An Introduction to
_Digital_ Delta-Sigma Modulators

Michael Peter Kennedy FIEEE

Circuits and Systems Group

IEEE Circuits and Systems Society
Santa Clara Valley Chapter
28 July 2014
Dedication

- William F. Egan
“Although the number of commercially deployed DDSMs far exceeds that of analog ΔΣ modulators, most of the published ΔΣ modulator analyses apply only to analog ΔΣ modulators.

Interestingly, most of these analyses do not apply or even readily extend to the case of DDSMs”
• **The ideal DDSM**
  - The Big Idea
  - Signal processing assumptions
  - Applications of DDSMs

• **The real DDSM**
  - DDSM architectures
  - Origins of spurious tones
  - Minimization of spurious tones

• **Current research**
The ideal DDSM

- The Big Idea
- Signal processing assumptions
- Applications of DDSMs
The ideal DDSM

- A bandlimited digital signal $x (n \text{ bits})$ is requantized to a shorter word $y (m \text{ bits})$
- The additive quantization noise $e_q$ is highpass filtered for later removal by a lowpass filter
The ideal DDSM

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
The ideal DDSM

- $x$ is bandlimited
- $y$ includes highpass filtered quantization noise
- quantization noise can be removed by lowpass filtering
• $x$ is bandlimited

• $y$ includes highpass filtered quantization noise

• quantization noise can be removed by lowpass filtering
The ideal DDSM

- The oversampled output has almost the same S/N ratio as the input but much fewer bits (e.g. 16 reduced to 1)
- Fewer bits means it’s easier to implement the digital to analog conversion further along the signal processing path
- Oversampling makes it easier to do continuous-time reconstruction filtering
The ideal DDSM

- One typically assumes that the Classical Model of Quantization (CMQ) applies...
Classical model of quantization (CMQ)

- $e_q$ is statistically independent of $x$
- $e_q$ is uniformly distributed in $[-\Delta/2, + \Delta/2]$
- $e_q$ is stationary with a flat power spectrum
“CMQ can be applied when the quantizer input traverses several quantization levels between two successive samples”
The ideal DDSM

- The output depends on the signal $x$ and the quantization noise $e_q$
  $$Y(z) = STF(z) X(z) + NTF(z) E_q(z)$$

- The signal is typically scaled by the DDSM
  $$STF(z) = (1/M)$$

- The quantization noise is highpass filtered
  $$NTF(z) = (1-z^{-1})^l$$
• The output comprises the signal plus the filtered quantization noise

\[ Y(z) = \frac{1}{M}X(z) + E_q(z) (1-z^{-1})^l \]

x[n] spectrum

y[n] spectrum

Shaped white quantization noise
The ideal DDSM

- The noise scales as $f^{2l}$ at low frequencies

$$|E_q(z) (1-z^{-1})|^2 = \left(\frac{\Delta^2}{12}\right) 2^{2l} \sin^2(\pi f/f_s)$$

$$\approx \left(\frac{\Delta^2}{12}\right) (2\pi f/f_s)^{2l} \text{ when } f \ll f_s$$

CMQ approximation

Simulation +20l dB/decade
The ideal DDSM

- $x$ is bandlimited
- $y$ includes highpass filtered quantization noise
- Quantization noise can be removed by lowpass filtering
Applications of DDSMs

- Example 1: Class D audio amplifier
- Example 2: Oversampled DAC
- Example 3: Fractional-N frequency synthesizer
Applications of DDSMs

- Example 1: Class D audio amplifier
Class D digital audio power amplifier
Class D digital audio power amplifier

- Example: uniformly-sampled PWM; $N=16$
• Example 2: Oversampled DAC
Oversampled DAC

Figure 1.
By using oversampling and DDSM, the filter rolloff and DAC linearity specifications are relaxed.
Applications of DDSMs

- Example 3: Fractional-N frequency synthesizer
Fractional-N Frequency Synthesis

\[ f_{out} = \left( N_{int} + \frac{X}{2^n} \right) f_{ref} \]

