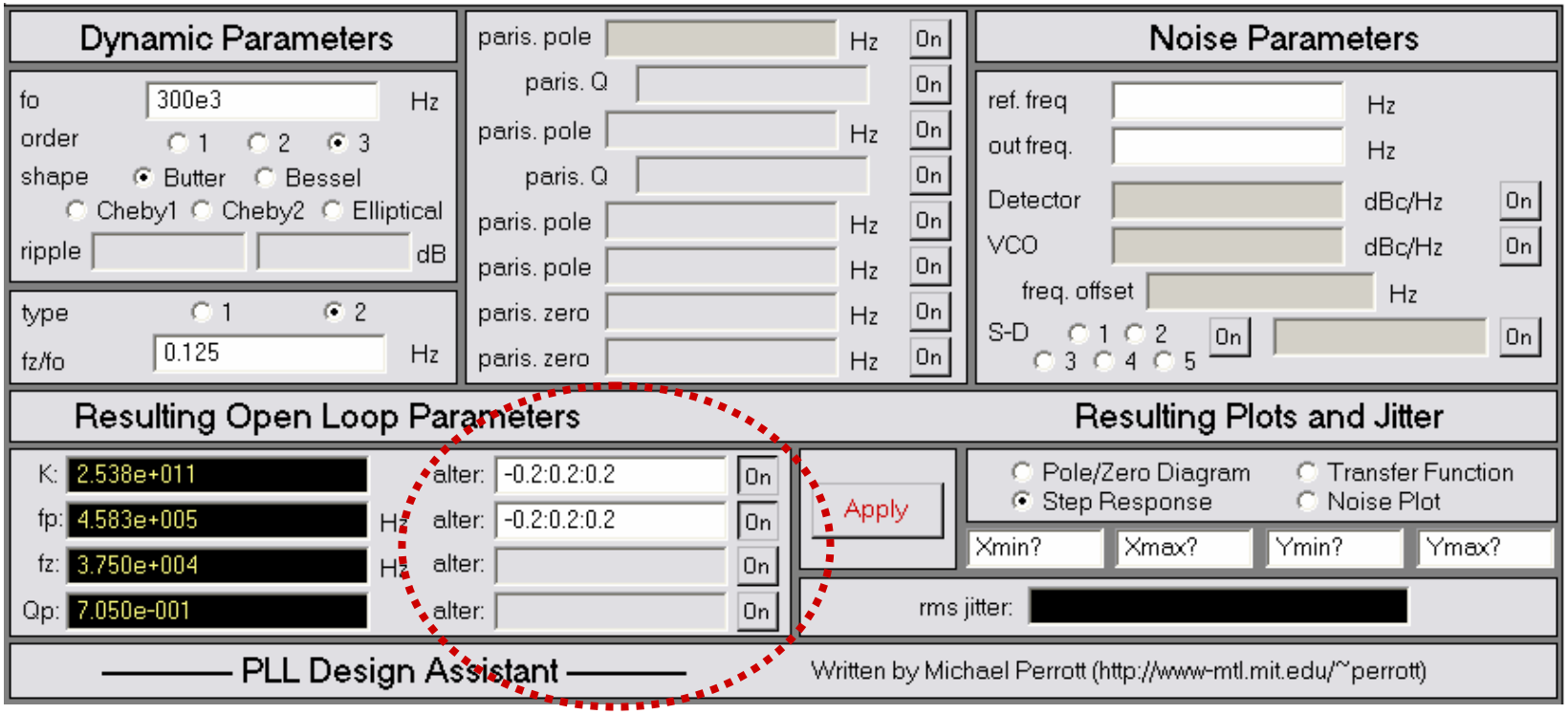


Resulting Step Response and Pole/Zero Diagram



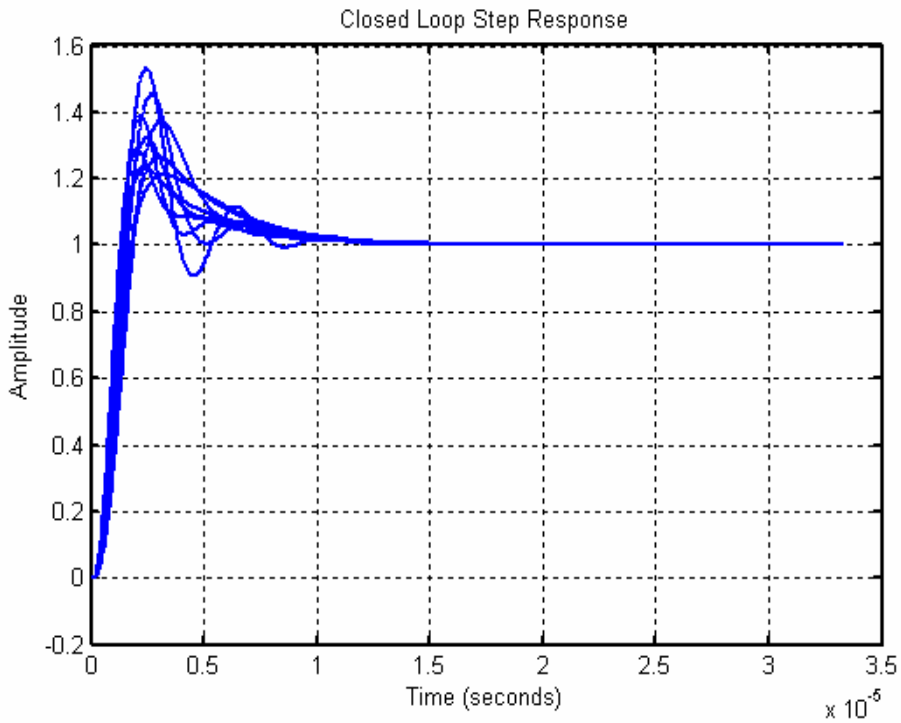


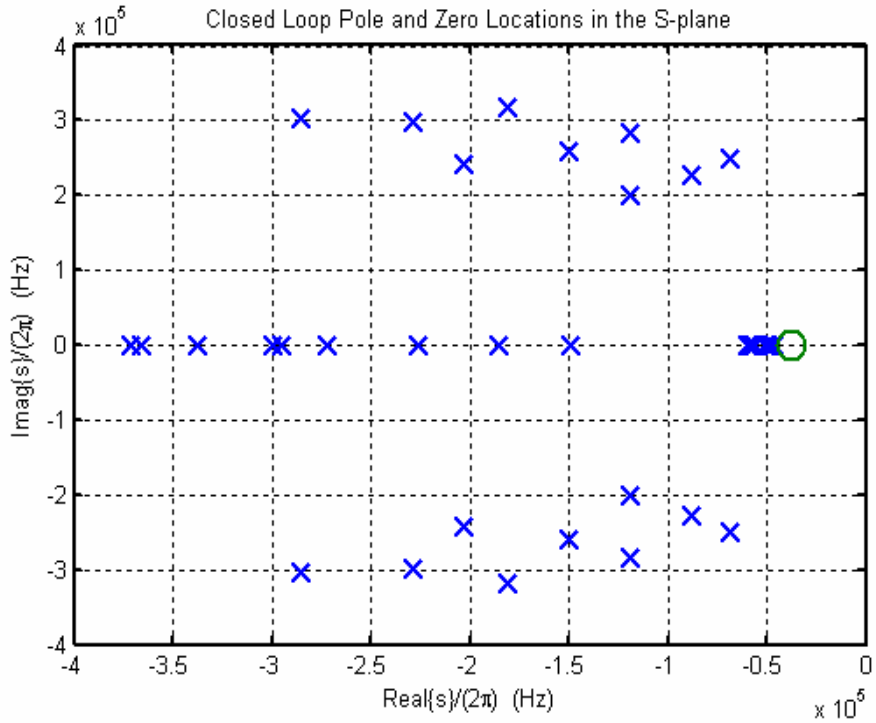
Impact of Open Loop Parameter Variations



- **Open loop parameter variations can be directly entered into tool**

Resulting Step Responses and Pole/Zero Diagrams



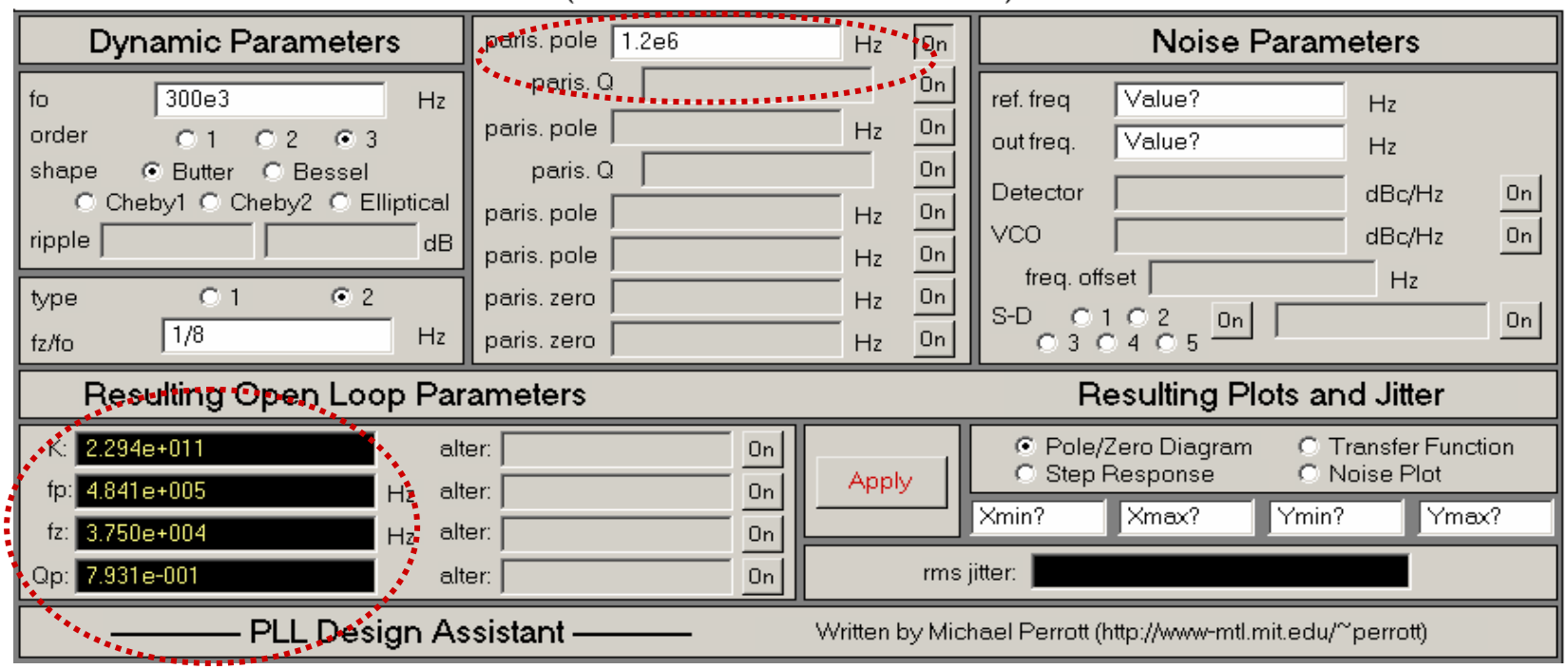


- **Impact of variations on the loop dynamics can be visualized instantly and taken into account at early stage of design**

Design with Parasitic Pole

- **Include a parasitic pole at nominal value f_{p1} = 1.2MHz**

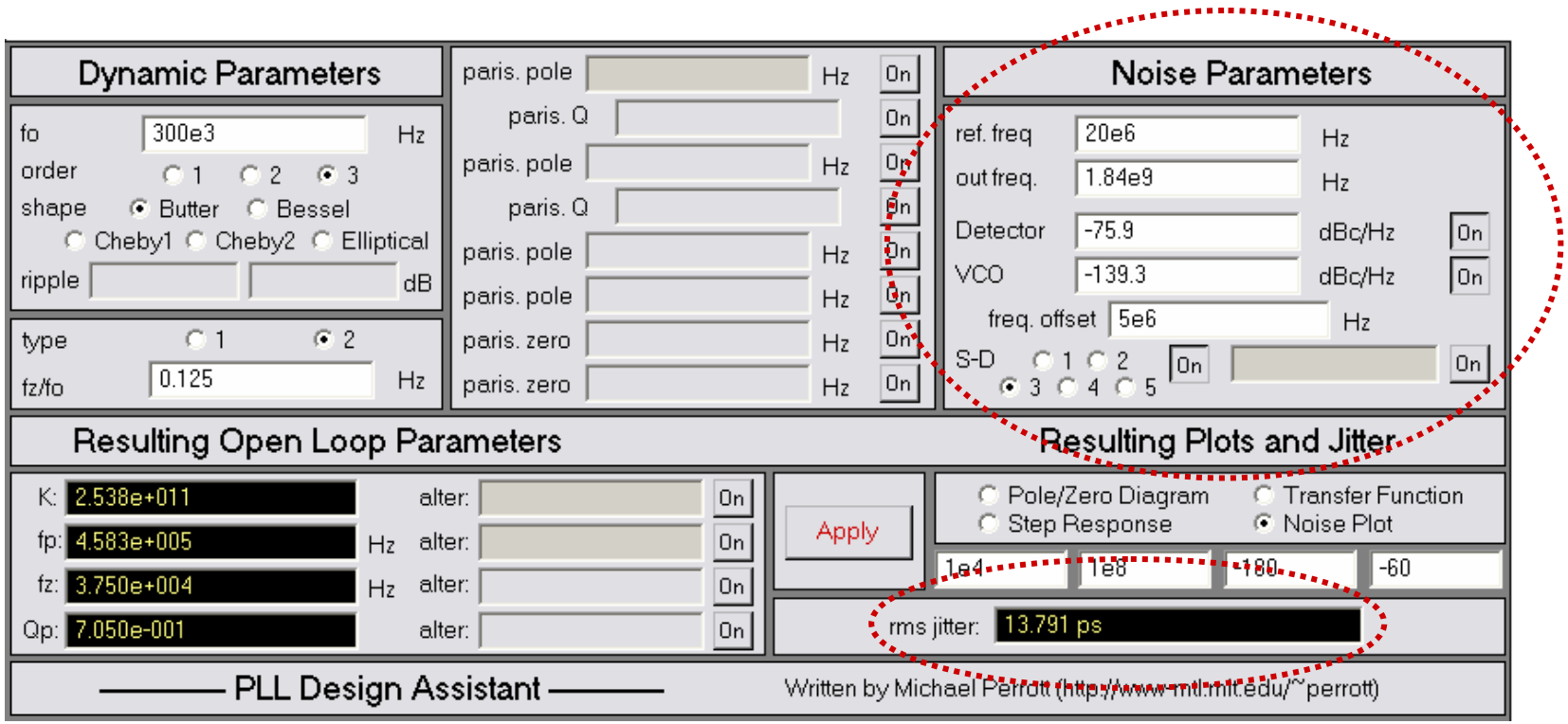
$$H(s) = \frac{K_{LF}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_p Q_p}+\left(\frac{s}{w_p}\right)^2\right)\left(1+\frac{s}{w_{p1}}\right)}$$



⇨ K, f_p and Q_p are adjusted to obtain the same dominant pole locations

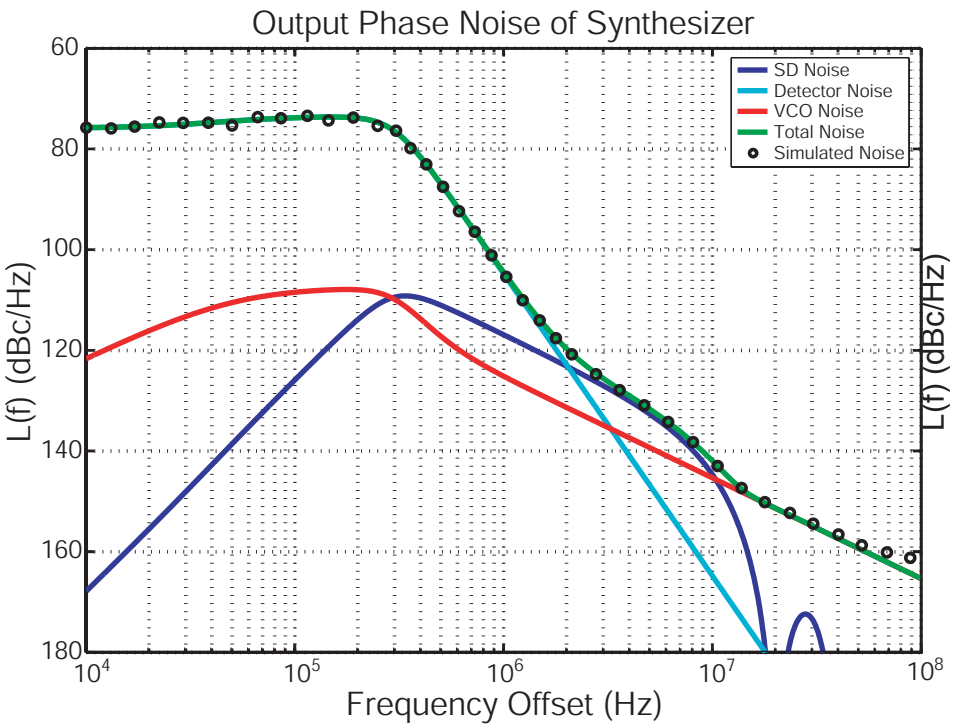
Noise Estimation

- Phase noise plots can be easily obtained
  - Jitter calculated by integrating over frequency range

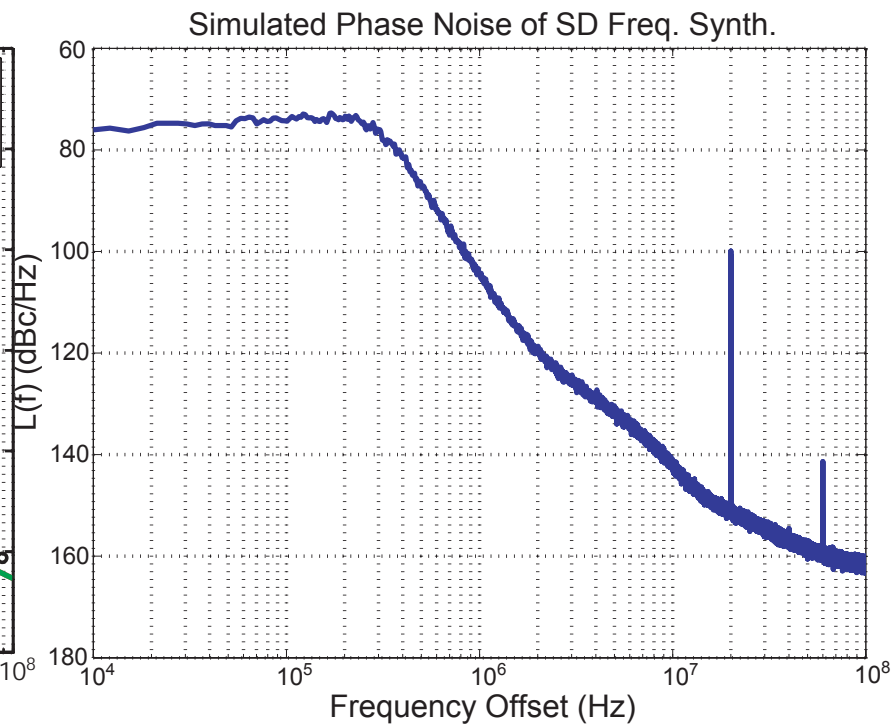


Calculated Versus Simulated Phase Noise Spectrum

Without parasitic pole:

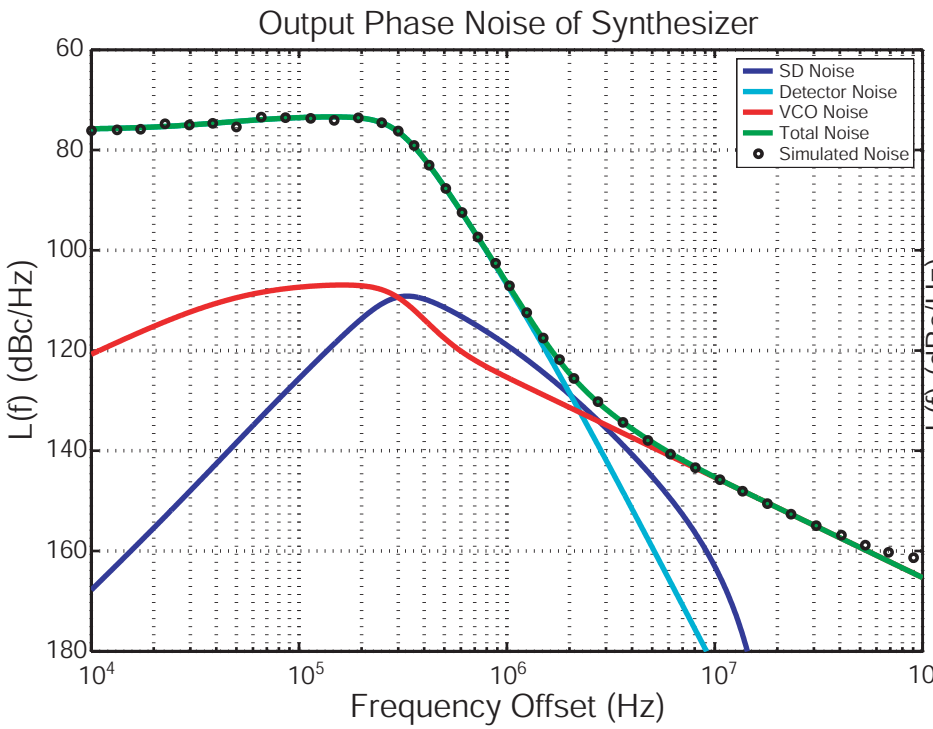


RMS jitter = 13.791ps

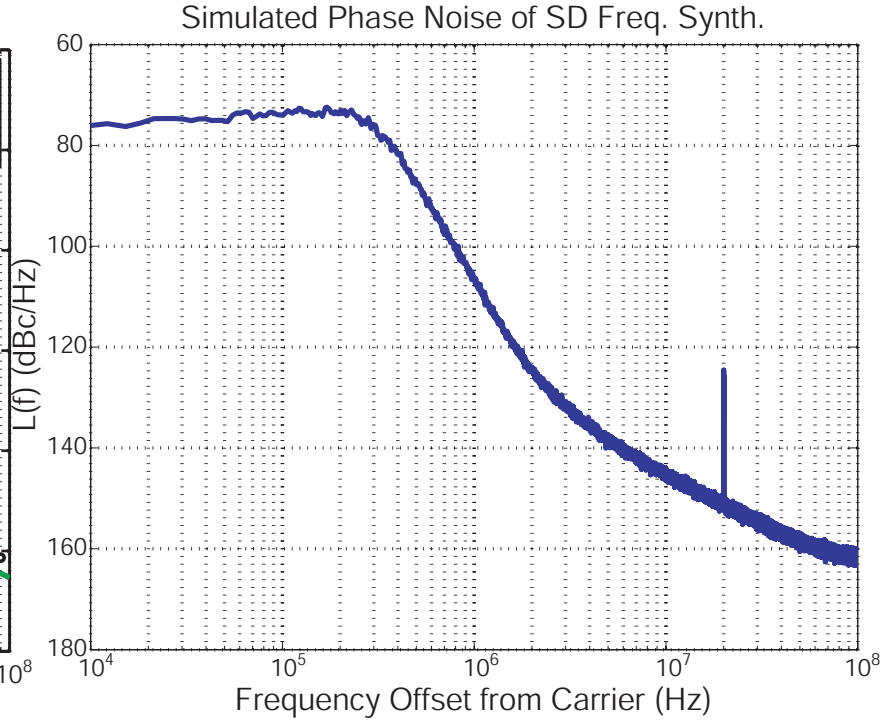


Calculated Versus Simulated Phase Noise Spectrum

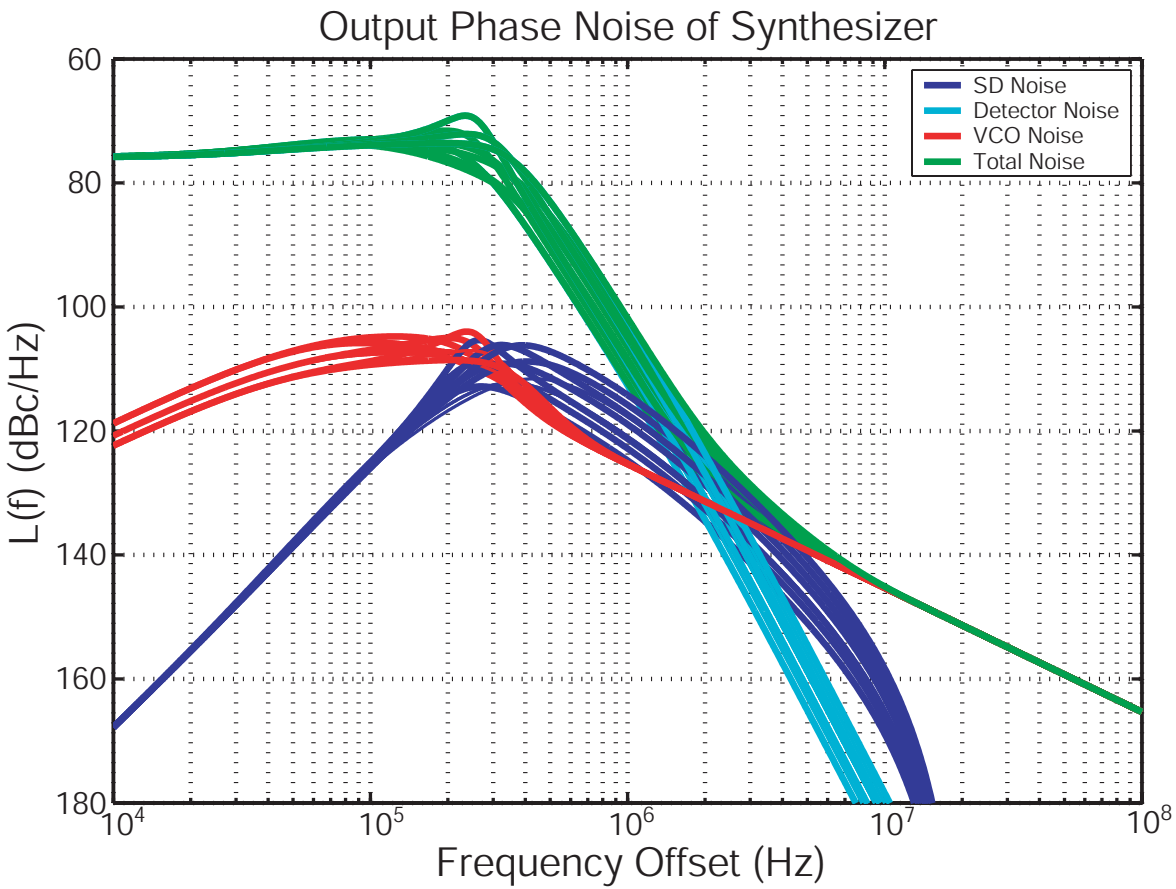
With parasitic pole at 1.2 MHz:



RMS jitter = 14.057ps



Noise under Open Loop Parameter Variations



RMS jitter = 11.678ps (min), 18.211ps (max)

- **Impact of open loop parameter variations on phase noise and jitter can be visualized immediately**

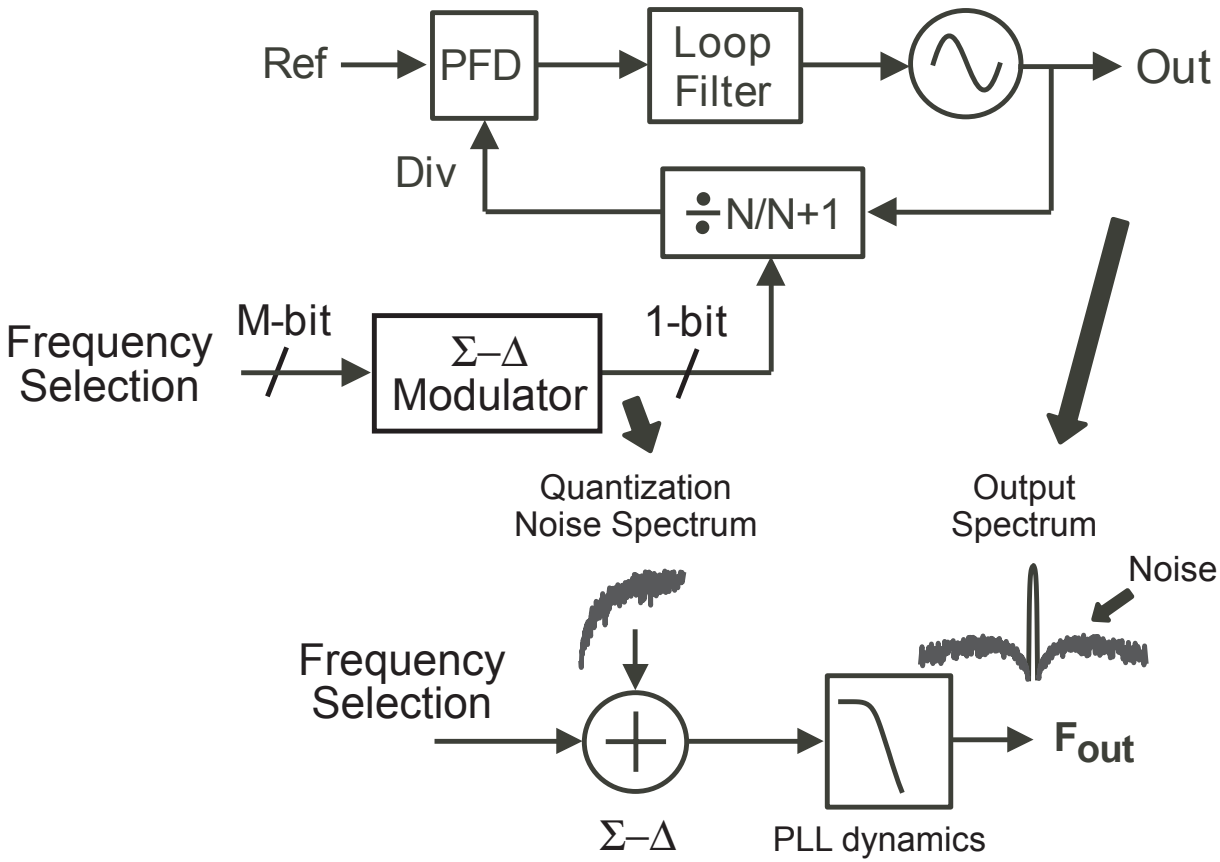
Conclusion

- **New closed loop design approach facilitates:**
  - **Accurate control of closed loop dynamics**
    - Bandwidth, Order, Shape, Type
  - **Straightforward design of higher order PLL's**
  - **Direct assessment of impact of parasitic poles/zeros**
- **Techniques implemented in a GUI-based CAD tool**

- **Beginners can quickly come up to speed in designing PLL's**
- **Experienced designers can quickly evaluate the performance of different PLL configurations**

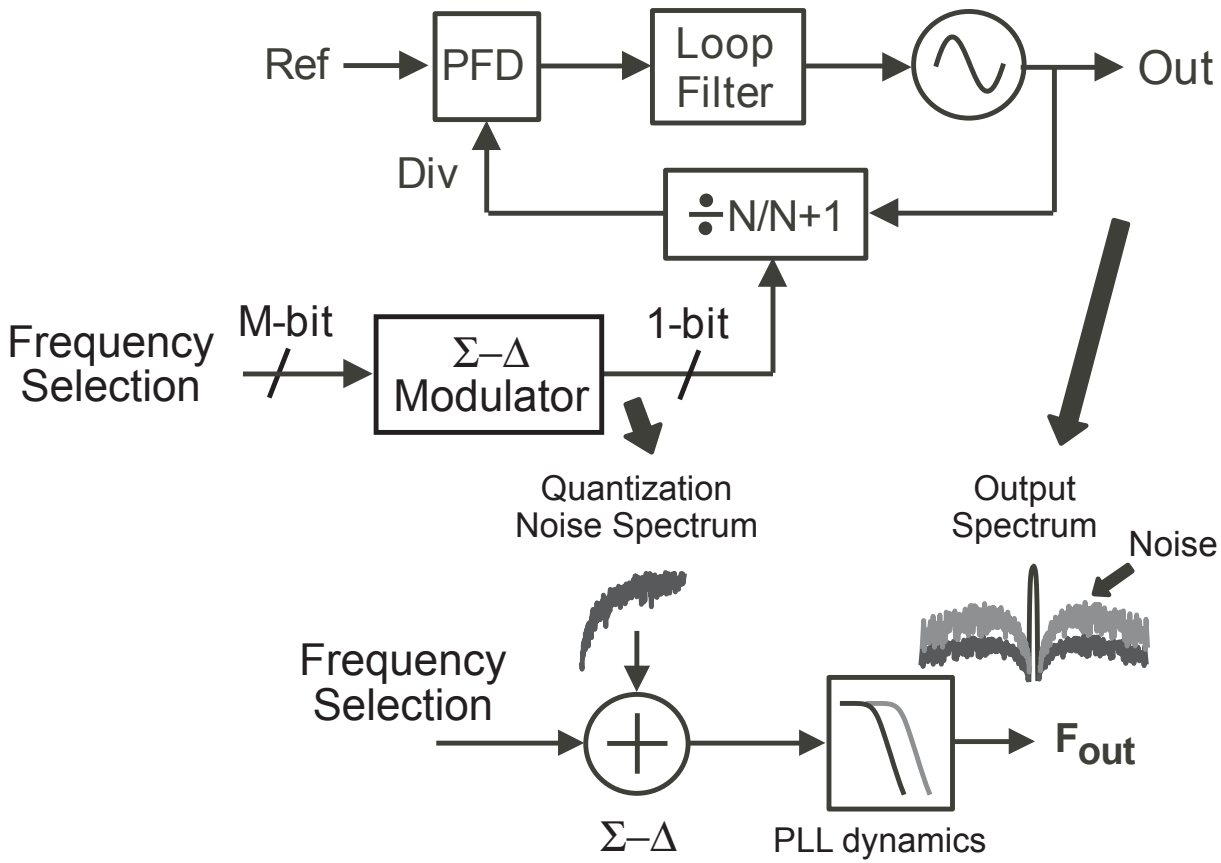
Bandwidth Extension of Fractional-N Frequency Synthesizers

Impact of Σ–Δ Quantization Noise on Synth. Output



- **Lowpass action of PLL dynamics suppresses the shaped Σ-Δ quantization noise**

Impact of Increasing the PLL Bandwidth



- **Higher PLL bandwidth leads to less quantization noise suppression**

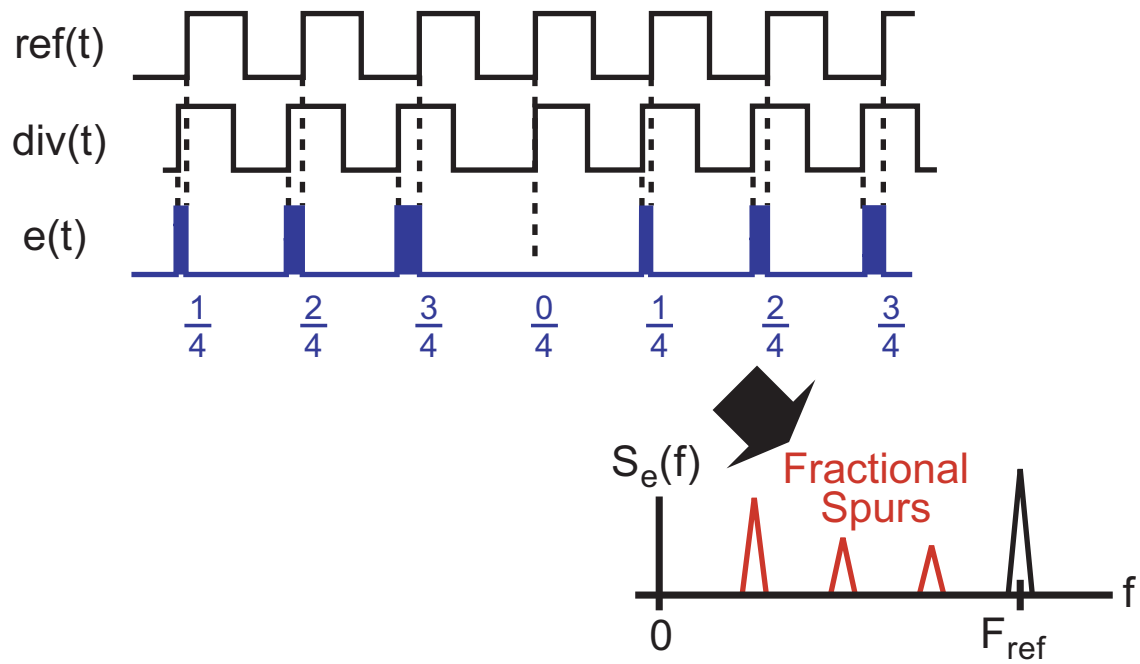
Tradeoff: Noise performance vs PLL bandwidth

Recent Approaches to Bandwidth Extension

- [1] C. Park, O. Kim, and B. Kim, "A 1.8GHz Self-Calibrated Phase-Locked Loop with Precise I/Q Matching," IEEE JSSC, May 2001.
- [2] K. Lee, et. al., "A Single Chip 2.4GHz Direct-Conversion CMOS Receiver for Wireless Local Loop Using Multiphase Reduced Frequency Conversion Technique ," IEEE JSSC, May 2001.
- [3] S. Pamarti, L. Jansson, and I. Galton, "A Wideband 2.4GHz Delta-Sigma Fractional-N PLL With 1Mb/s In-Loop Modulation", IEEE JSSC, Jan 2004
- [4] E. Temporiti, et. al., "A 700kHz Bandwidth Σ–Δ Fractional-N Frequency Synthesizer with Spurs Compensation and Linearization Techniques for WCDMA Applications", IEEE JSSC, Sept 2004

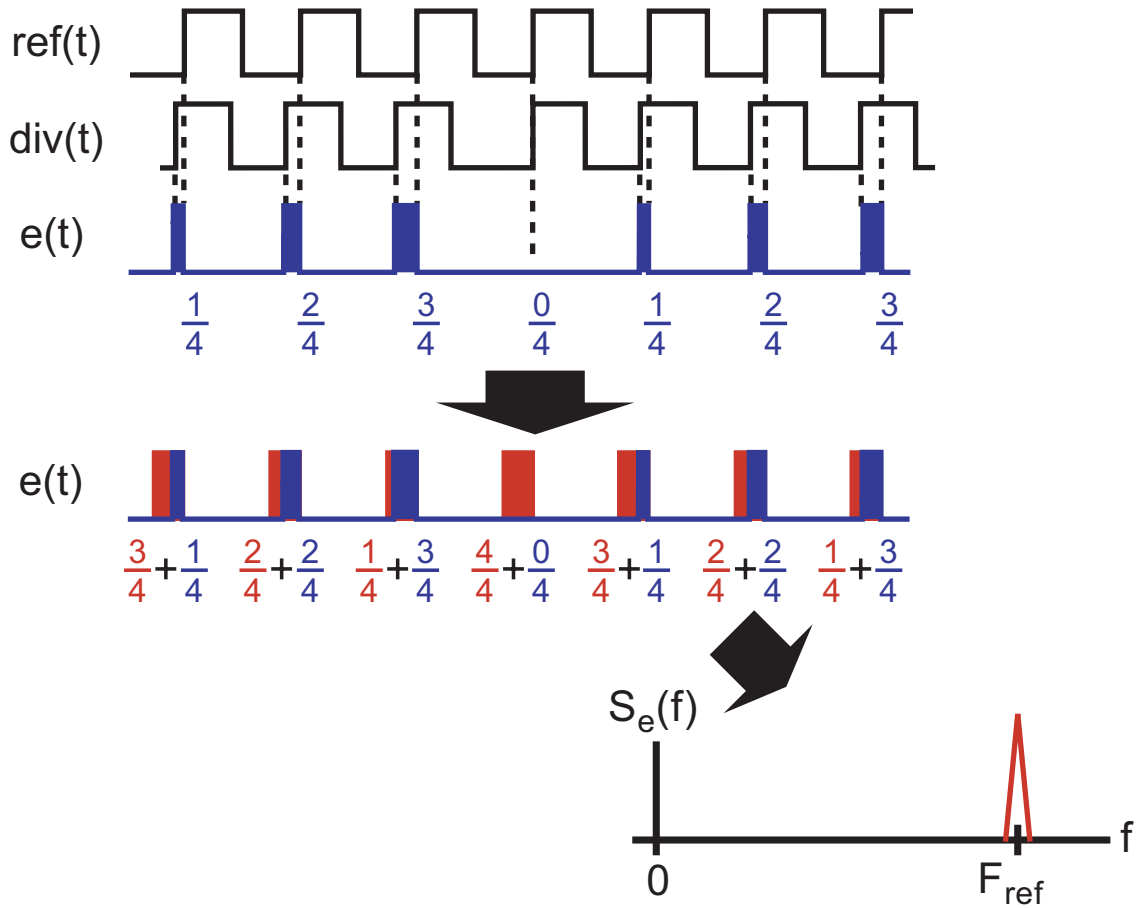
We will focus on our own approach in this talk

Examine Classical Fractional-N Signals



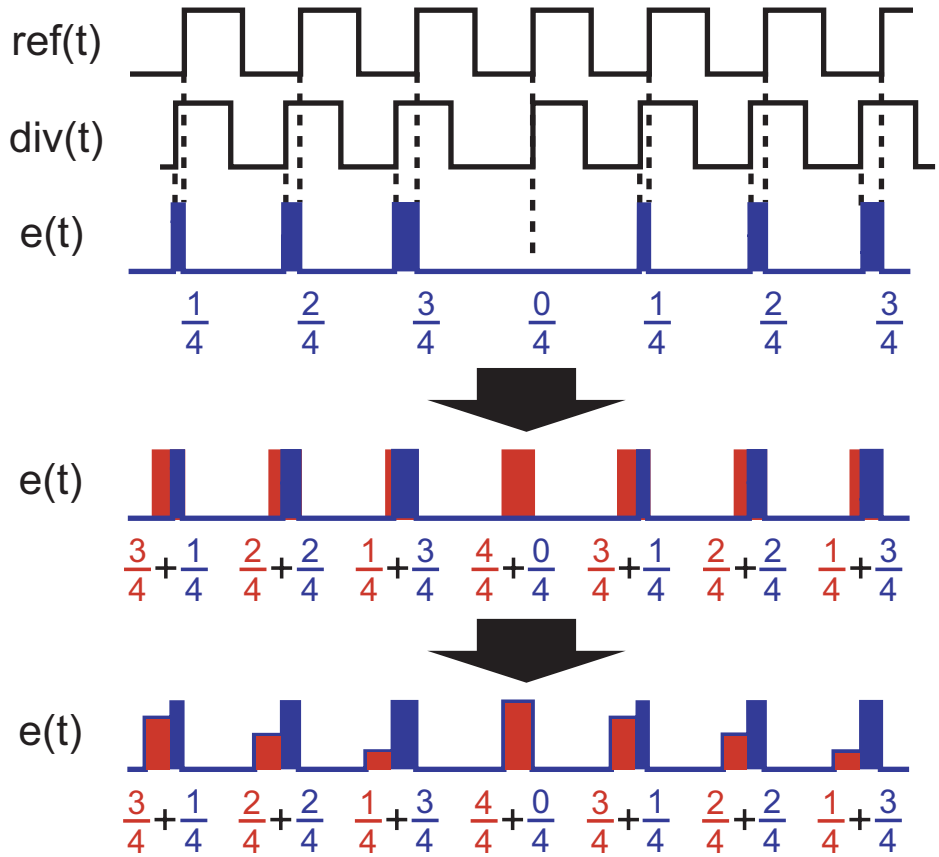
- **Goal: eliminate the fractional spurs**

Method 1: Vertical Compensation

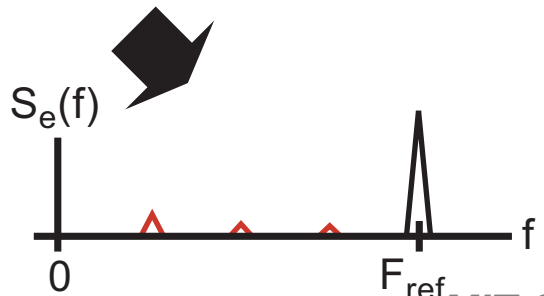


- "Fill in" pulses so that they are constant area
  - Fractional spurs are eliminated!