- **Phase Detector (PFD)**
- **LPF**
- **VCO**
- **DDSM**

\[ f_{ref} \]
\[ f_d \]
\[ X \]
\[ n \text{ bits} \]
\[ 13 \text{ MHz} \]
\[ 1800 \text{ MHz} \]
\[ \div (N_{int} + y) \]
• DDSM sets the *average* division ratio

\[ f_{out} = (N_{int} + \frac{X}{2^n}) f_{ref} \]

• The frequency *resolution* is determined by the wordlength of the DDSM
INT, FRAC, MOD, AND R RELATIONSHIP

The INT, FRAC, and MOD values, in conjunction with the R counter, make it possible to generate output frequencies that are spaced by fractions of the phase frequency detector (PFD). See the RF Synthesizer: A Worked Example section for more information. The RF VCO frequency (RF<sub>OUT</sub>) equation is

\[ RF_{OUT} = F_{PFD} \times (INT + (FRAC/MOD)) \]  

(1)

where:
- \( RF_{OUT} \) is the output frequency of the external voltage controlled oscillator (VCO).
- \( INT \) is the preset divide ratio of the binary 9-bit counter (31 to 511).
- \( MOD \) is the preset fractional modulus (2 to 4095).
- \( FRAC \) is the numerator of the fractional division (0 to MOD – 1).

The PFD frequency is given by:

\[ F_{PFD} = REF_{IN} \times (1 + D)/R \]  

(2)

where:
- \( REF_{IN} \) is the reference input frequency.
- \( D \) is the \( REF_{IN} \) doubler bit.
- \( R \) is the preset divide ratio of the binary 4-bit programmable reference counter (1 to 15).
• **GSM example:**

\[ f_{\text{ref}} = 13 \text{ MHz}; f_{\text{out}} = 1.8 \text{ GHz} \]

\[ N \ (\text{INT }) = 138 \]

\[ X \ (\text{FRAC}) = 30 \]

\[ M \ (\text{MOD}) = 65 \]

\[ f_{\text{out}} = f_{\text{ref}} \ (\text{INT} + \text{FRAC/MOD}) \]

\[ = 13 \ (138 + 30/65) \text{ MHz} \]

\[ = 1794 + 6 \text{ MHz} \]

\[ = 1800 \text{ MHz} \]
Fractional-N Frequency Synthesis

\[ N + \frac{F}{M} = \frac{f_{VCO}}{f_{PFD}} \]
Fractional-N Frequency Synthesis

- Output phase noise

\[ N_{\text{int}} + y \]
Fractional-N Frequency Synthesis

Noise

\[ \Phi_{jfi}[k] \xrightarrow{\alpha \cdot \frac{T}{2\pi}} \]

\[ e_{spur}(t) \]

\[ I_{cpn}(t) \xrightarrow{\frac{1}{I}} \]

\[ \Phi_{npfd}(t) \]

PFD-referred Noise

\[ S_{E_n}(f) \]

\[ \Phi_{vn}(t) \]

VCO-referred Noise

\[ \Phi_{wn}(t) \]

\[ -20 \text{ dB/dec} \]

Divide value variation

\[ n[k] \xrightarrow{\frac{2\pi}{1 - z^{-1}}} \]

\[ \Phi_d(t) \]

\[ T \cdot G(f) \]

\[ \Phi_c(t) \]

\[ \Phi_{out}(t) \]

Alternate Representation

\[ G(f) \]

\[ n[k] \xrightarrow{\Phi_c(t)} \]

\[ \frac{1}{\text{freq filter}} \]

D/A and Filter

Freq. \rightarrow Phase

\[ \Phi_c(t) \]
• DDSM contributes little in-band noise; loop filter attenuates out-of-band components.

\[ f_0 = 84 \text{ kHz} \]

MASH 1-1 with 2\textsuperscript{nd} order loop
- Quantization noise is first highpass filtered by the DDSM’s NTF
- Shaped quantization noise is then attenuated by the (lowpass CT) loop filter
- We get high frequency resolution for free!
The ideal DDSM: Summary

- **The Big Idea**
  
  *Linear; allpass STF, NTF attenuates quantization noise in the signal band*

- **Signal processing assumptions**
  
  *CMQ applies, quantization noise is uncorrelated with the signal*

- **Applications of DDSMs**
  
  *Digital Power Amplifiers, DACs, Fractional-N Frequency Synthesizers*
The real DDSM

- DDSM architectures
- Origins of spurious tones (spurs)
- Minimization of spurious tones
The real DDSM

- The output depends on the signal $x$ and the quantization noise $e_q$

$$Y(z) = STF(z) X(z) + NTF(z) E_q(z)$$

where $STF(z) = 1/M$ and $NTF(z) = (1-z^{-1})^l$

- Consider the simplest case $l=1$

$$Y(z) = (1/M)X(z) + (1-z^{-1}) E_q(z)$$
The real DDSM

- A 1st order DDSM can be implemented using a *digital accumulator*
• 1st order Error Feedback Modulator (EFM1)
• $x$ is the input; $y$ (the carry out) is the output; $s$ (the value stored in the register) is the state; $M$ is the modulus
• 1st order Error Feedback Modulator (EFM1)

\[ s[n+1] = (x[n] + s[n]) \mod M \]

\[ y[n] = Q(x[n] + s[n]) \]
The real DDSM

- 1st order Error Feedback Modulator (EFM1)

\[ Q(v[n]) = \left(\frac{1}{M}\right)v[n] + e_q[n] \]

The real DDSM

- **1st order Error Feedback Modulator (EFM1)**

\[
y[n] = Q(x[n] + s[n]) = Q(x[n] + e[n-1]) = (1/M)(x[n] + e[n-1]) + e_q[n] = (1/M)(x[n] - Me_q[n-1]) + e_q[n] = (1/M)(x[n]) + (e_q[n] - e_q[n-1])
\]
The real DDSM

- **1st order Error Feedback Modulator (EFM1)**

\[ Y(z) = \frac{1}{M}X(z) + E_{q1}(z) (1-z^{-1}) \]
The real DDSM

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
• Higher-order noise-shaping architectures
  
  Single Quantizer (SQ-DDSM)
  Multi-Stage Noise Shaping (MASH)
• MASH: Higher order filtering with *exact* cancellation of intermediate errors

\[ Y(z) = \left( \frac{1}{M} \right) X(z) + E_q l(z) (1-z^{-1})^l \]
The real DDSM

- **MASH**: Idealized power spectra (white noise with 2\textsuperscript{nd} and 3\textsuperscript{rd} order filters)

![Graph showing idealized power spectra with +40 dB/decade and +60 dB/decade slopes](image)
The real DDSM

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
The real DDSM

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
A MASH 1-1-1 with a constant input and an even initial condition can produce **spurious tones** (spurs)

Wordlength: 18 bits

Spurs

+60 dB/decade
The real DDSM

- In audio DACs, these are called *idle tones*
• Phase noise in fractional-N frequency synthesizer with *odd* initial condition in MASH 1-1-1
• Phase noise in fractional-N frequency synthesizer with constant input and *even* initial condition in MASH 1-1-1-1
What can go wrong and why?

- Different *wordlengths* can produce different spectra
- Different *inputs* can produce different spectra
- Different *initial conditions* can produce different spectra
- *All of the above* can cause spurs!
What can go wrong and why?

- Different *wordlengths* produce different spectra

![Graph showing different wordlengths](image)

MASH 1-1-1

green plot: 9 bits

magenta plot: 18 bits
What can go wrong and why?

- Different inputs produce different spectra

MASH 1-1-1

wordlength: 17 bits

(i) $X=1$

(ii) $X=2^{16}$
What can go wrong and why?

- Different *initial conditions* produce different spectra

MASH 1-1-1

Input = 8 (decimal)

accumulator word length: 14 bits

cycle length for green plot: 4096

cycle length for blue plot: 32768

red plot: CMQ approximation
What can go wrong and why?