Method 2: Horizontal Compensation

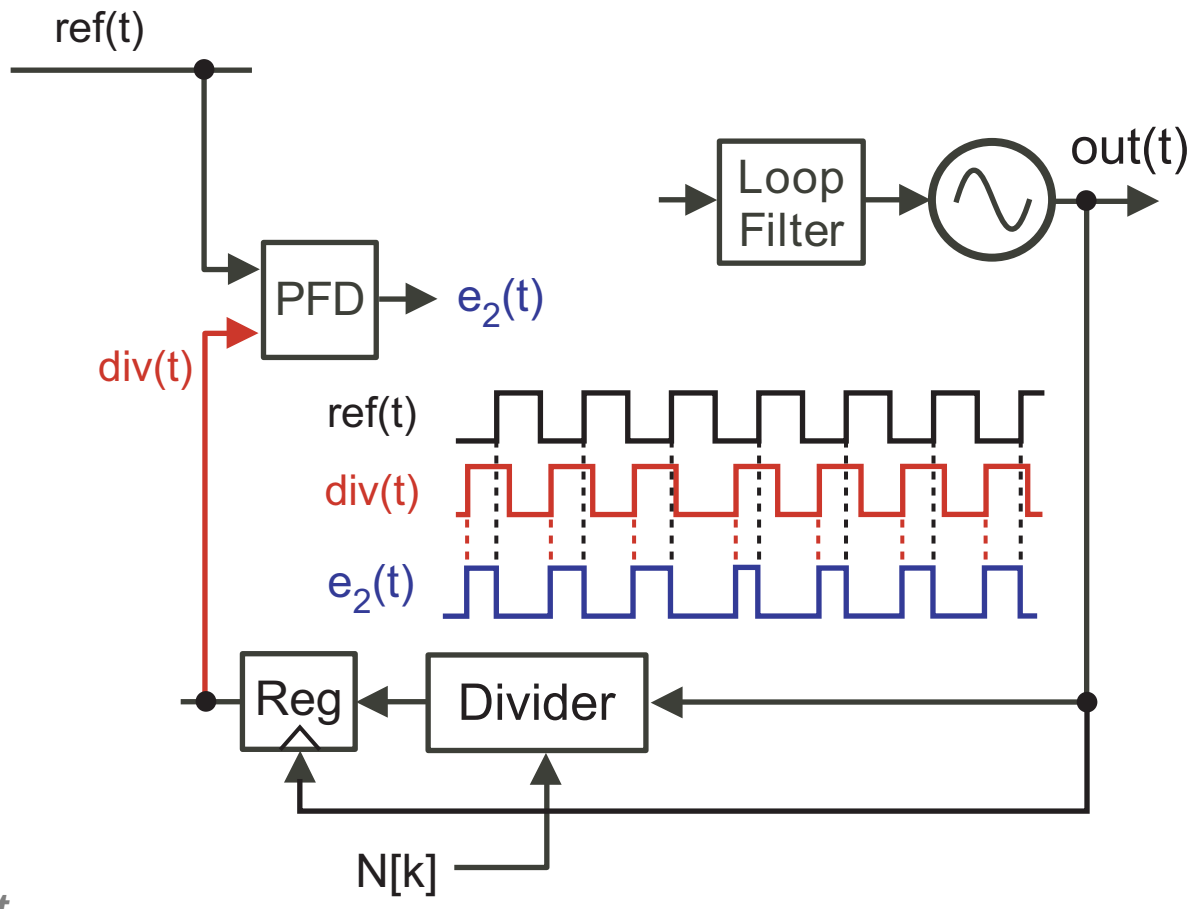


- **Use constant width pulses of varying height to achieve constant area pulses**
  - **Largely eliminates fractional spurs**

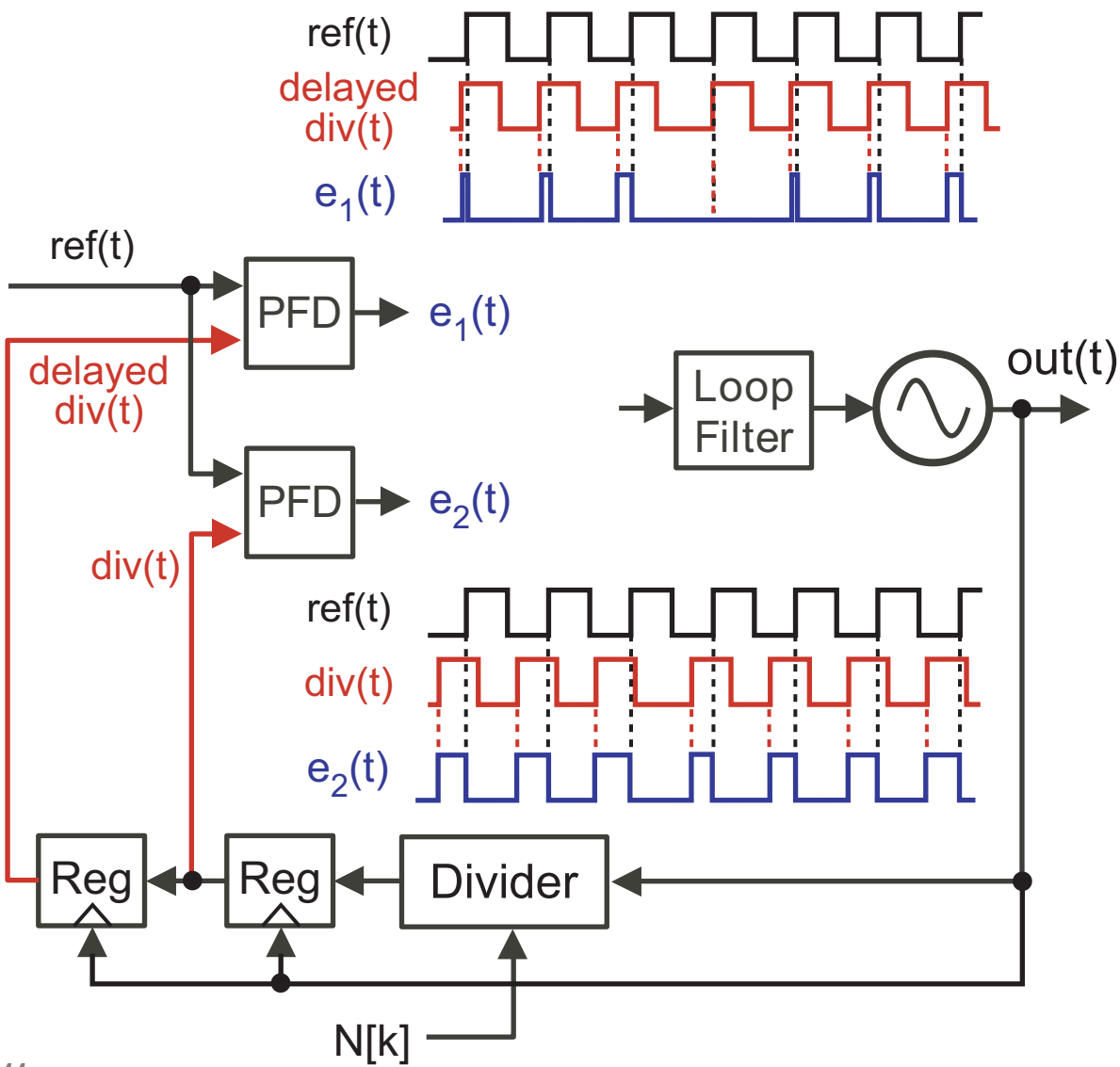


Implementation of Horizontal Cancellation

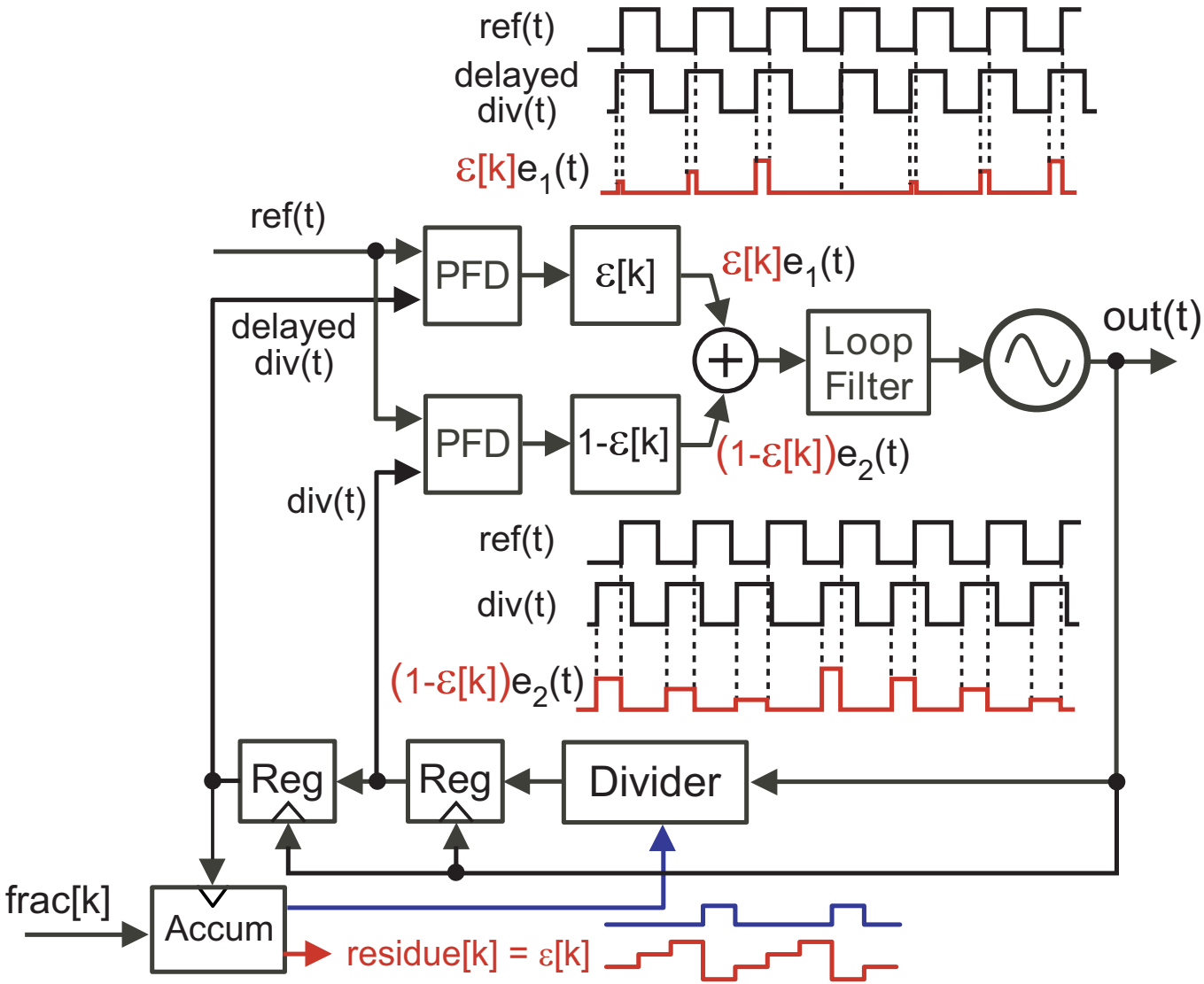
- **We begin with the basic fractional-N structure**



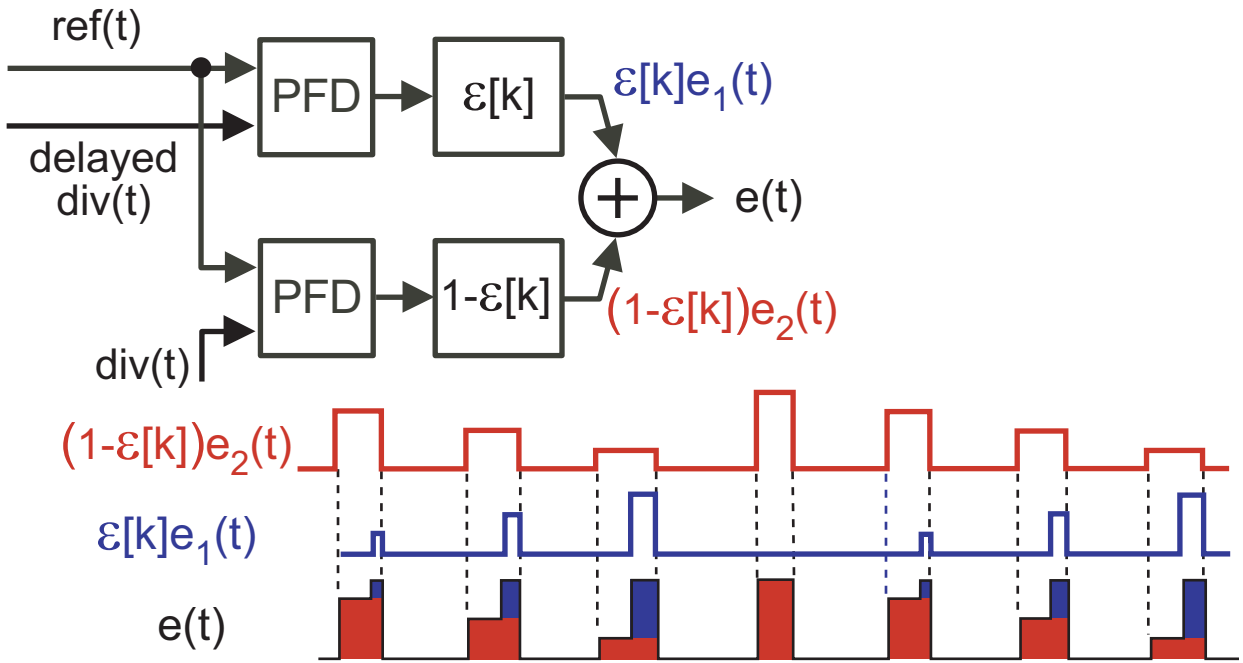
Add a Second PFD with Delayed Divider Signal



Scale Error Pulses According to Accumulator Residue



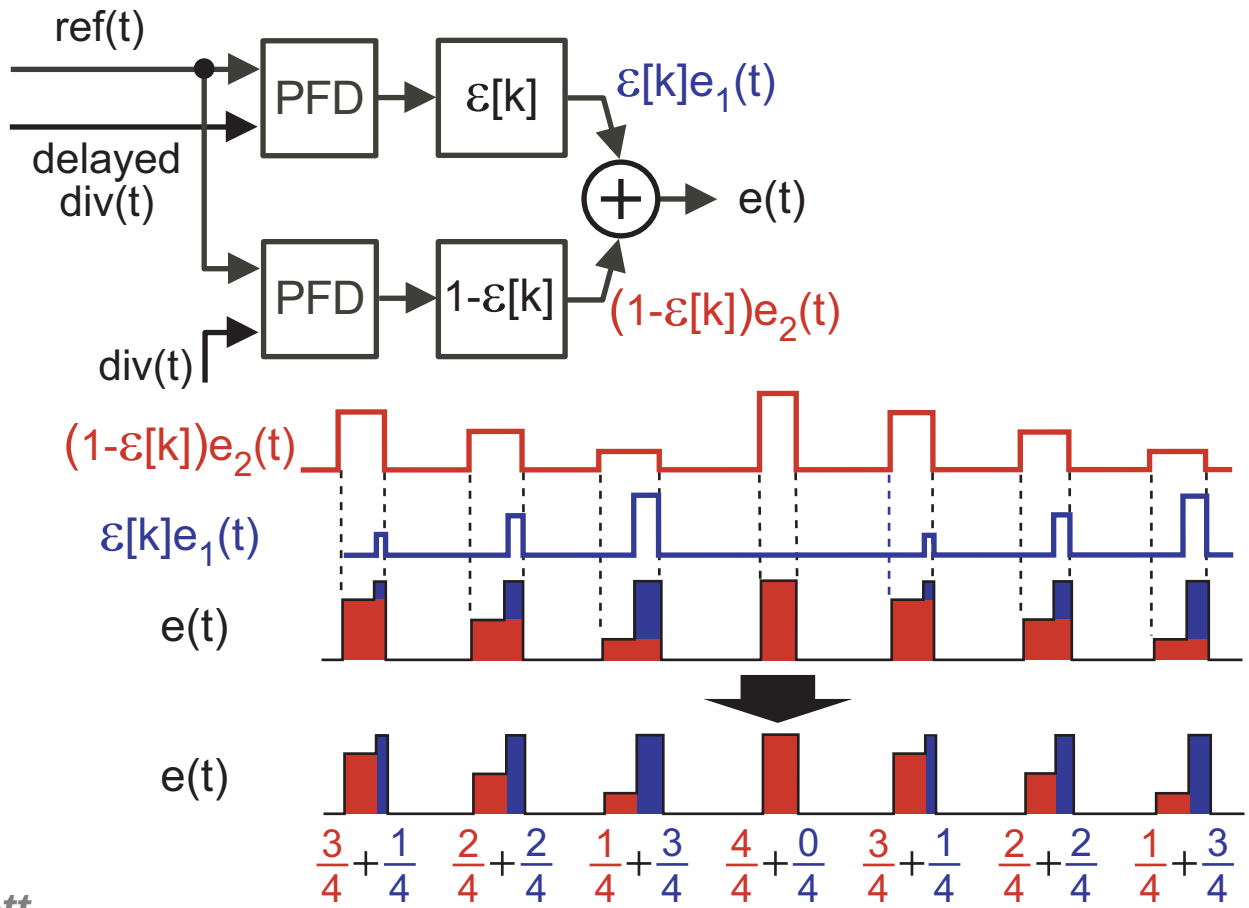
A Closer Look at Adding the Scaled Error Pulses



- **Goal – keep area constant for each pulse**
  - **It's easier to see this from a slightly different viewpoint**

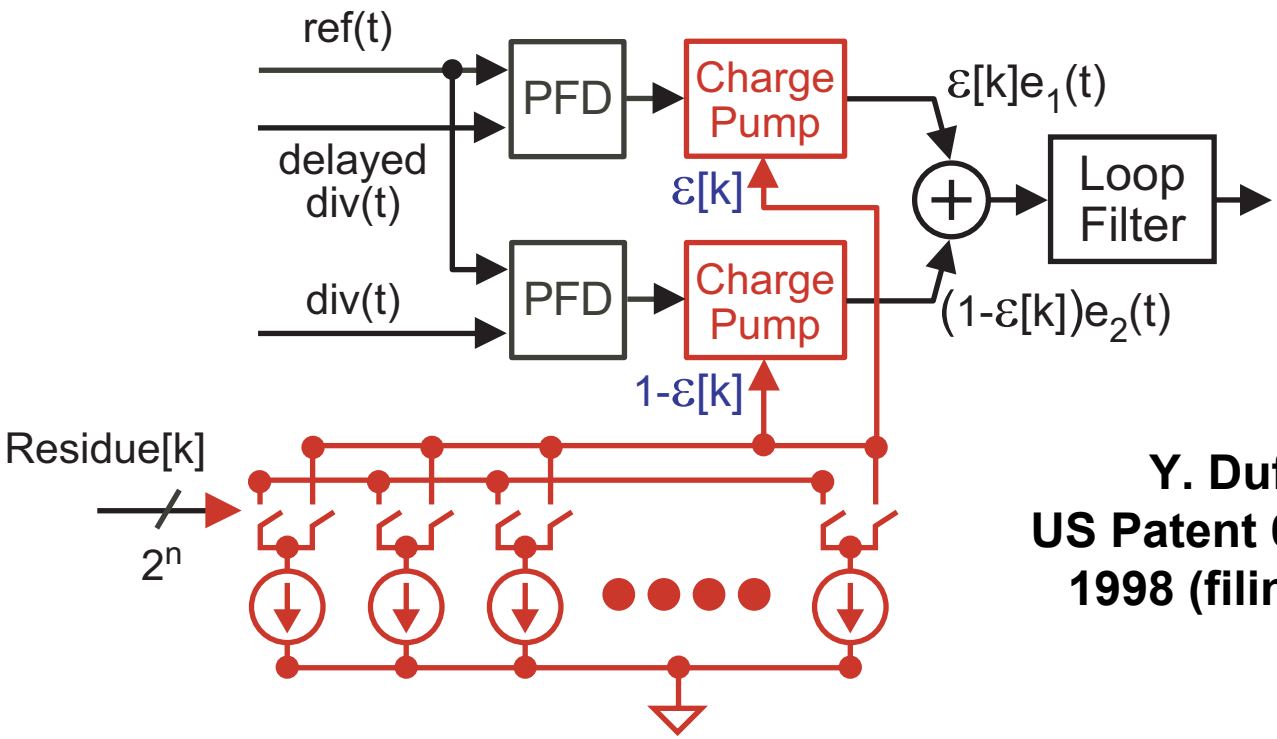
Alternate Viewpoint

- **The sum of scaled pulses can now be viewed as horizontal cancellation**



Implementation of Pulse Scaling Operation

- **Direct output of a differential current DAC into two charge pumps**



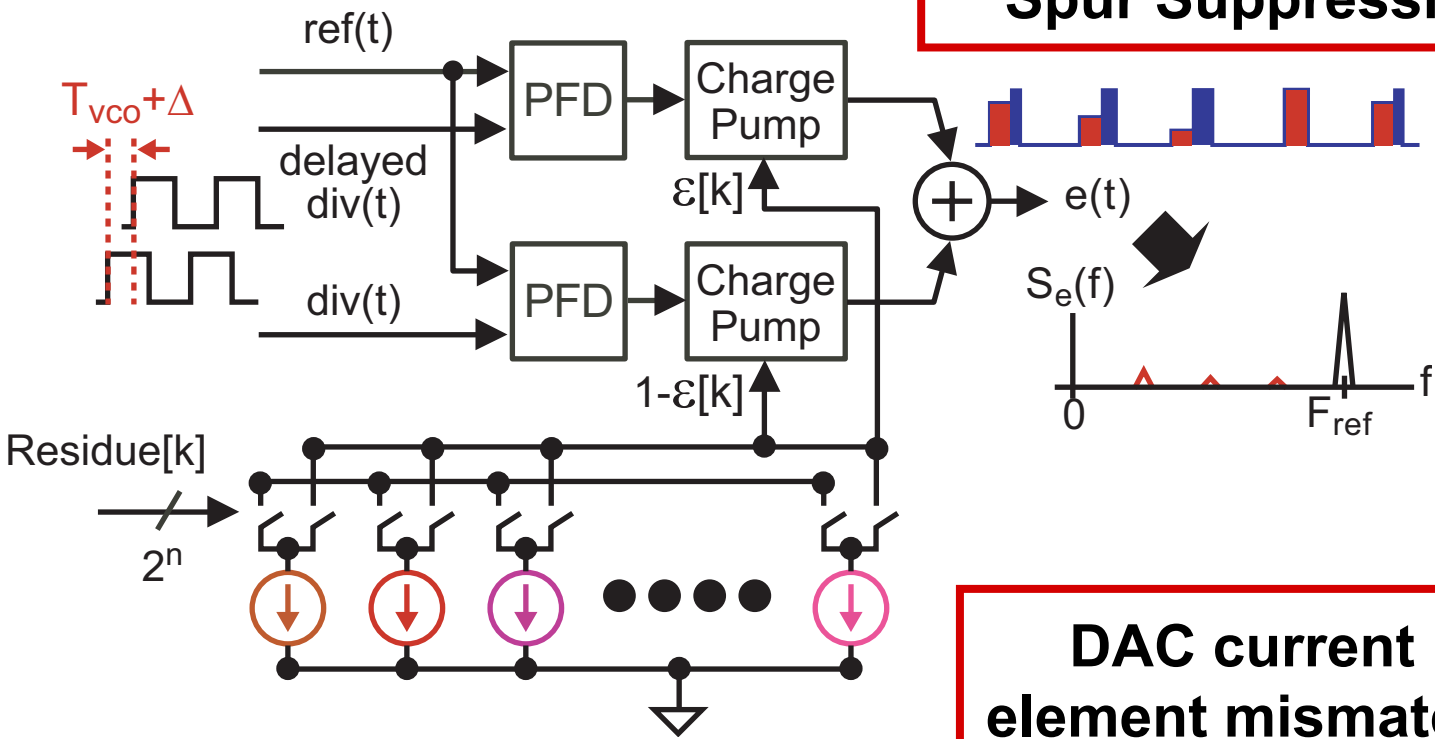
Y. Dufour
US Patent 6,130,561
1998 (filing date)

- **Issue: practical non-idealities kill performance**

Primary Non-idealities of Concern

Delay mismatch

Incomplete Fractional Spur Suppression

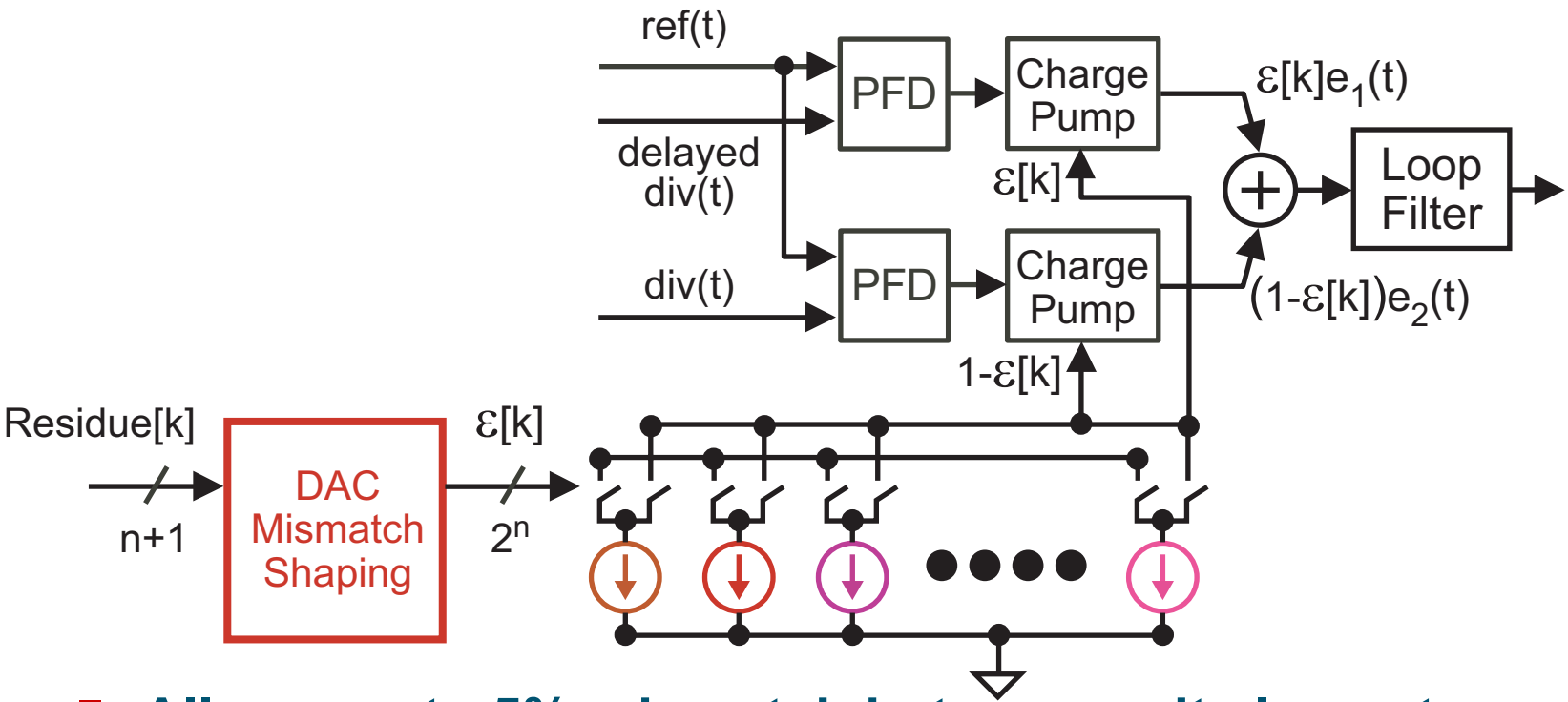


DAC current element mismatch

Proposed approach: dramatically reduce impact of these non-idealities using mixed-signal processing techniques

Eliminate Impact of DAC Current Element Mismatch

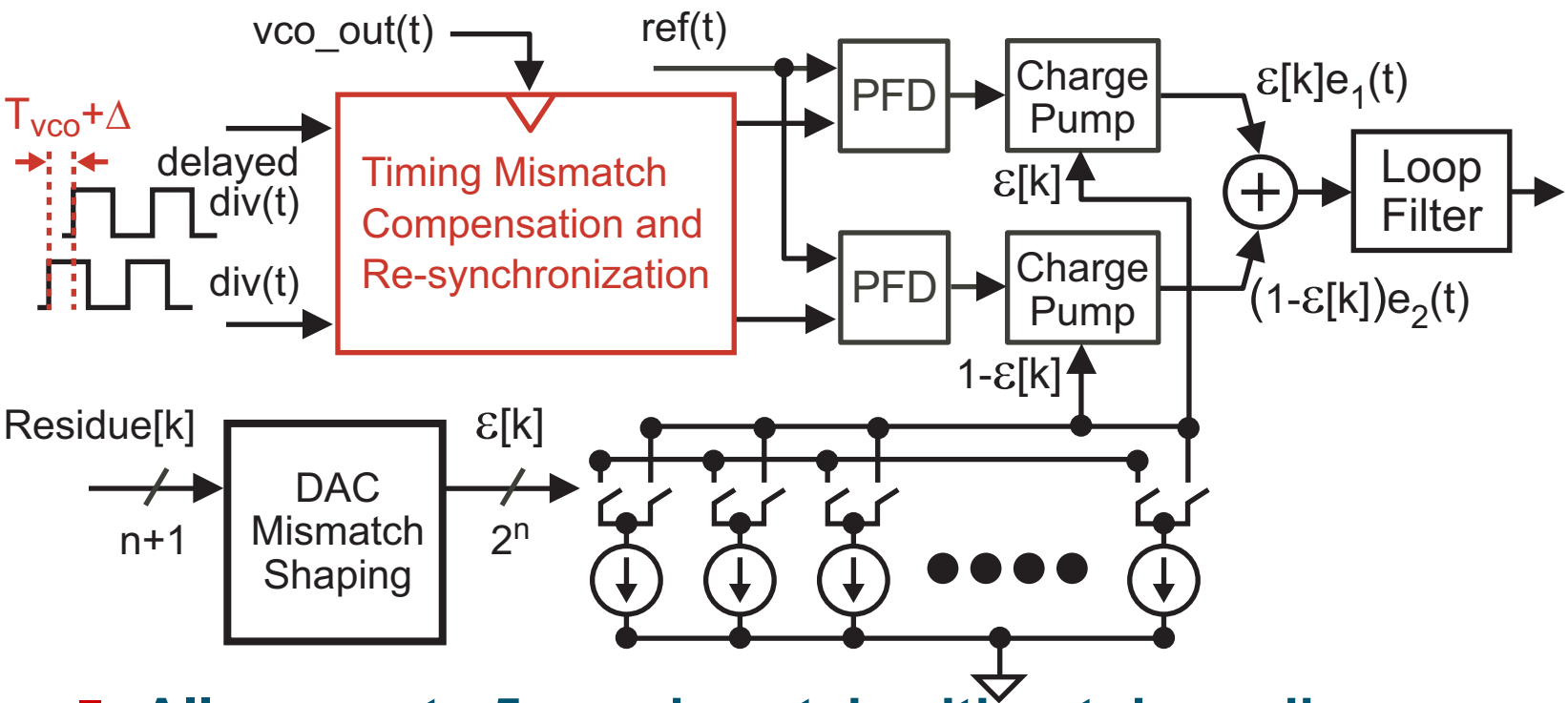
- Apply standard DAC noise shaping techniques to shape mismatch noise to high frequencies
  - See Baird and Fiez, TCAS II, Dec 1995



- Allows up to 5% mismatch between unit elements without degrading our desired performance targets

Eliminate Impact of Timing Mismatch

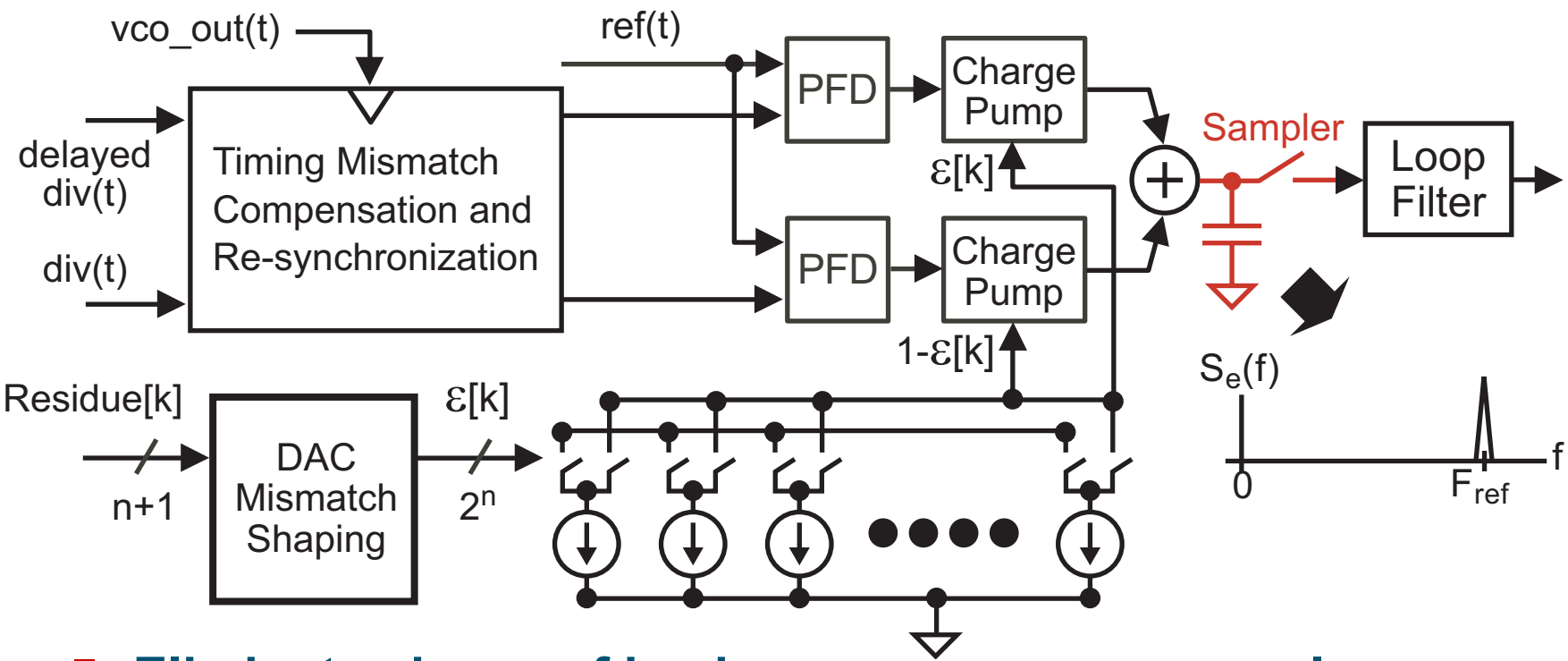
- Swap paths between divider outputs in a pseudo-random fashion
  - Need to also swap ε[k] and 1-ε[k] sequence



- Allows up to 5 ps mismatch without degrading our desired performance targets

Improve Horizontal Cancellation Performance

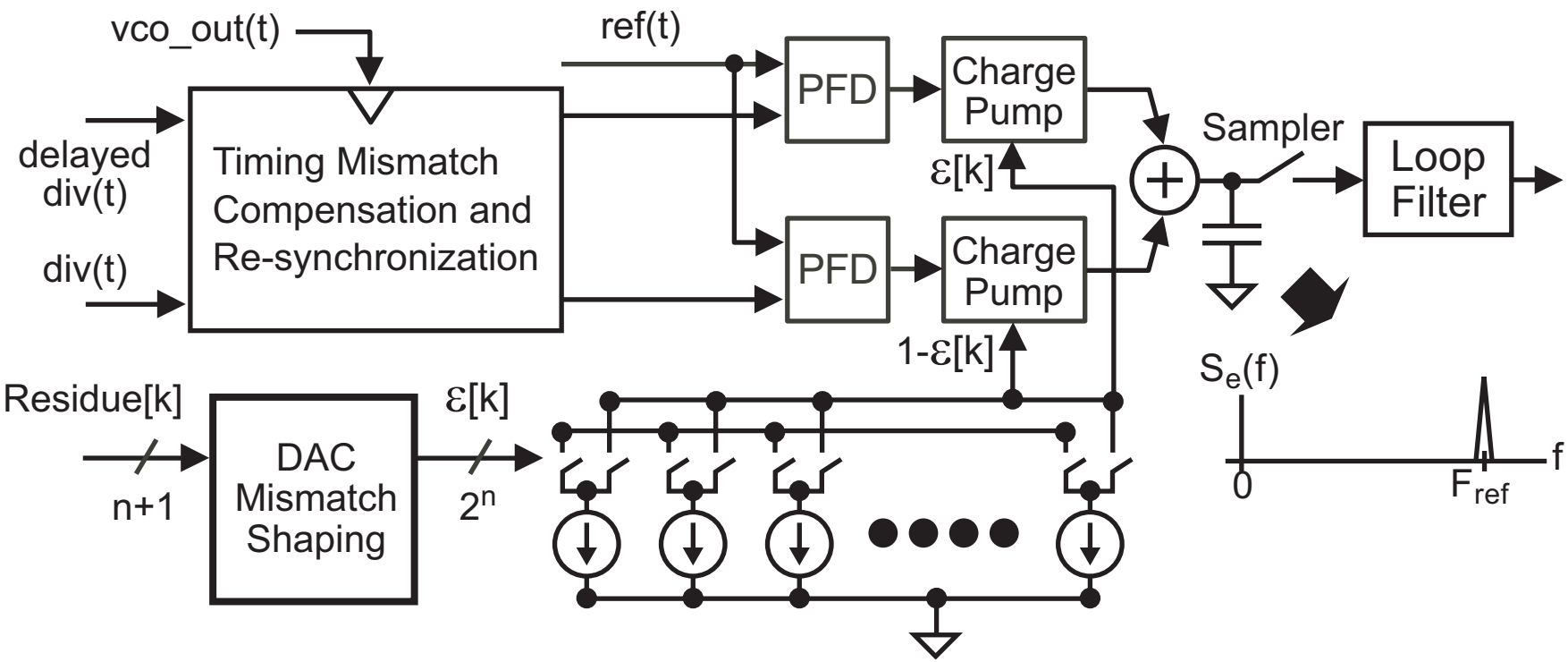
- **Sampling circuit accumulates error pulses before passing their information to the loop filter**
  - **A common analog trick used for decades**



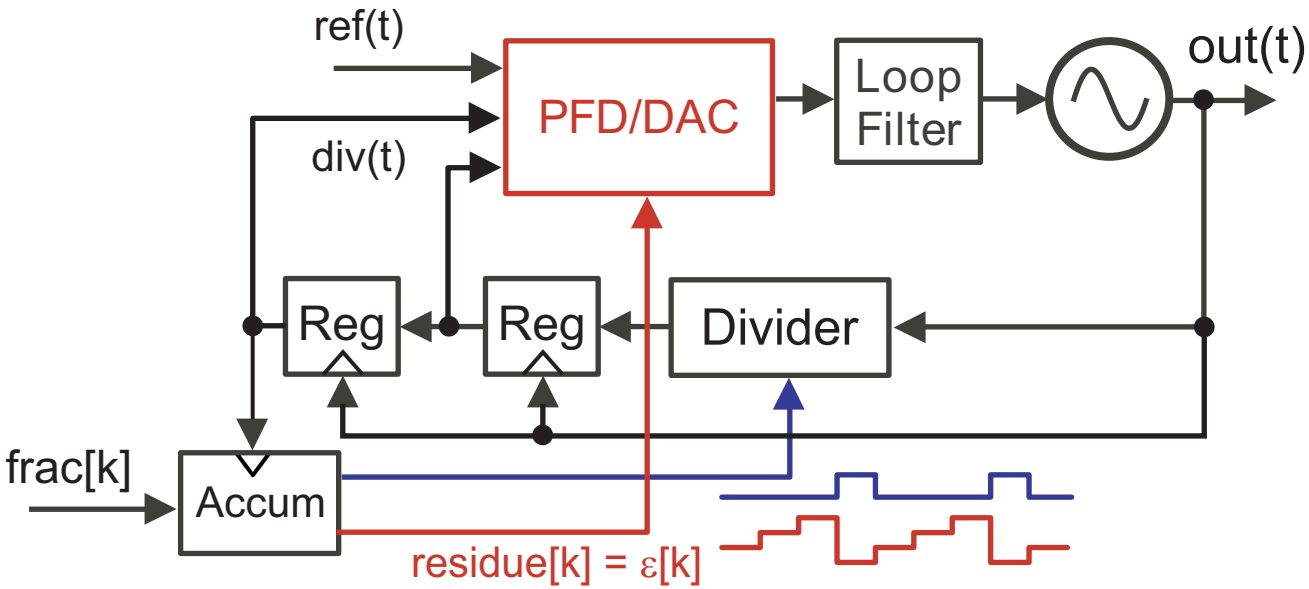
- **Eliminates issue of having non-square error pulse shapes**

For More Details on This Approach

- **Theory and simulations presented in TCAS II paper**
  - **Meninger and Perrott, TCASII, Nov 2003**



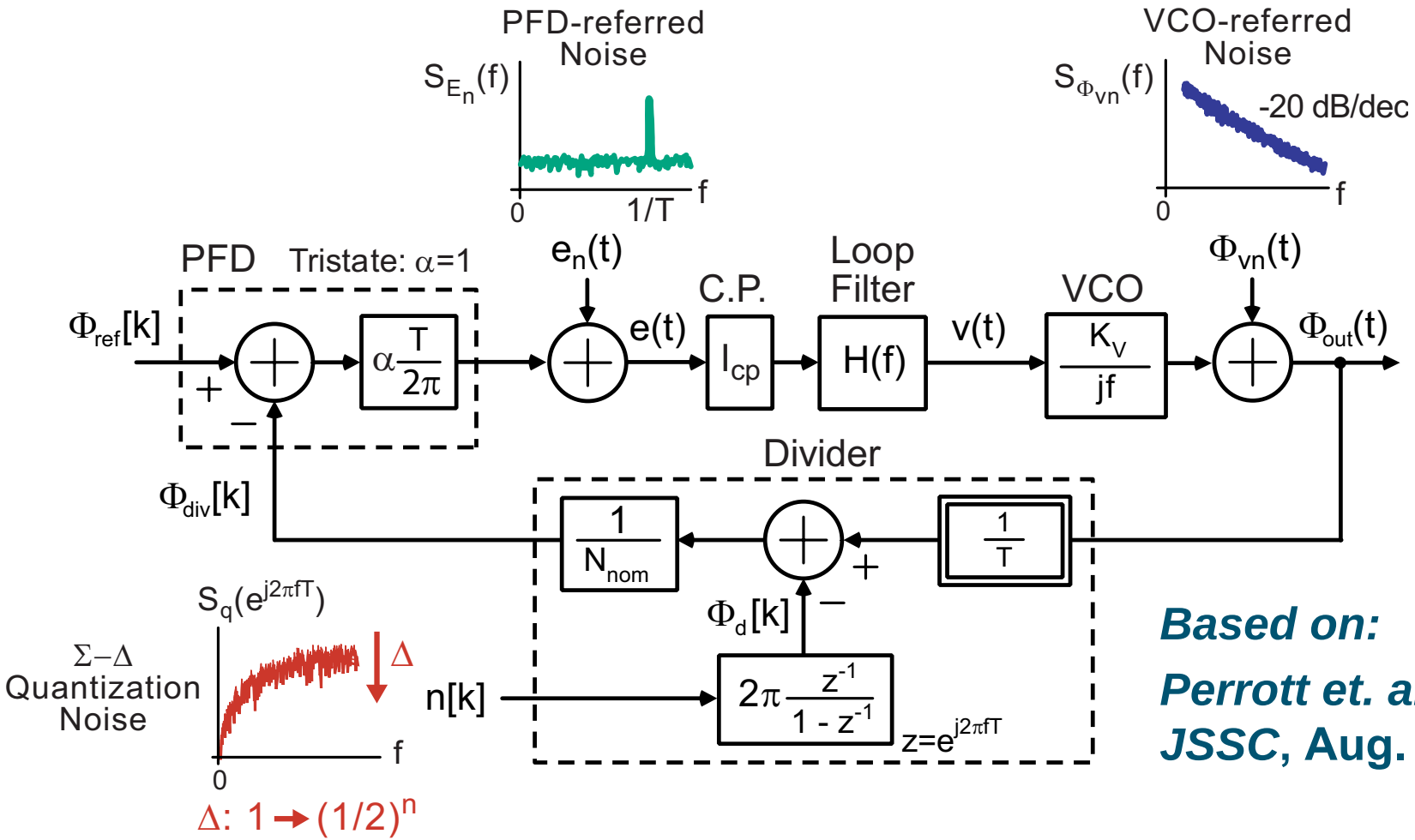
Design and Simulation of 'PFD/DAC' Synthesizer



- **Step 1: Derive analytical model**
- **Step 2: Design at system level**
- **Step 3: Simulate at system level (CppSim)**
- **Step 4: Simulate at transistor level (SPICE)**