- Fundamental problem: *Short cycles* produce strong tones
“CMQ can be applied when the quantizer input traverses several quantization levels between two successive samples”

• When the input is constant or slowly moving, CMQ does not apply...
What can go wrong and why?

- The DDSM is a Finite State Machine (FSM)
- The FSM has a finite state space $S$ (containing $N_s$ states) and a deterministic rule $G_D$ (called the dynamic) that governs the evolution of states
- The next state is determined completely by the current state and the input:

$$s[n+1] = G_D(s[n], x[n])$$
What can go wrong and why?

• If the input is fixed, the most complex trajectory visits each state in the state space once before repeating; the longest cycle has period \( N = N_s - 1 \)

• In the worst case, the trajectory in a MASH 1-1-1-1 repeats with period \( N = 4 \)
What can go wrong and why?

- Example worst case period: $N = 4$

- $f_s/4$

- MASH 1-1-1
  - wordlength: 17 bits

  (i) $X=1$
  (ii) $X=2^{16}$
• Parseval’s theorem

\[
\frac{1}{N} \sum_{n=0,1,\ldots,N-1} |x[n]|^2 = \sum_{k=0,1,\ldots,N-1} |X[k]|^2
\]

power in time domain = power in frequency domain

• \(X[k]\) are the Discrete Time Fourier Series (DTFS) coefficients
• Quantization noise power is distributed over $N$ tones
• Fewer tones results in greater power per tone
How to fix the problem?

- **Problem**: When the cycle length is short, the quantization noise is distributed over a small number of tones, resulting in high average quantization noise power per tone.

- **Solution**: *Maximize the cycle length* to distribute the quantization noise power over more tones, thereby reducing the average noise per tone.
How to fix the problem?

• **Stochastic approach**
  
  Add dither

• **Deterministic approaches**
  
  Set initial states
  
  Restrict inputs
  
  Change the architecture
Stochastic approach

- Make the dynamic *stochastic*
- The next state depends on the current state $s$, the input $x$, and a *random* dither signal $d$

$$s[n+1] = G_s(s[n], x[n], d[n])$$

- Periodicity is destroyed
- Trajectories can be much longer than $N_s$
• Additive “LSB” dither: \( V(z) = (1-z^{-1})^R \)
• Dither spreads the power over more tones

9 bit without dither

9 bit with LSB dither
Fractional-N frequency synthesizer

Figure 5. Single-Sideband Phase Noise Plot (Lowest Noise Mode)
Fractional-N frequency synthesizer

Figure 7. Single-Sideband Phase Noise Plot (Low Spur Mode)
Fractional-N frequency synthesizer

2679.4MHz FRACTIONAL-N PHASE NOISE vs. FREQUENCY (LOW-NOISE MODE)

2679.4MHz FRACTIONAL-N PHASE NOISE vs. FREQUENCY (LOW-SPUR MODE)
Stochastic approach: LSB dither

- LSB dither is indistinguishable from $x...$

$$Y(z) = X(z) + E_{ql}(z)(1-z^{-1})^l + D(z) (1-z^{-1})^R$$
• **MASH 1-1-1 with LSB dither:**  
  \[ V(z) = (1-z^{-1})^R \]

- **Green:**  
  \[ V(z) = 0 \]

- **Red:**  
  \[ V(z) = 1 \]

- **Blue:**  
  \[ V(z) = 1-z^{-1} \]
**Stochastic approach: LSB dither**

- **Maximum filter order for LSB dither [PG07]**

<table>
<thead>
<tr>
<th>Type of MASH</th>
<th>Filter</th>
<th>( V(z) = (1-z^{-1})^R ) allowed ( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1 lowpass</td>
<td>( z^{-2}(1-z^{-1})^2 )</td>
<td>0</td>
</tr>
<tr>
<td>1-1-1, 1-2, 2-1 lowpass</td>
<td>( z^{-3}(1-z^{-1})^3 )</td>
<td>1</td>
</tr>
<tr>
<td>( m_1-m_2-\ldots-m_k, \sum m_j = l )</td>
<td>( z^{-l}(1-z^{-1})^l )</td>
<td>( l-2 )</td>
</tr>
</tbody>
</table>
Deterministic approaches: ICs