Iterate between all of these steps in practice

Analytical Model of 'PFD/DAC' Fractional-N PLL

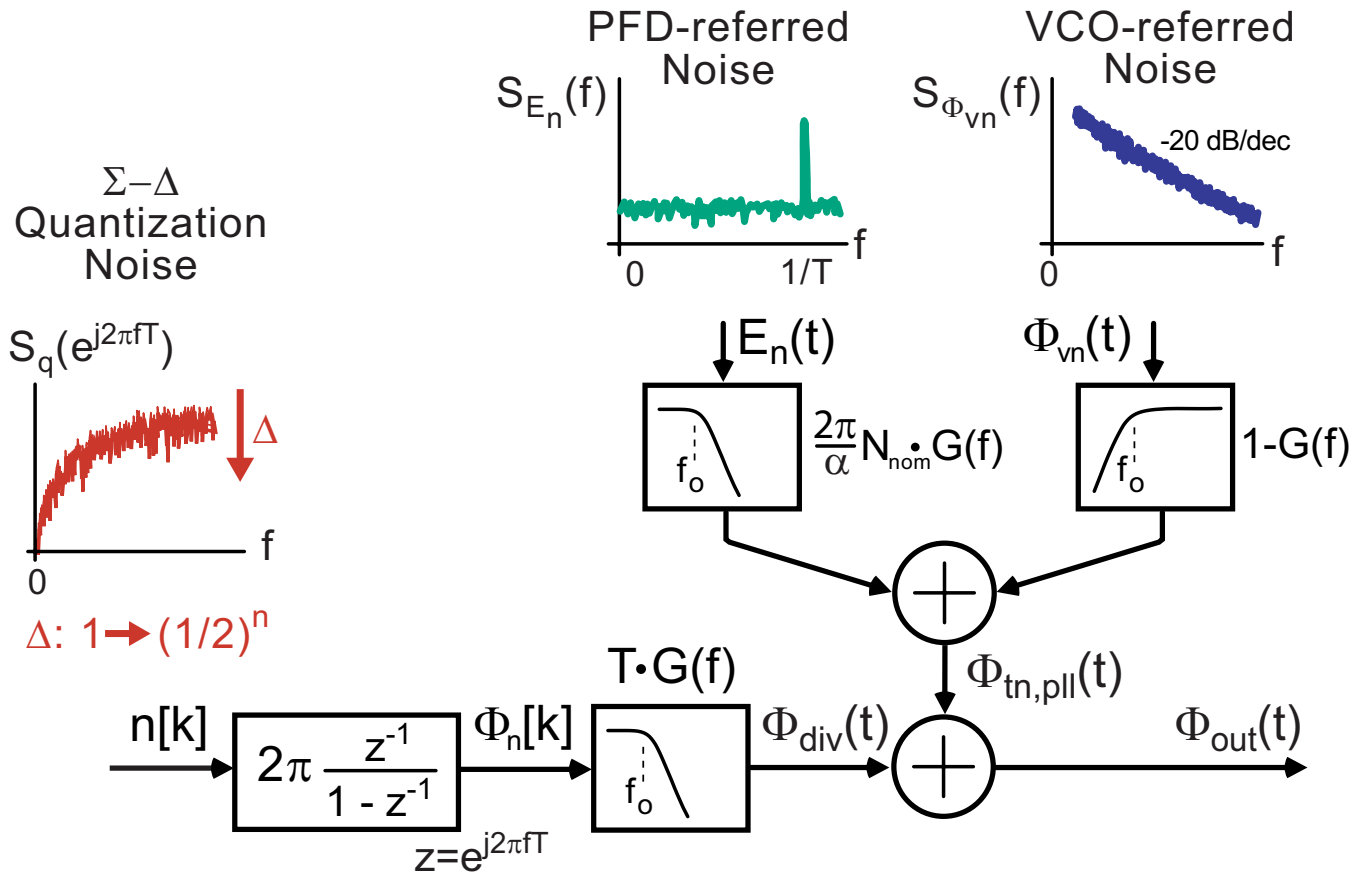


Based on:

Perrott et. al., JSSC, Aug. 2002

n-bit PFD/DAC → Δ is reduced from 1 to $(1/2)^n$

Parameterized PLL Model



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

Application:
A 1 MHz Bandwidth Fractional-N Frequency Synthesizer Implementation

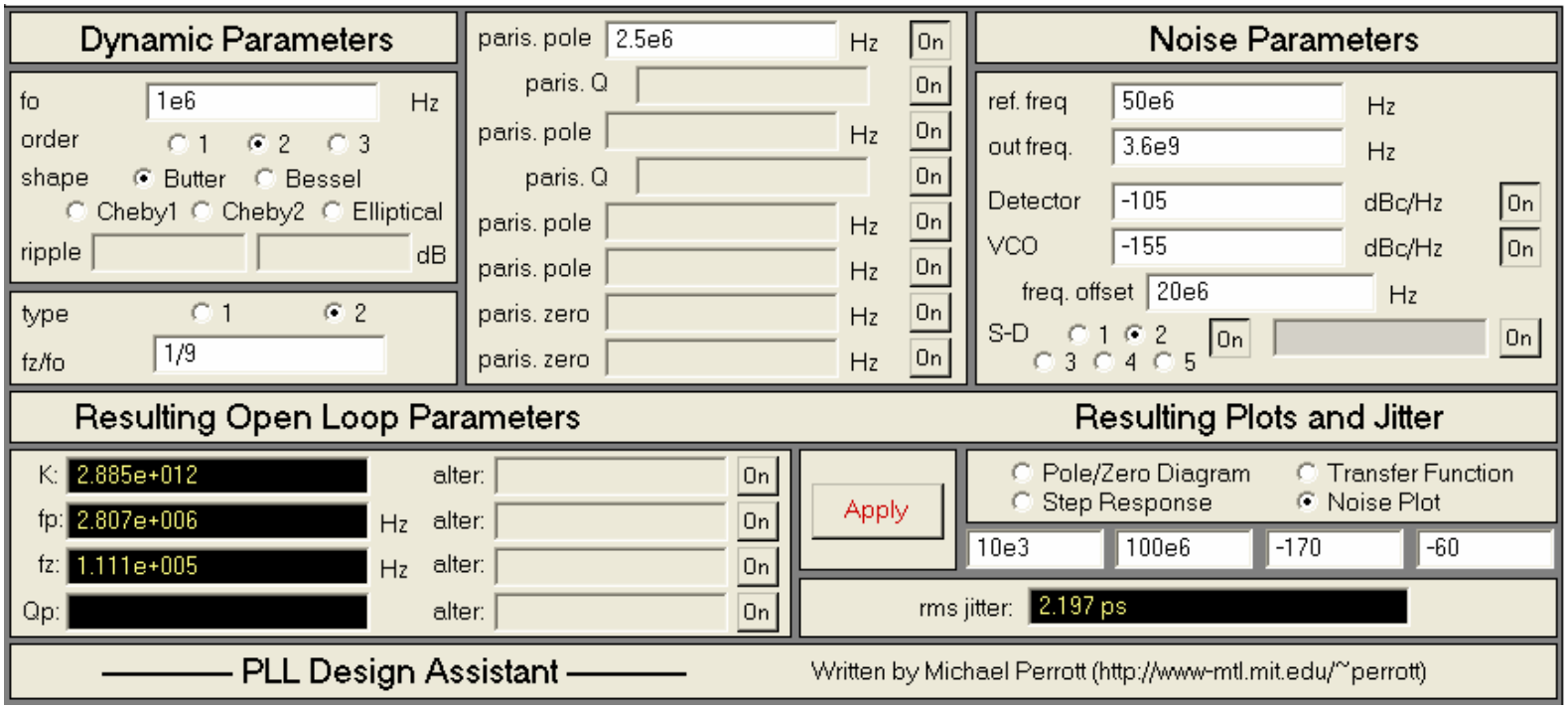
Design Goals

- **Output frequency: 3.6 GHz**
  - **Allows dual-band output (1.8 GHz and 900 MHz)**
- **Reference frequency: 50 MHz**
  - **Allows low cost crystal reference**
- **Bandwidth: 1 MHz**
  - **Allows fast settling time and ~1 Mbit/s modulation rate**
- **Noise: < -150 dBc/Hz at 20 MHz offset (3.6 GHz carrier)**
  - **Phase noise at the 20 MHz frequency offset is very challenging for GSM and DCS transmitters**
    - GSM: -162 dBc/Hz at 20 MHz offset (900 MHz carrier)
    - DCS: -151 dBc/Hz at 20 MHz offset (1.8 GHz carrier)

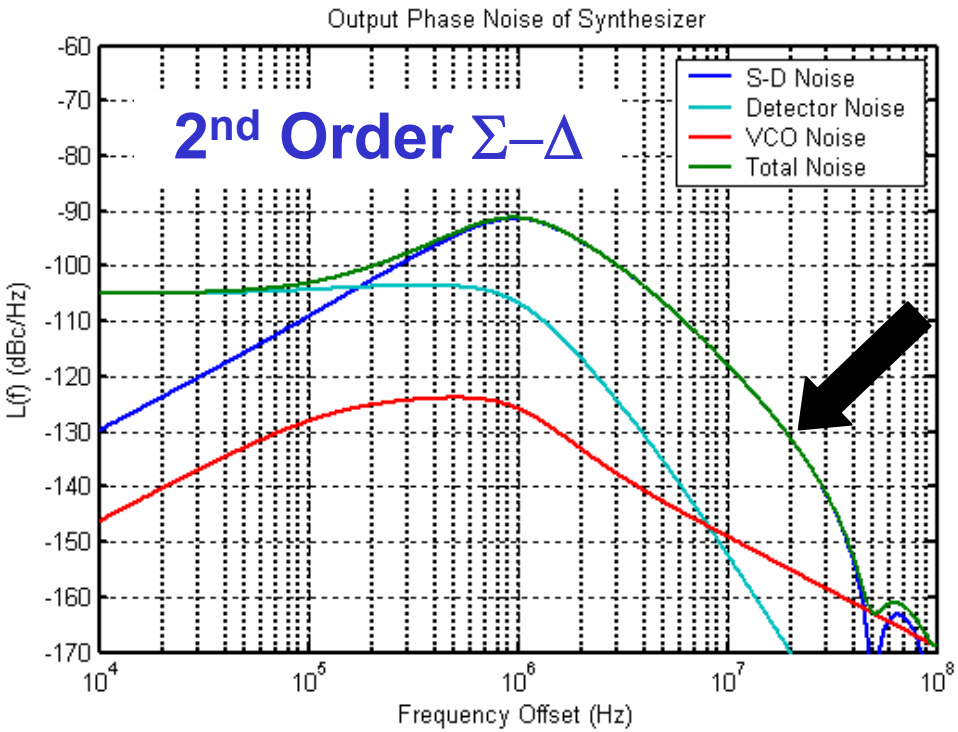
Simultaneous achievement of the above bandwidth and noise targets is very challenging

Evaluate Noise Performance with 1 MHz PLL BW

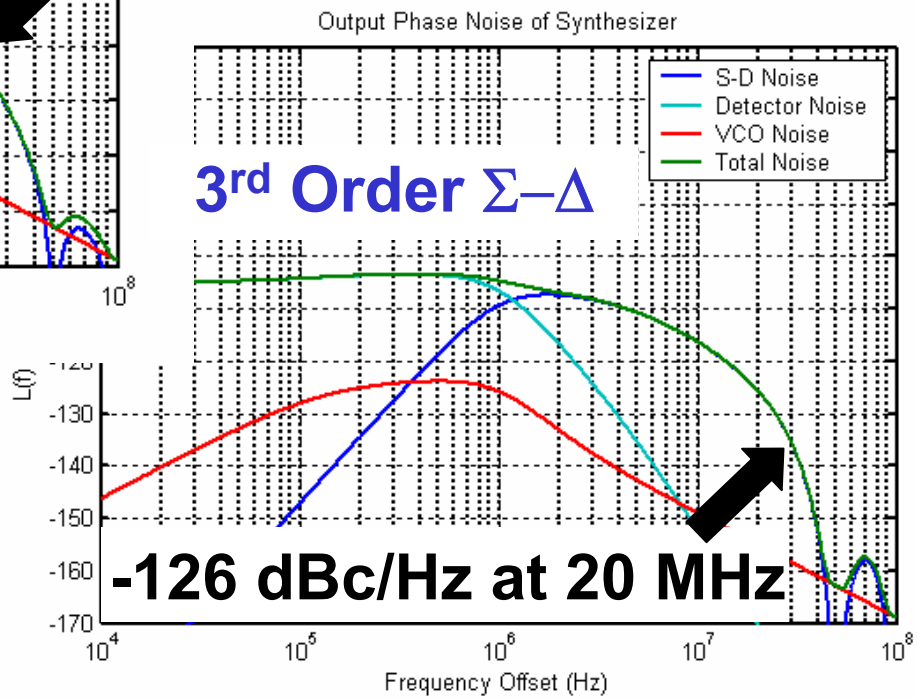
- G(f) parameters
  - 1 MHz BW, Type II, 2nd order rolloff, extra pole at 2.5 MHz
- Required PLL noise parameters (with a few dB of margin)
  - Output-referred charge pump noise: -105 dBc/Hz
  - VCO noise: -155 dBc/Hz at 20 MHz offset (3.6 GHz carrier)



Calculated Phase Noise for Classical Fractional-N

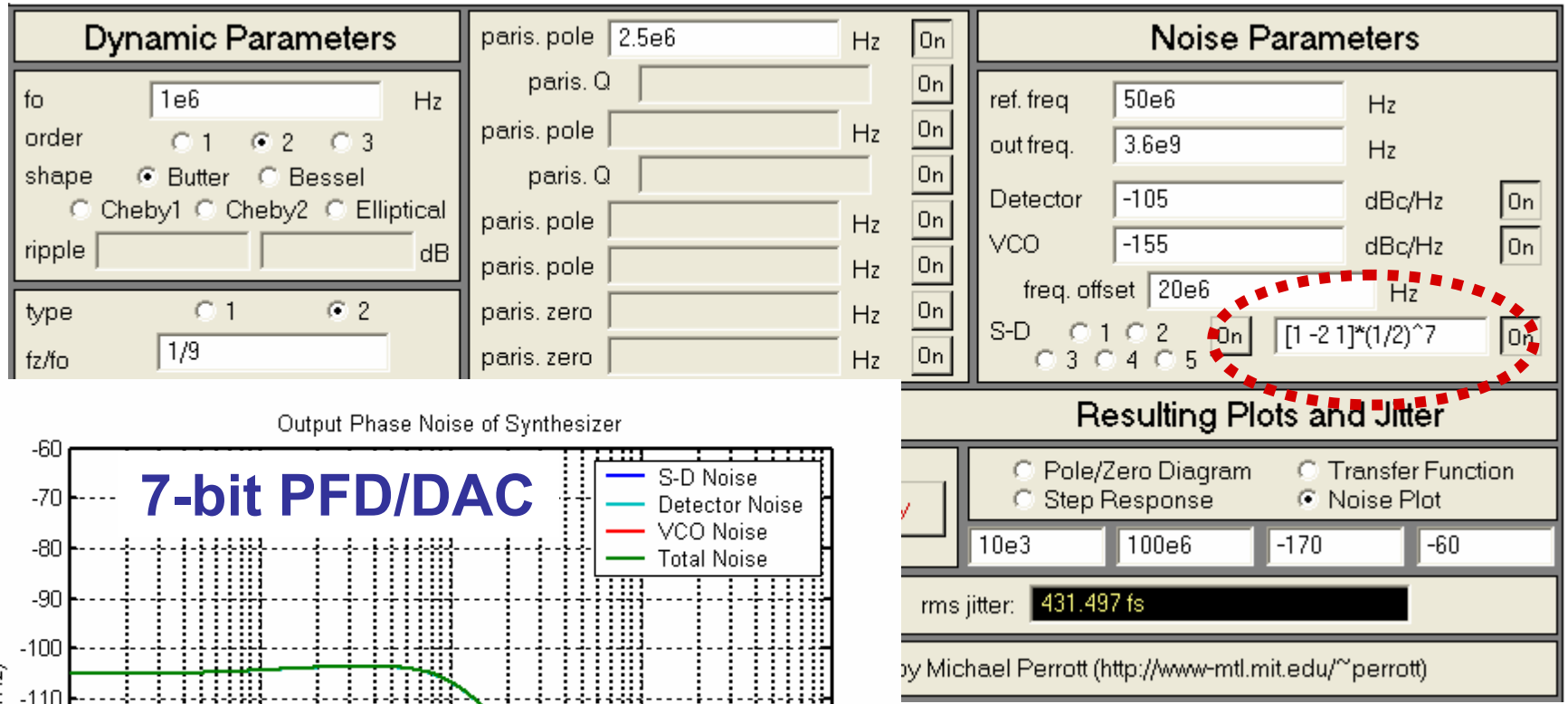


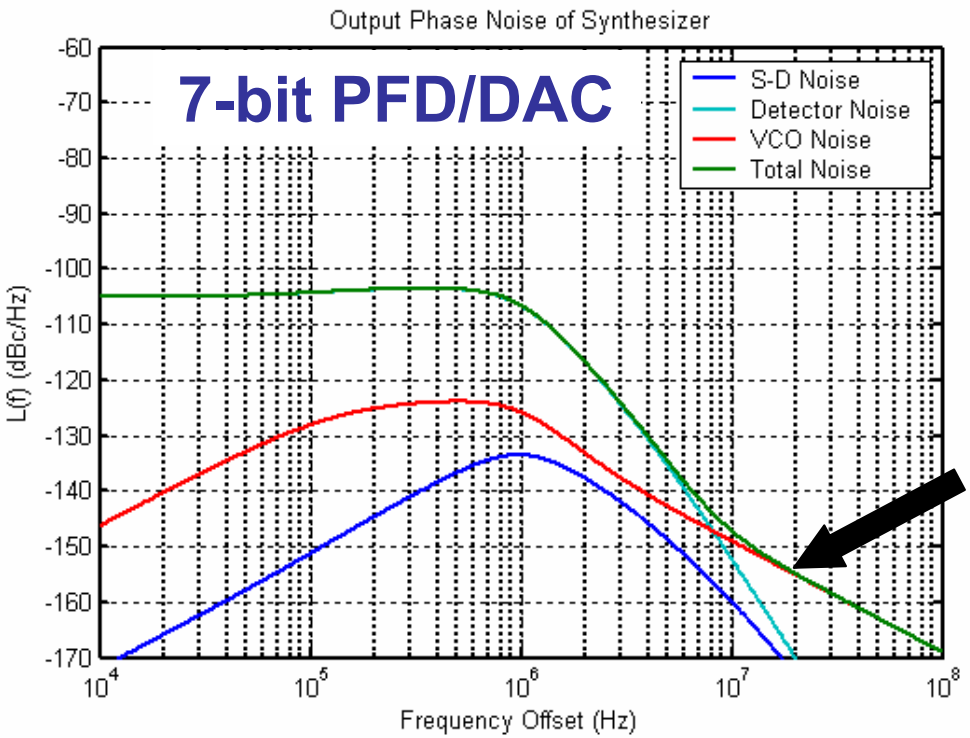
-132 dBc/Hz at 20 MHz



These do NOT meet our target of -150 dBc/Hz at 20 MHz (3.6 GHz carrier freq.)

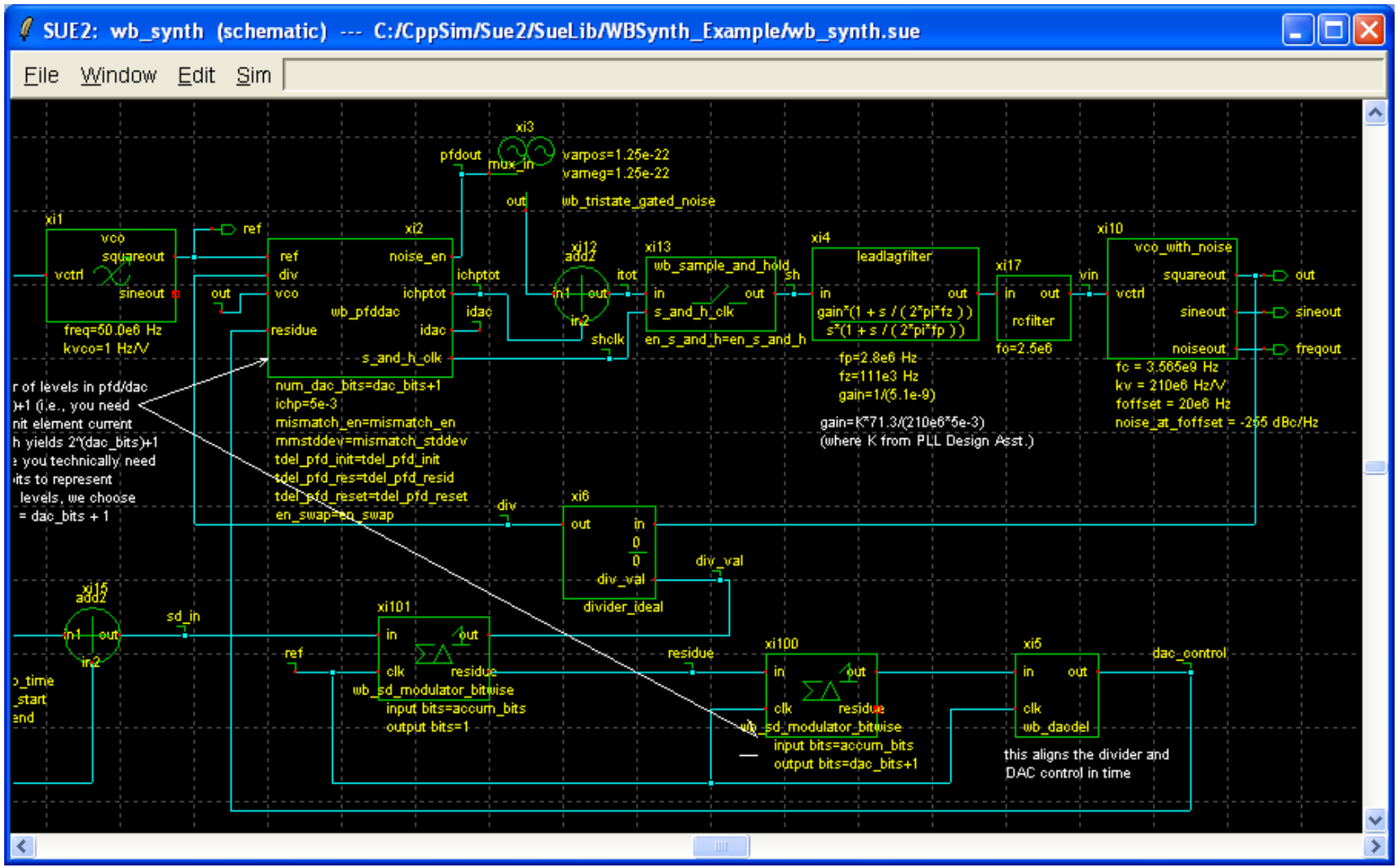
Calculated Phase Noise for 7-bit PFD/DAC Synth





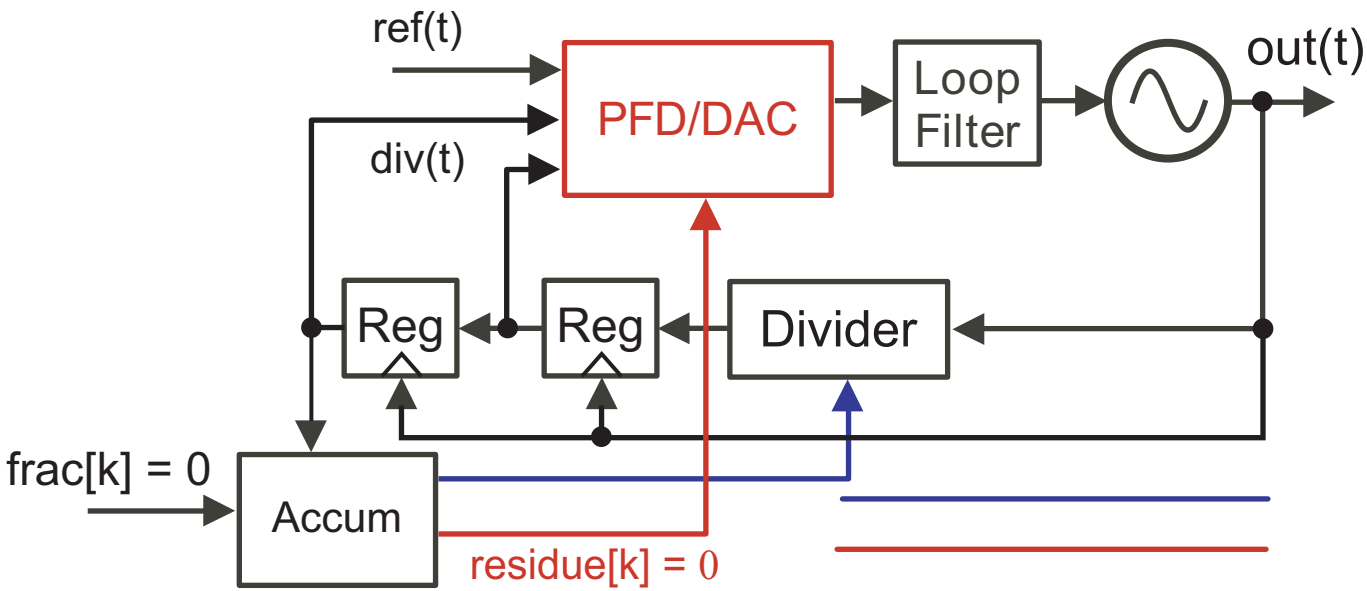
-155 dBc/Hz at 20 MHz !