- **MASH 1-1-1**: Choose $s_1[0]$ odd ("seeding")

  **MASH 1-1-1**

  Input = 8 (decimal),
  
  accumulator word length: 14 bits
  
  cycle length for green plot: 4096
  
  cycle length for blue plot: 32768
Deterministic approaches: ICs

- MASH cycle lengths with $s_1[0]$ odd ($M=2^n$)

<table>
<thead>
<tr>
<th>Modulator order $l$</th>
<th>Guaranteed minimum $N$</th>
<th>Maximum $N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$M/2$</td>
<td>$2M$</td>
</tr>
<tr>
<td>3</td>
<td>$2M$</td>
<td>$2M$</td>
</tr>
<tr>
<td>4</td>
<td>$2M$</td>
<td>$4M$</td>
</tr>
<tr>
<td>5</td>
<td>$4M$</td>
<td>$4M$</td>
</tr>
</tbody>
</table>
**HK-MASH: for maximum length cycles**

![Block diagram for HK-MASH](image)
### Guaranteed minimum cycle lengths

<table>
<thead>
<tr>
<th>Order ($l$)</th>
<th>Conventional MASH with odd initial condition</th>
<th>HK-MASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$M/2$</td>
<td>$\approx M^2$</td>
</tr>
<tr>
<td>3</td>
<td>$2M$</td>
<td>$\approx M^3$</td>
</tr>
<tr>
<td>(M=512)</td>
<td>(1024)</td>
<td>(509$^3 \approx 132 \times 10^6$)</td>
</tr>
<tr>
<td>4</td>
<td>$2M$</td>
<td>$\approx M^4$</td>
</tr>
<tr>
<td>5</td>
<td>$4M$</td>
<td>$\approx M^5$</td>
</tr>
</tbody>
</table>
Deterministic approaches: Architecture

- HK-MASH vs seeded and dithered MASH
The real DDSM: Summary

• DDSM architectures
  Nonlinear, CMQ does not always apply

• Origins of spurious tones
  Constant and/or period inputs produce periodic quantization noise
  Shorter cycles yield higher power per tone
  Cycle lengths depend on the input, initial conditions, and word length

• Minimization of spurious tones
  Stochastic: add filtered dither
  Deterministic: choose odd input or initial condition, or modify architecture
• **Error masking**
  Deliberately introduce errors and then mask their effects spectrally

• **Bus-splitting**
  Save area and power by nesting DDSMs

• **Mixed-radix divider controller**
  Set frequency precisely by using mixed-radix fractions
• **Dither**
  
  Determine the effects of periodic dither

• **Nonlinearity**
  
  Spurs regrow when the “noiselike” output of the DDSM encounters a nonlinearity

• **Successive requantizers**
  
  Sequences which are less likely to produce tones when they encounter nonlinearities
The output of a DDSM contains two terms:

\[ Y(z) = STF(z)X(z) + NTF(z)E_q(z) \]

The second (shaped noise) term is removed by filtering further along the signal processing chain.

Any additional quantization noise that lies below the mask of the shaped noise will also be removed by that filter!
Error Masking

• The signal itself is also quantized, providing a noise floor below which other errors can be masked.
Error Masking

![Graph showing error masking with normalized frequency on the x-axis and power/frequency bin in dB/sample on the y-axis.]

- $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$
- $f_0$, $L_{nf_0}$

Normalized Frequency ($x\pi$ rad/sample) vs. Power/frequency bin (dB/sample)
• Exploit error masking to reduce the bus width, thereby reducing area and power and increasing speed

• We consider two ideas:
  MASH DDSMs with interstage quantization
  Bus-splitting
Exact 3$^{rd}$ order MASH architecture

All modulators have the same wordlength $N$. 
• All stages have the *same* wordlength $N$

• The hardware requirement scales as $3N$

• This is overkill...
Later stages have smaller wordlengths: $L < M < N$. 