Simulation of PFD/DAC Synthesizer using CppSim



- **Phase noise plots to follow: 40e6 time steps in 10 min**

Intrinsic PLL Noise Performance



- **Set fractional portion of divide value to zero**
  - **Leads to residue variation of zero**
    - No quantization noise!
- **Need to calculate and simulate impact of detector and VCO noise**

In essence, operate as integer-N synthesizer

Calculate Intrinsic PLL Noise Sources

- **Estimate detector noise (dominated by charge pump)**
  - **From SPICE Simulation, $S_{Icp}(f)$ = Duty*3e-22 A^2/Hz**
  - **Output-referred PLL noise due to above noise:**

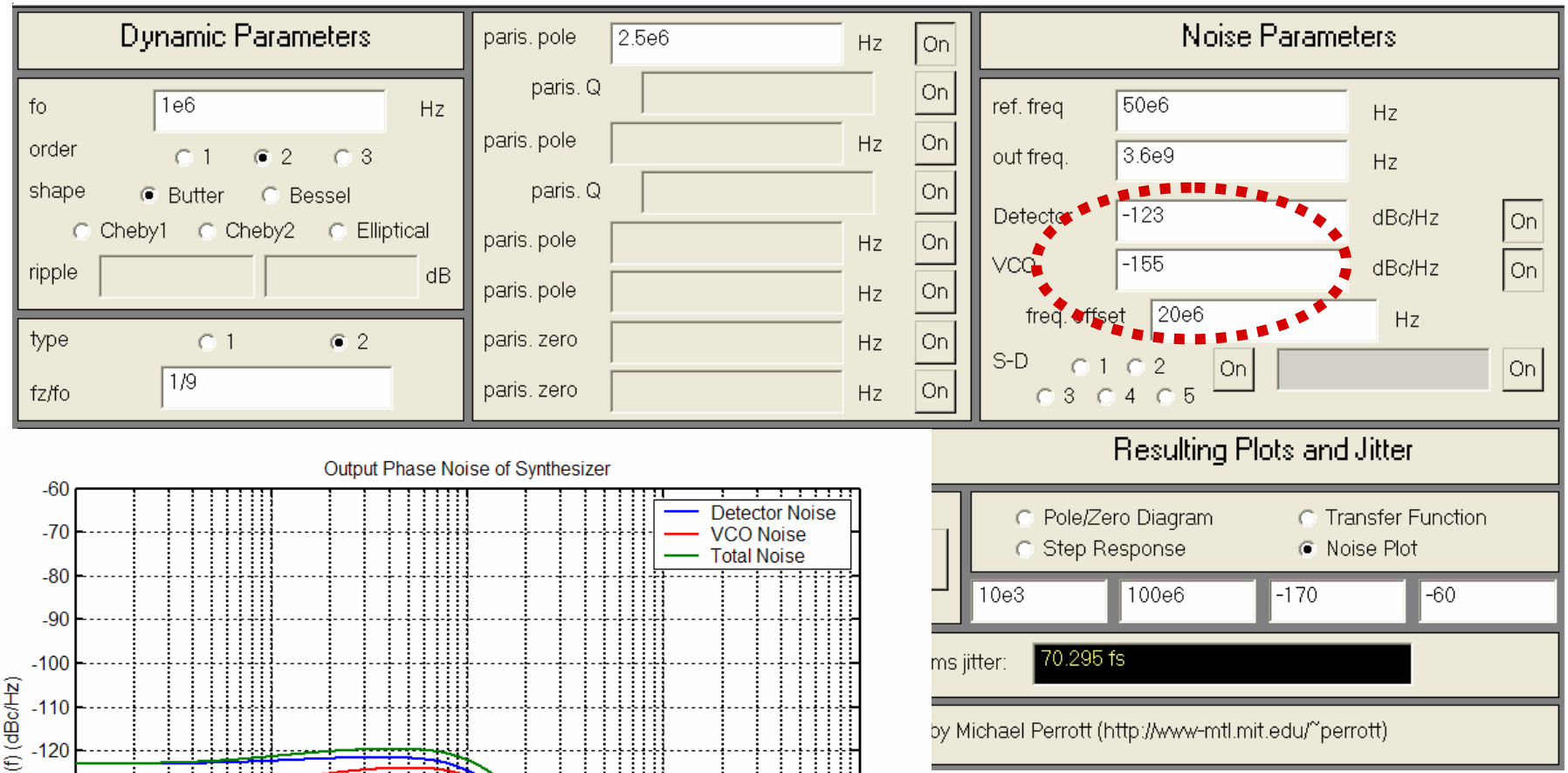
$$S_{\Phi_{out}}(f) = \left(\frac{1}{I_{cp}}\right)^2 \left(\frac{2\pi}{\alpha} N_{nom}\right)^2 |G(f)|^2 S_{Icp}(f)$$

$$= \left(\frac{1}{5 \cdot 10^{-3}}\right)^2 \left(\frac{2\pi}{1} 71.3\right)^2 |G(f)|^2 \, 0.2 \cdot 3 \cdot 10^{-22}$$

$$\Longrightarrow 10 \log(S_{\Phi_{out}}(f)) = -123 \, |G(f)|^2 \, \text{dBc/Hz}$$

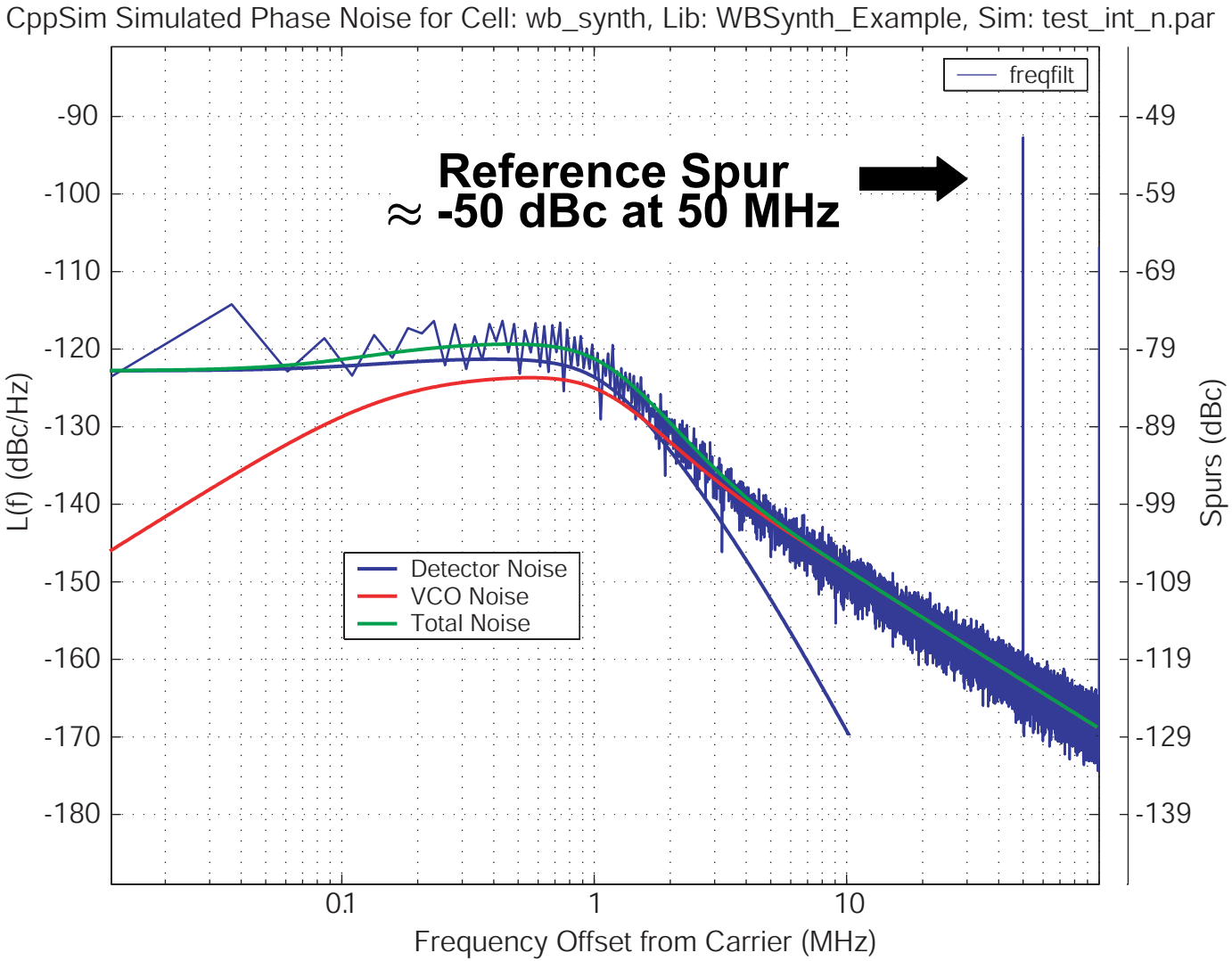
- **Estimate VCO noise**
  - **For off-chip VCO, examine data sheet:**
    - In this case, $S_{\Phi vco}$ = -155 dBc/Hz at 20 MHz offset
  - **For on-chip VCO, use Spectre RF or other CAD tool**

Calculate PLL Noise Due to Intrinsic Noise Sources



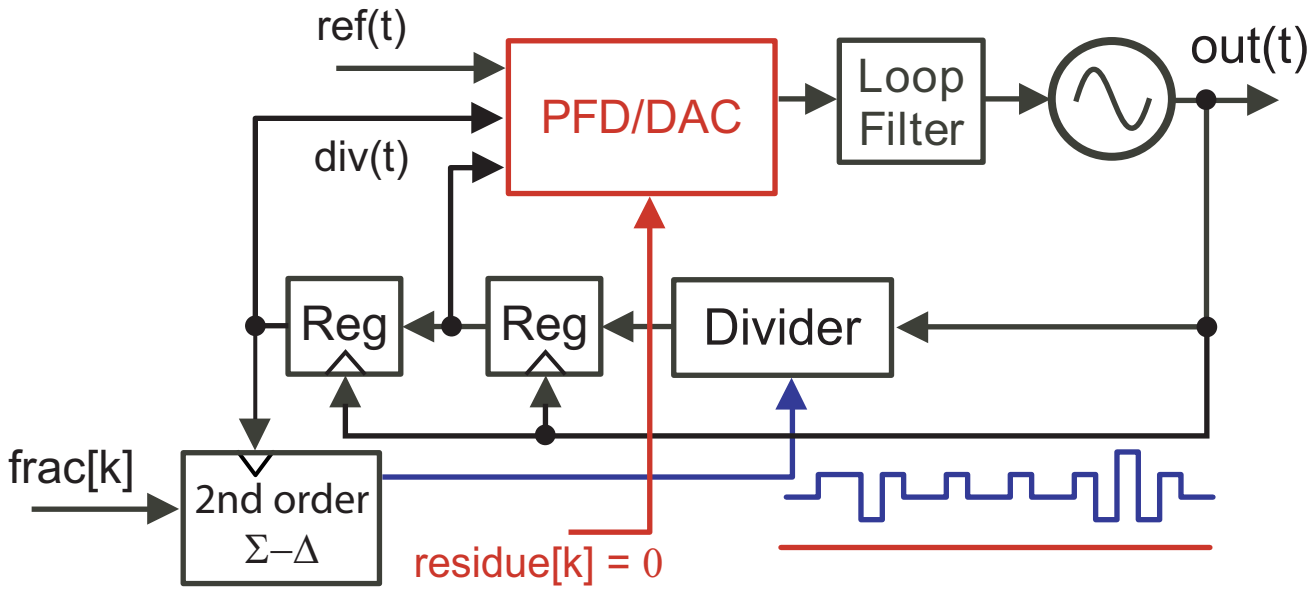
- We will see that we will need to include:
  - Reference noise
  - 1/f noise

Simulated Phase Noise due to Intrinsic Noise Sources



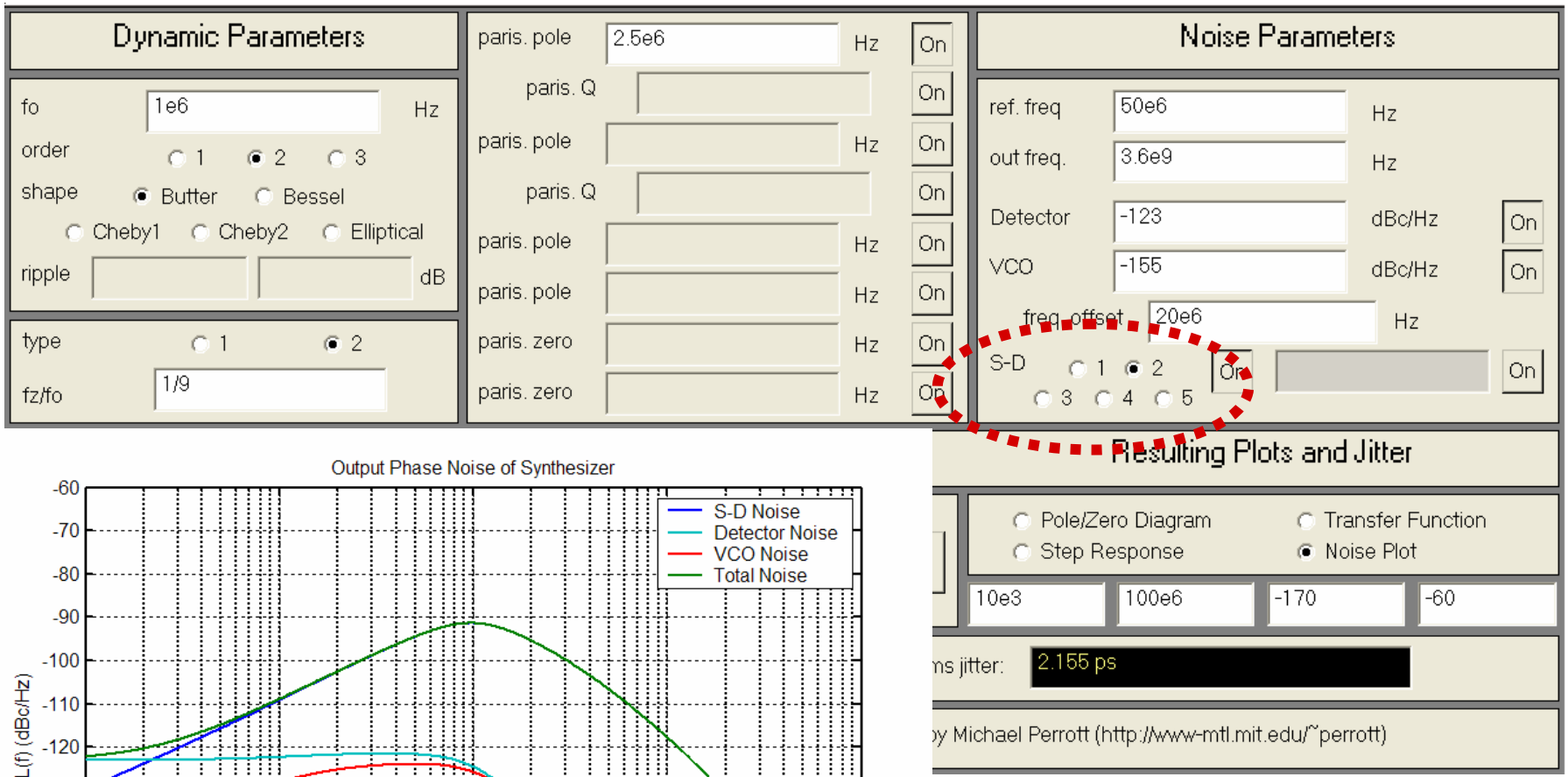
- **PLL Design Assistant accurately models simulated noise!**

2nd Order Σ–Δ Fractional-N Performance



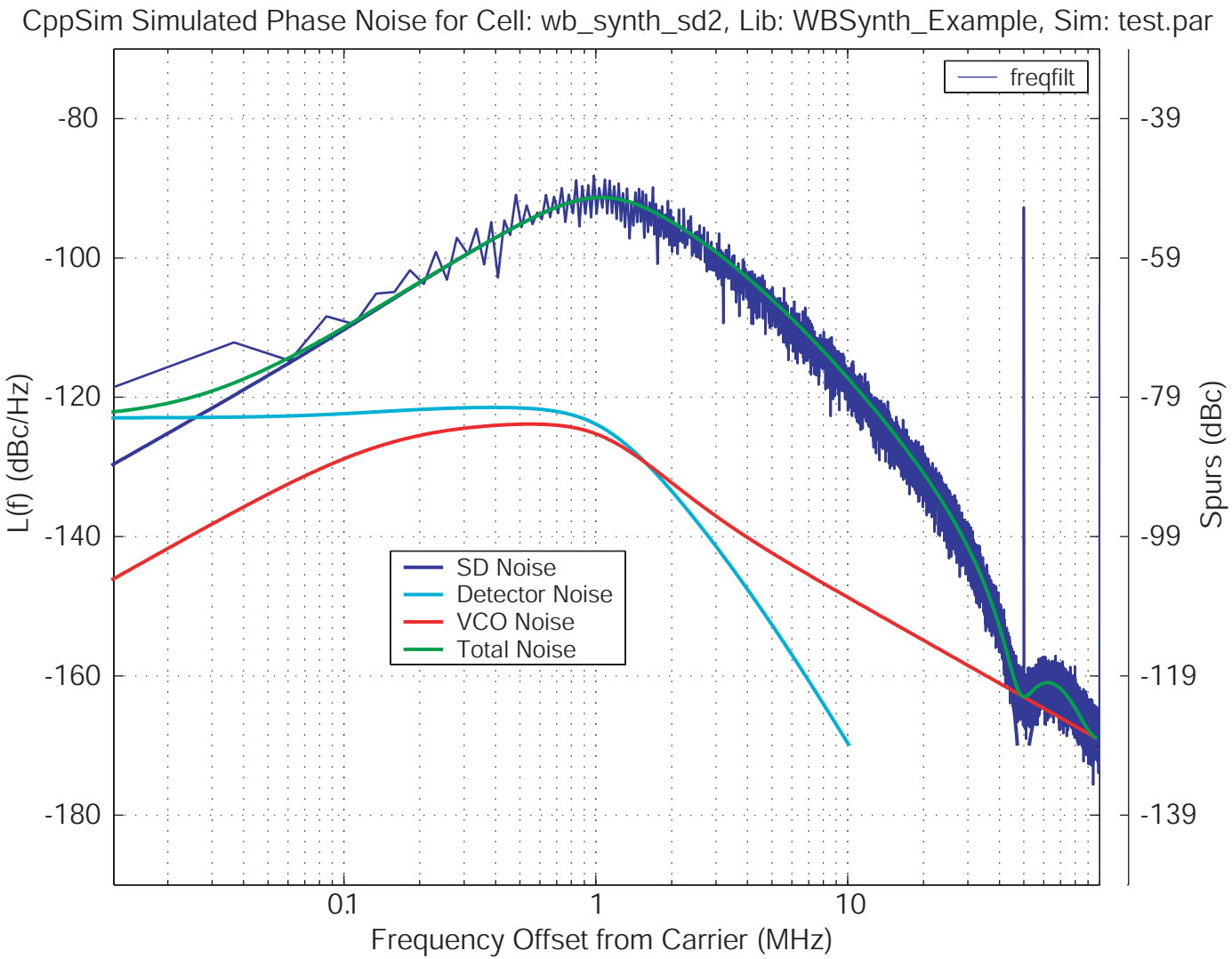
- Replace accumulator with second order Σ–Δ modulator
- Set residue into PFD/DAC equal to zero

Calculate PLL Noise for 2nd Order Σ–Δ Synthesizer



- 2nd order Σ–Δ
  - Click on 2nd order S-D quantization noise in tool

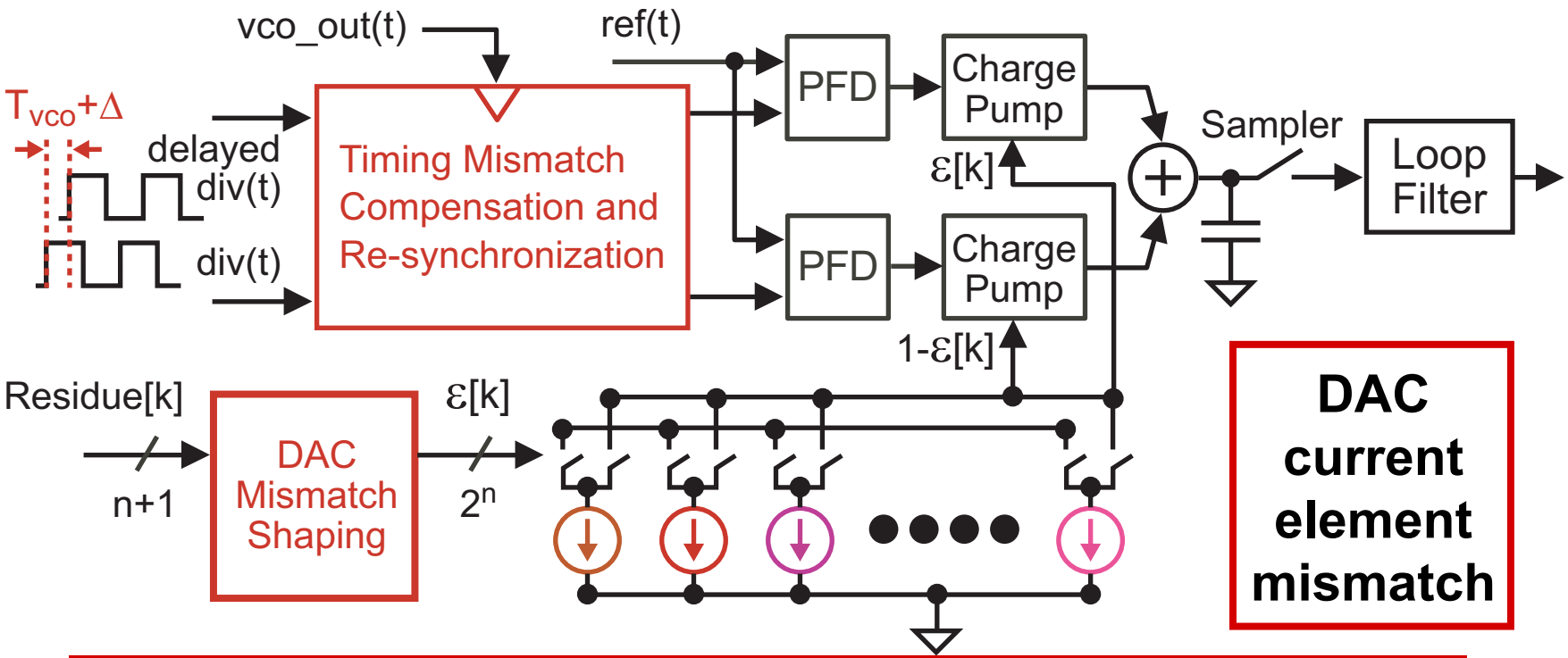
Simulated Phase Noise of 2nd Order Σ–Δ Synthesizer



- **PLL Design Assistant accurately models simulated noise!**

7-bit PFD/DAC Synthesizer Performance

Delay mismatch



Application of proposed noise scrambling/shaping techniques leads to broadband noise from delay and DAC current mismatch

Impact on PLL Noise due to Non-idealities of PFD/DAC

- **Impact of DAC mismatch**
  - **Lowers achievable quantization noise suppression**
  - **Negligible in this case**
- **Impact of delay mismatch**
  - **Model as white reference noise uniformly distributed from 0 to Δt**
    - Calculation for $\Delta t = 5$ ps:

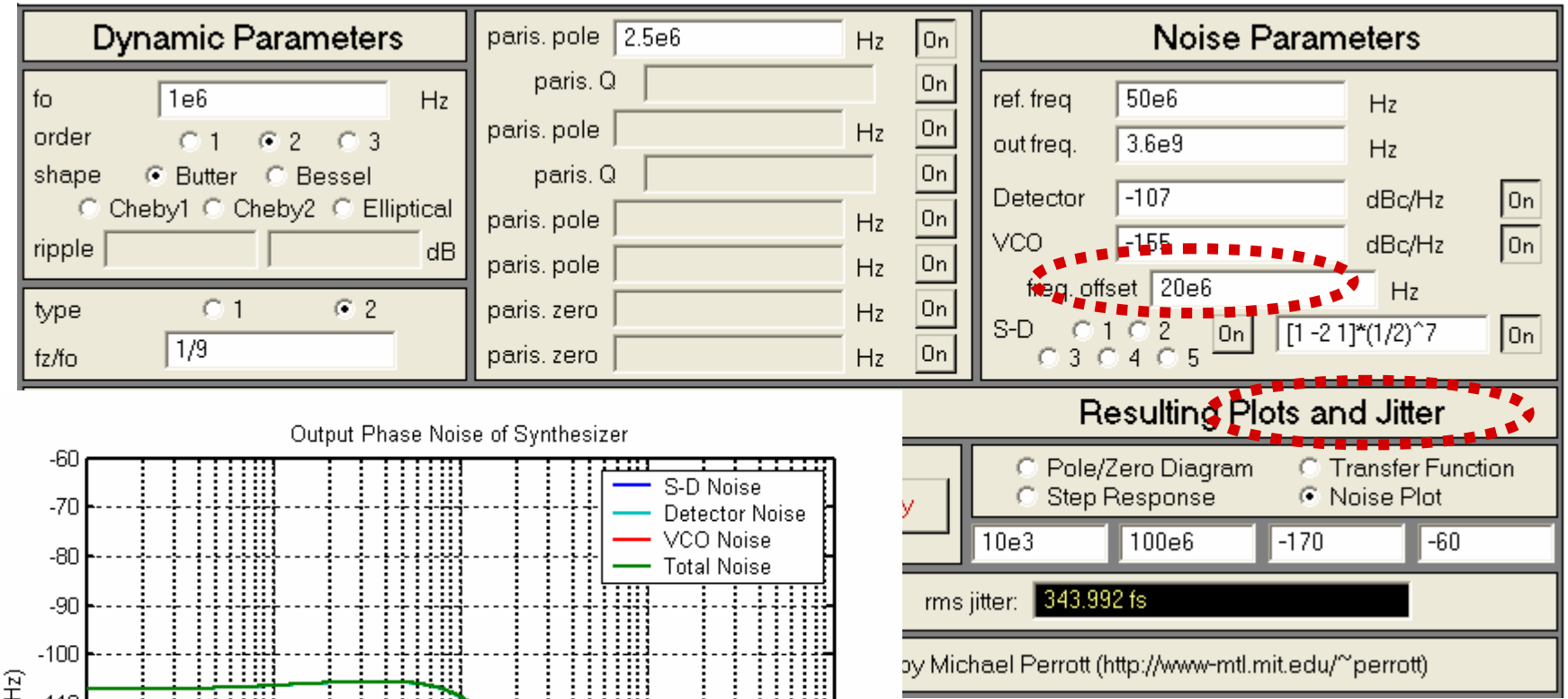
$$S_{\Phi_{out}}(f) = \frac{1}{T}\left|TN_{nom}G(f)\right|^2 S_{\Phi_{jit}}(e^{j2\pi fT})$$

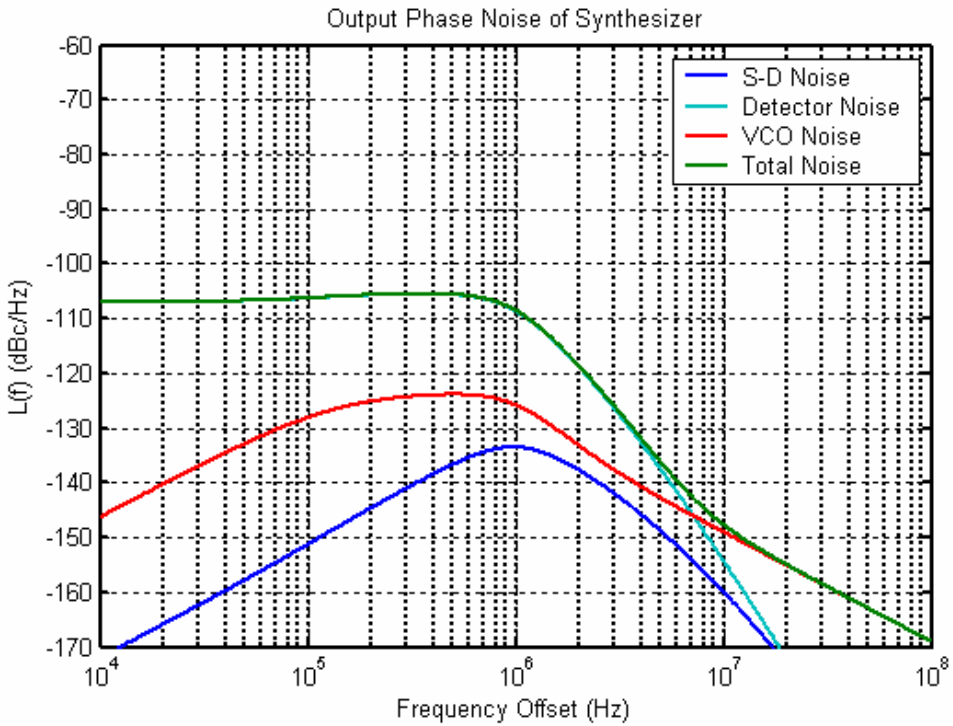
$$= \frac{1}{T}\left|TN_{nom}G(f)\right|^2 \left|\frac{2\pi}{T}\right|^2 \frac{(\Delta t)^2}{12}$$

$$= (50\ \text{MHz})(71.3)^2(2\pi)^2\frac{(5\text{ps})^2}{12}$$

$$\Longrightarrow 10\log(S_{\Phi_{out}}(f)) = -107\left|G(f)\right|^2 \text{dBc/Hz}$$

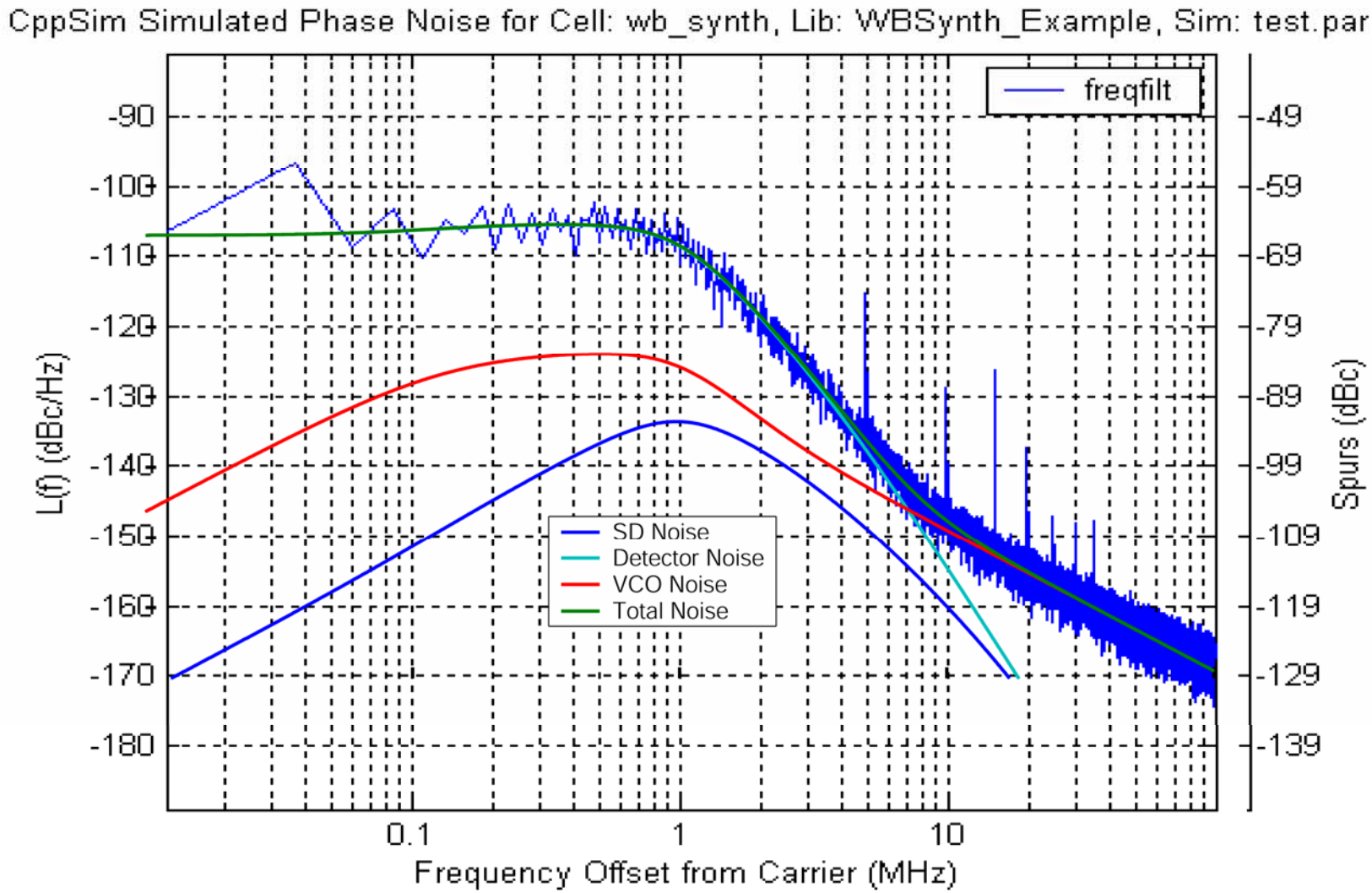
Calculate PLL Noise for 7-bit PFD/DAC Synthesizer





- PFD/DAC
  - Adjust S-D Quant. Noise
- Delay mismatch
  - Adjust Detector noise

Simulated PLL Phase Noise of 7-bit PFD/DAC



- **PLL Design Assistant accurately models simulated noise!**

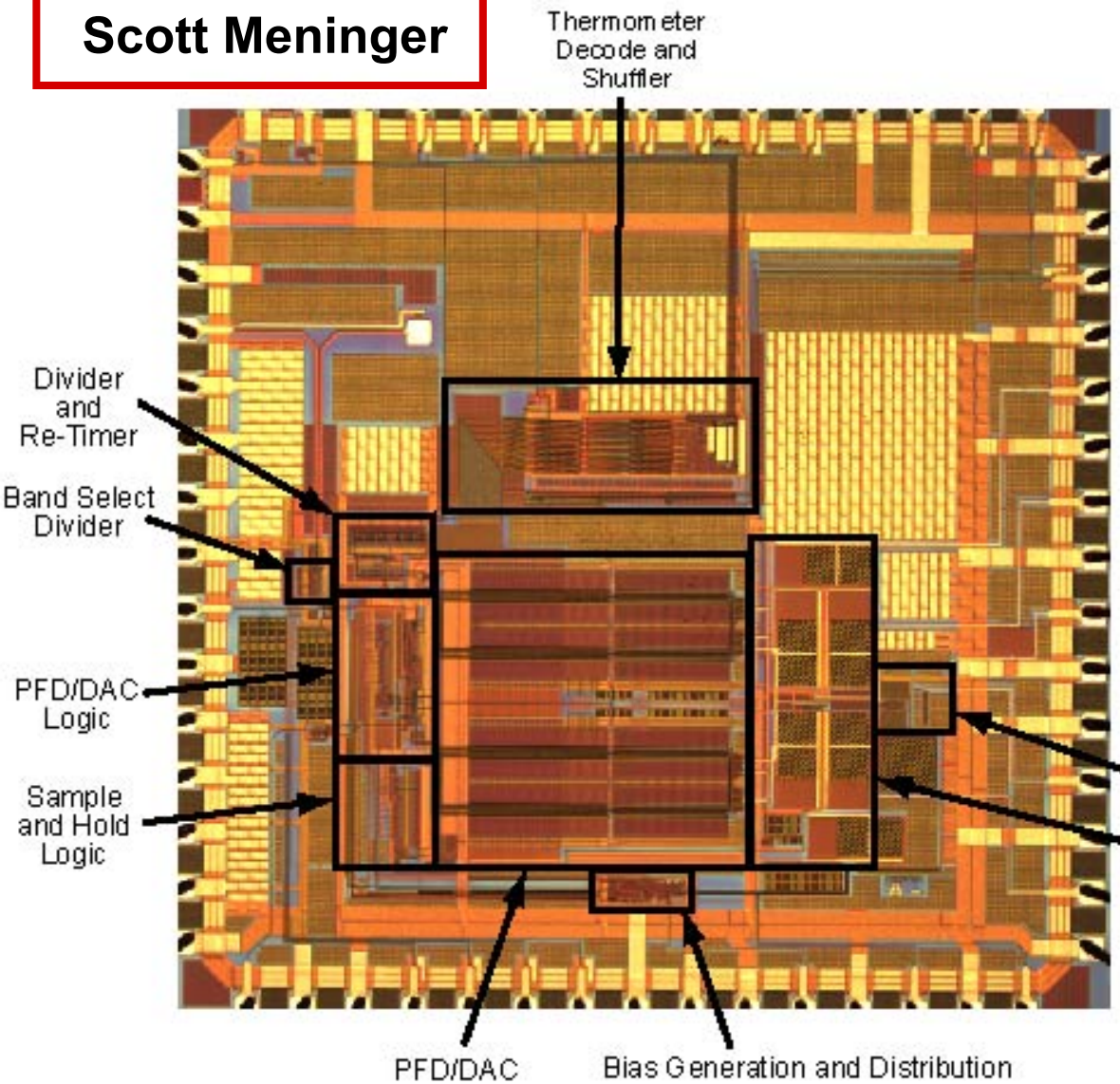
Summary of Design/Simulation Results

- **The PLL Design Assistant can be used to model the impact of**
  - **Intrinsic PLL noise sources**
  - **Quantization noise due to Σ–Δ dithering of divide value**
  - **Suppression of quantization noise by n-bit PFD/DAC**
  - **Impact of delay and current mismatch on PLL phase noise**
- **CppSim simulations confirm the accuracy of the above analysis**

How do PLL Design Assistant calculations compare to measured results?

A 1 MHz BW Fractional-N Frequency Synthesizer IC

Scott Meninger

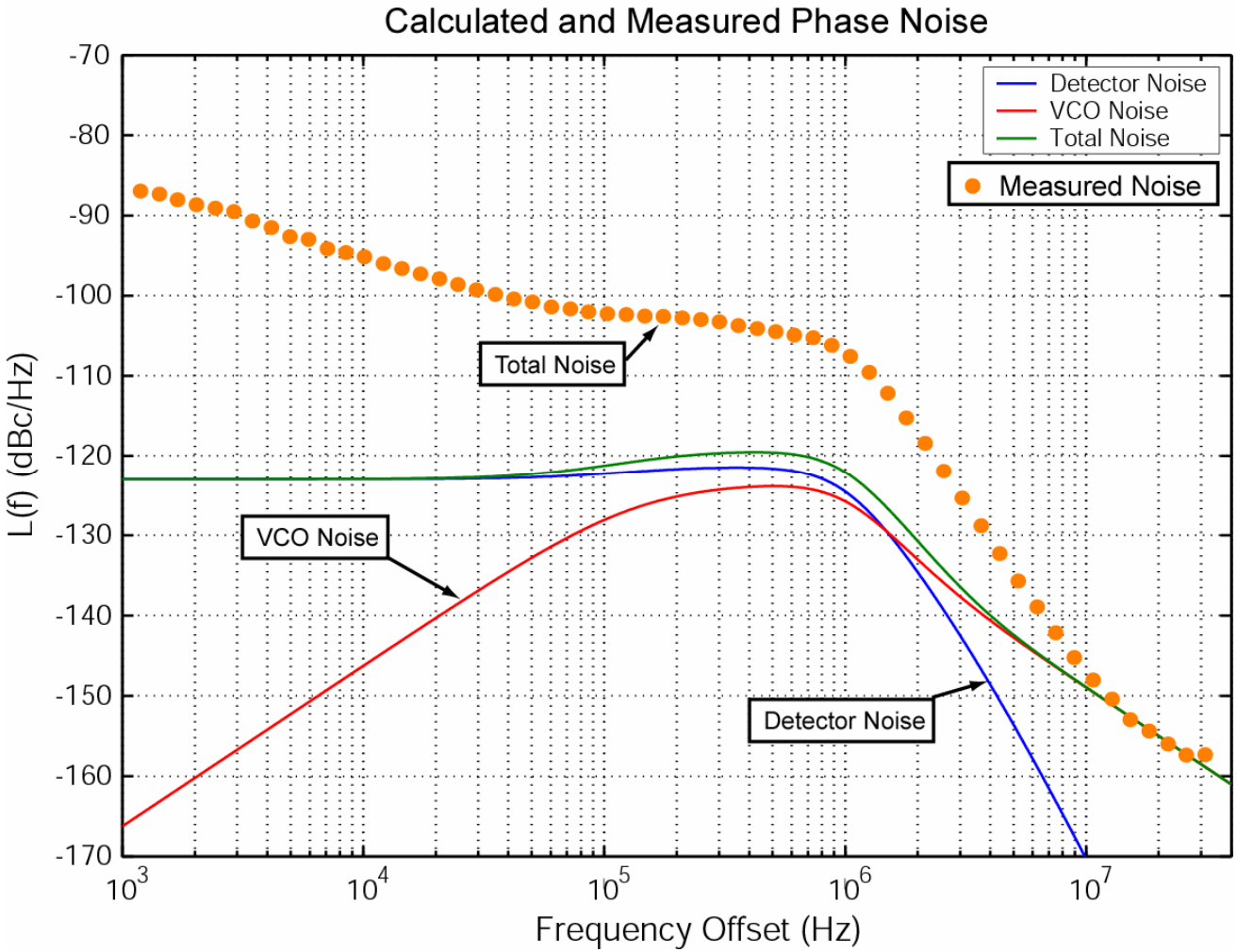


- **Implements proposed 7-bit PFD/DAC structure**
  - **0.18u CMOS**
  - **Circuit details to be published in the future**

Funded by MARCO C2S2

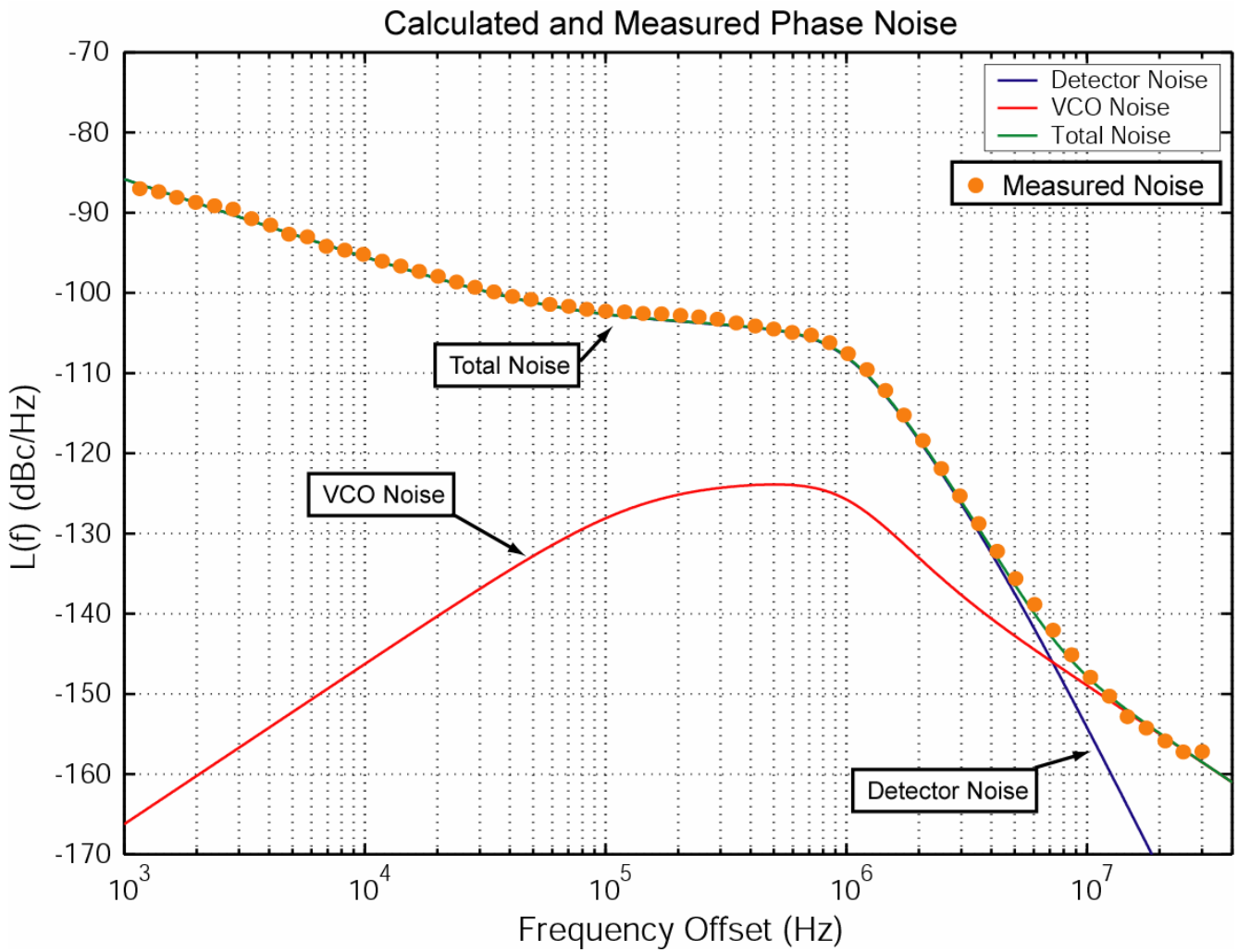
Fabricated by National Semiconductor

Measured vs Calculated Phase Noise (Integer-N)



- **Calculated noise is way off!**
  - **Issue: we did not consider reference jitter and 1/f noise**

Adjustment of Calculations to Fit Measured Result



- **Calculated noise now assumes:**
  - **Detector noise is -107 dBc/Hz with 1/f corner of 130 kHz**

Back Extraction of Reference Jitter

$$S_{\Phi_{out}|REF}(f) = \overline{\Delta t_{jitt}^2} \cdot \frac{1}{T} \cdot \left(\frac{2\pi}{T}\right)^2 \cdot (N_{nom}T)^2 \cdot |G(f)|^2 \cdot \frac{i_{up}}{i_{down}}$$

Accounts for PFD structure with reduced i_{up}

- **Assuming G(f) = 1:**

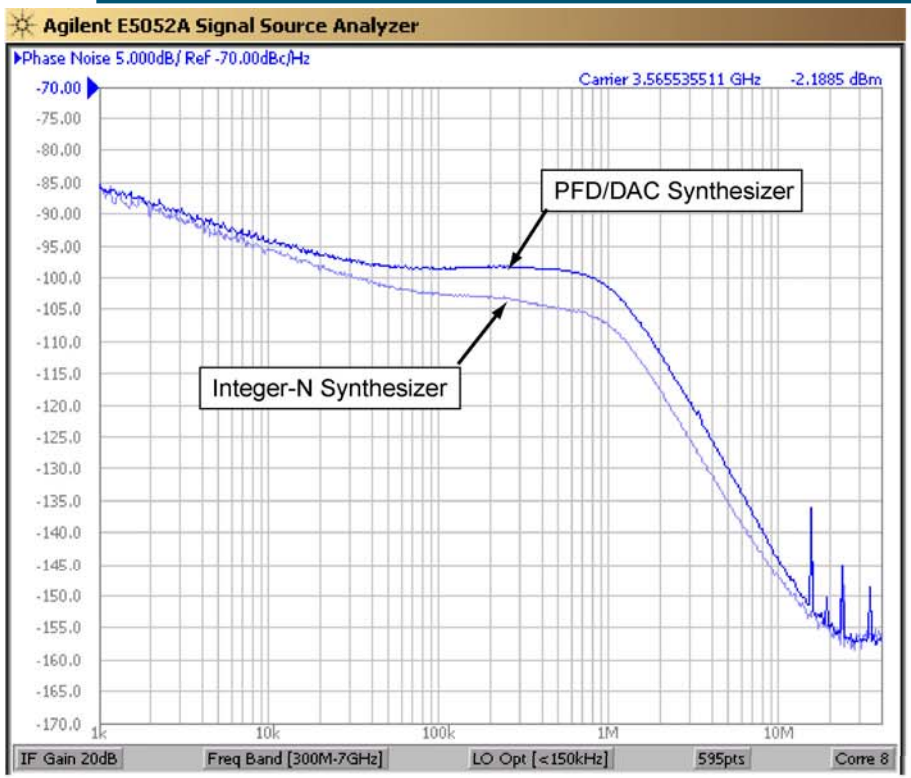
$$S_{\Phi_{out}|REF}(f) = 10^{(-107/10)}$$

$$\Rightarrow \overline{\Delta t_{jitt}^2} = S_{\Phi_{out}|REF} \cdot \frac{T}{(2\pi N_{nom})^2} \cdot \frac{i_{down}}{i_{up}}$$

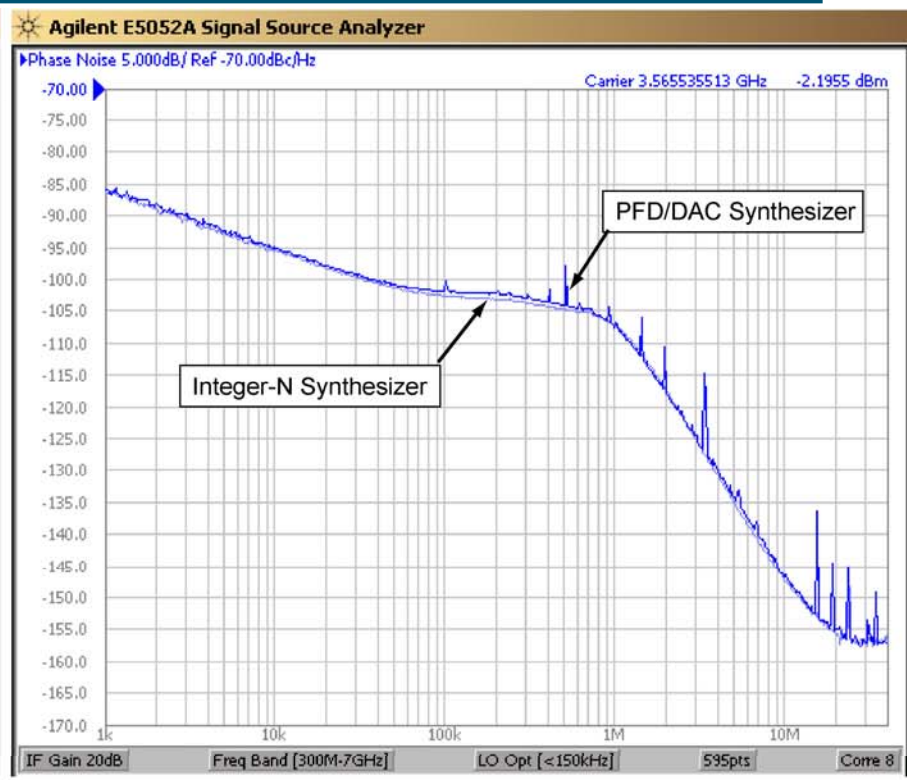
- **Assuming N_{nom} = 73, T = 1/50MHz, i_{up}/i_{down} = 1/5**

$$\Rightarrow \Delta t_{jitt} = 3.08\text{ps}$$

7-bit PFD/DAC Synthesizer Vs Integer-N Configuration



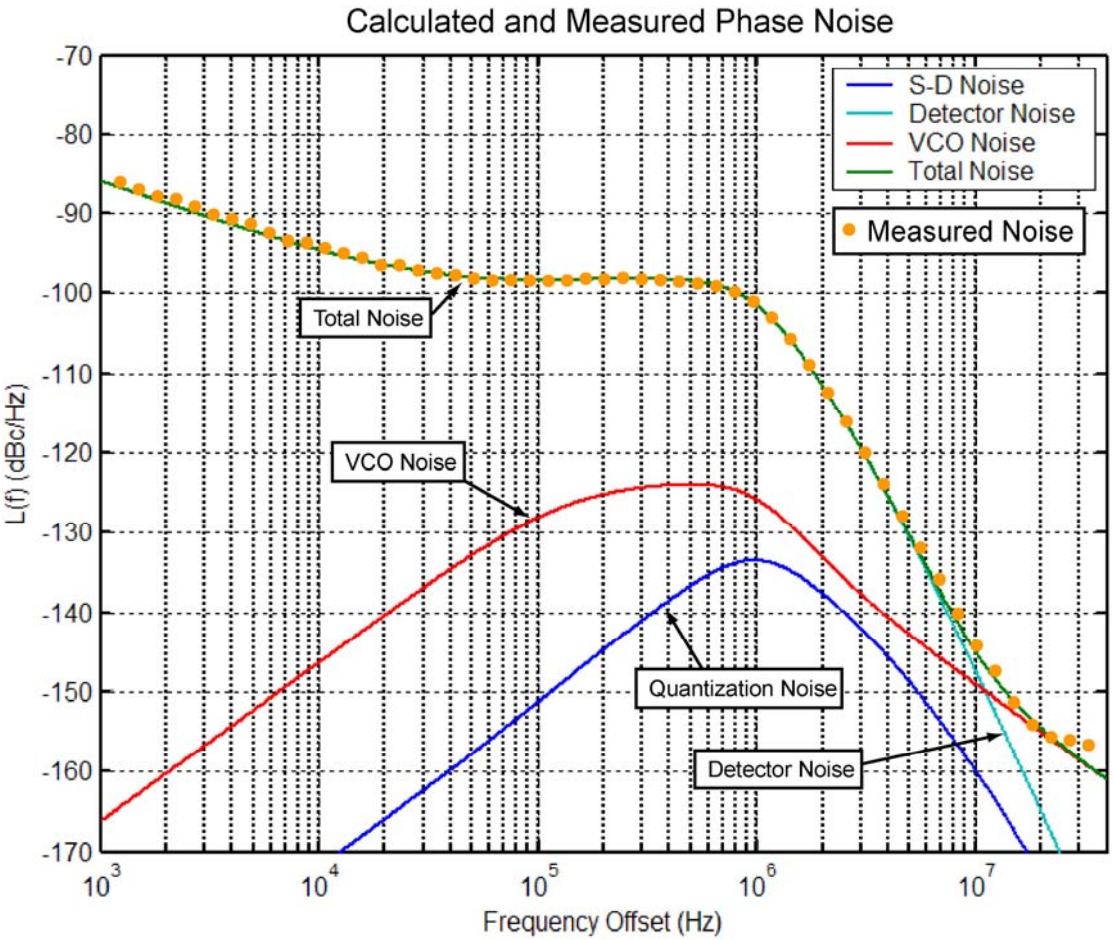
Phase Swapping Enabled



Phase Swapping Disabled

- **Left: Phase swapping enabled**
  - **Timing mismatch converted into broadband noise**
- **Right: Phase swapping disabled**
  - **More fractional spurs, lower broadband noise**

Adjustment of Calculations to fit Measured Results



- Calculated noise now assumes:
  - Detector noise is -100 dBc/Hz with 1/f corner of 20 kHz

Back Extraction of Timing Mismatch Using the Model

$$S_{\Phi_{out}|\Delta_t} = \frac{1}{T}\frac{\overline{\Delta_{t2}^2}}{12}(2\pi N_{nom})^2 \cdot |G(f)|^2$$

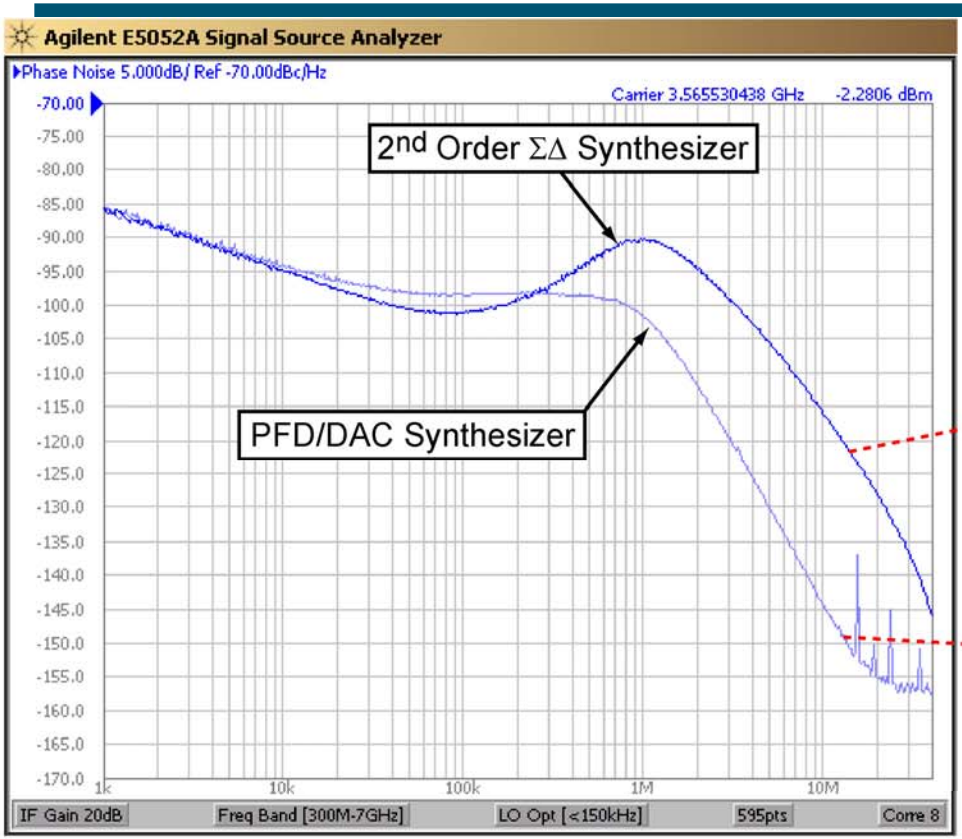
- **Assuming G(f) = 1:**

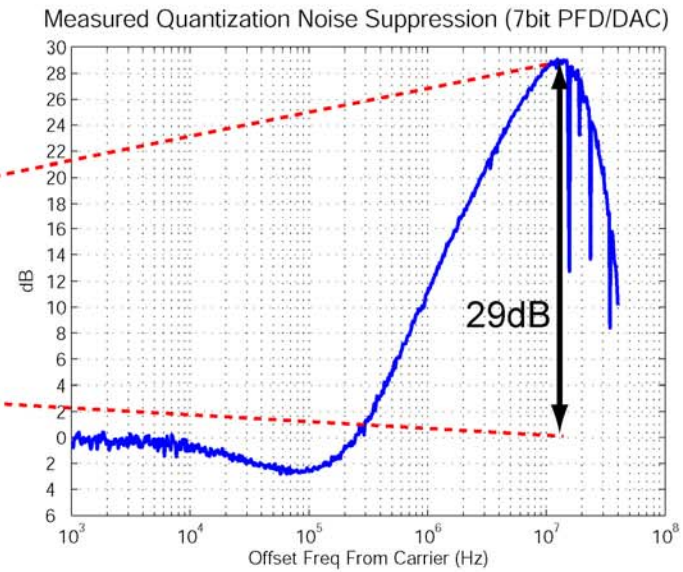
$$S_{\Phi_{out}|\Delta_t} = 10^{(-100/10)}$$

$$\Rightarrow \overline{\Delta_{t2}^2} = S_{\Phi_{out}|\Delta_t} \cdot \frac{12T}{(2\pi N_{nom})^2}$$

- **Assuming N_{nom} = 73, T = 1/50MHz $\Rightarrow \Delta_{t2}$ = 10.7 ps**

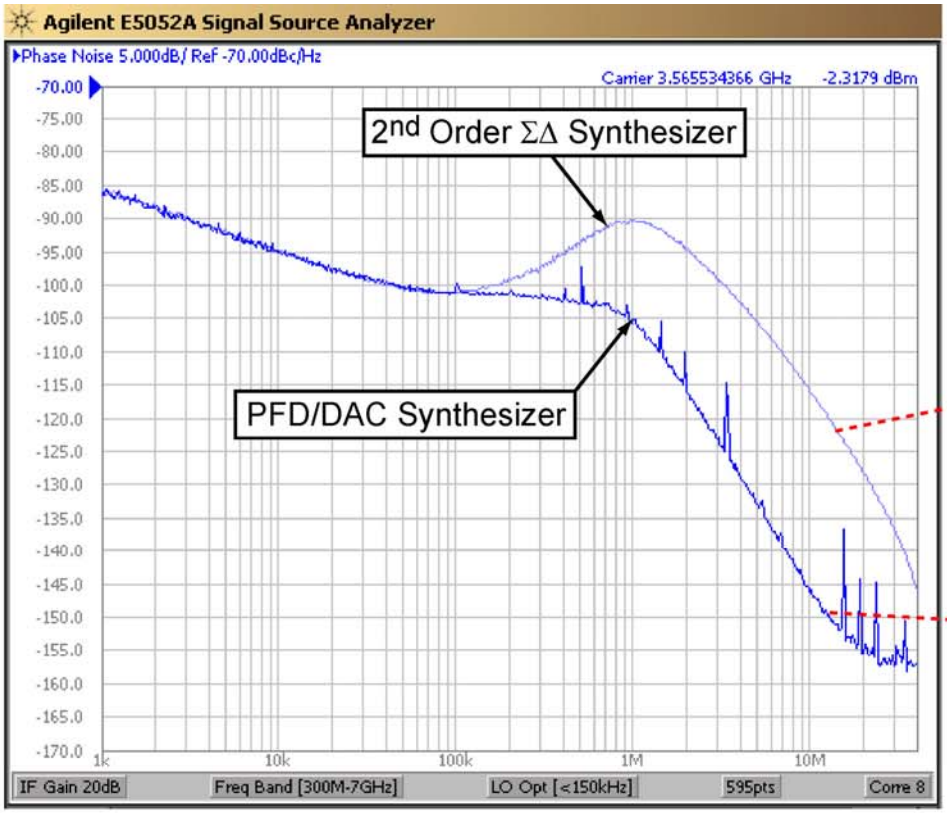
Measured Noise Suppression

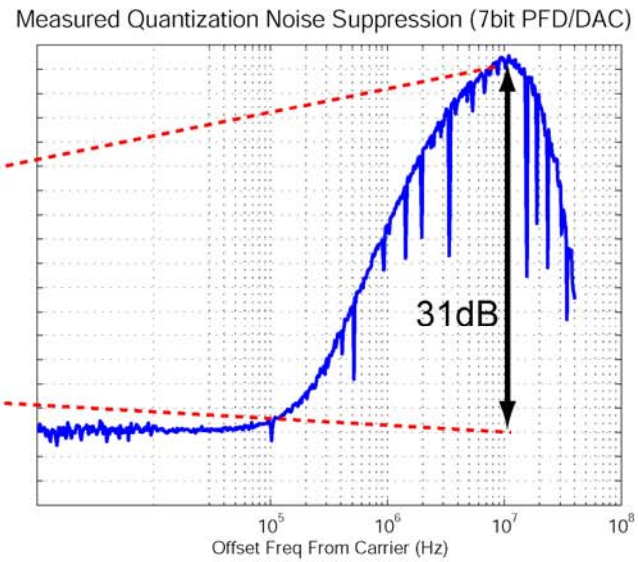




- Comparison of 7-bit PFD/DAC synthesizer with 2nd order ΣΔ Synthesizer
- Low freq noise ~2dB worse because of phase swapping
- 29dB quantization noise suppression measured at 10MHz !

Measured Noise Suppression: No Swapping





- Demonstrates that timing mismatch is degrading our maximum suppression by 2 dB (when swapping)
- Spurs occur due to gain error from timing mismatch

Summary of Calculation/Measured Results Comparison

- **Comparison of PLL Design Assistant results to measured data allow back extraction of key parameters:**
  - **Intrinsic noise**
    - Detector and VCO noise
  - **PFD/DAC nonidealities**
    - Delay mismatch value
- **Future work: better low frequency noise accuracy**

Conclusions

- **Fractional-N frequency synthesizers are about to undergo dramatic improvement in achieving high PLL bandwidth with excellent noise performance**
  - **The PFD/DAC approach presented here is only one of many possibilities to achieve this goal**
- **Design and simulation methodologies are starting to emerge**
  - **Analytical modeling of noise can be quite accurate**
    - The PLL Design Assistant can be useful in this area
  - **Behavioral simulation can be used to verify analytical models**
    - CppSim offers a convenient and fast framework for this

Research into High Bandwidth PLL Architectures is at an Exciting Crossroads