Reduced complexity MASH DDSM
• Successive stages have decreasing wordlengths

• The hardware requirement scales less than $3N$

• This is optimal...
Reduced complexity 3rd order MASH

- 3rd order MASH with interstage errors

\[
Y_1(z) = (1/2^N)X(z) + E_{q1}(z) \left( 1-z^{-1} \right)
\]

\[
Y_2(z) = (1/2^M)(-2^M E_{q1}(z) + E_{q12}(z)) + E_{q2}(z) \left( 1-z^{-1} \right)
\]

\[
Y_3(z) = (1/2^L)(-2^L E_{q2}(z) + E_{q23}(z)) + E_{q3}(z) \left( 1-z^{-1} \right)
\]

\[
Y(z) = Y_1(z) + Y_2(z) \left( 1-z^{-1} \right) + Y_3(z) \left( 1-z^{-1} \right)^2
\]

\[
= (1/2^N)X(z) + E_{q3}(z) \left( 1-z^{-1} \right)^3
\]

\[
+ (1/2^M)E_{q12}(z) \left( 1-z^{-1} \right) + (1/2^L)E_{q23}(z) \left( 1-z^{-1} \right)^2
\]
Error Masking

Graph showing the relationship between power/frequency bin (dB/sample) and normalized frequency ($\times \pi$ rad/sample) with markers $L_0$, $L_1$, $L_2$, and $L_3$.
Reduced complexity 3\textsuperscript{rd} order MASH

- Relative Hardware Consumption (RHC): 

\[ RHC \approx \frac{N + \text{ceil}(2N/3) + \text{ceil}(N/3)}{3N} . \]

- For large \( N \), \( RHC \) approaches 67\%, representing a 33\% saving in area relative to a MASH DDSM with identical EFM1 stages.
• We can split the bus multiple times...
• Example: *Nested* bus-splitting 1-2-3 DDSM3
Bus-splitting accelerator

Bus-splitting

\[ Y(z) = \left( \frac{1}{2^N} \right) X(z) + \left( \frac{1}{2^{(N_{MSB}+N_{ISB})}} \right) E_{q1}(z) \left( 1-z^{-1} \right) \]
\[ + \left( \frac{1}{2^{N_{MSB}}} \right) E_{q2}(z) \left( 1-z^{-1} \right)^2 + E_{q3}(z) \left( 1-z^{-1} \right)^3 \]
Comparison

**DDSM3**

\[ ENOB = 19.54 \]
\[ area, power = 1, 1 \]

**1-2-3 DDSM3**

\[ ENOB = 19.22 \]
\[ area, power = 0.62, 0.51 \]
Frequency synthesis: State of the art

- Conventional frac-N frequency synthesizer

\[ f_{VCO} = \left( N_0 + \frac{N_1}{M_1} \right) f_{PD} \]

Where:
- \( f_{PD} \) is the phase detector output frequency
- \( f_d \) is the divide frequency
- \( f_{VCO} \) is the voltage-controlled oscillator output frequency
- \( 122.88 \text{ MHz} \) is the divide frequency
- \( 3999.999... \text{ MHz} \) is the output frequency
- \( N_0 \) is the integer part
- \( N_1 \) is the fractional part
- \( M_1 \) is the modulus

Diagram:
- Phase Detector (PFD)
- Low-Pass Filter (LPF)
- Voltage-Controlled Oscillator (VCO)
- DDSM (Direct Digital Synthesizer Module)
- \( 2^n \) DDSM

Symbols:
- \( f_{PD} \)
- \( f_d \)
- \( y \)
Frequency synthesis

- **Mixed-radix frac-N frequency synthesizer**

\[ f_{VCO} = \left( N_0 + \frac{N_1 + \frac{N_2}{M_2}}{M_1} \right) f_{PD} \]

[References: Kennedy et al., *IEEE J. Solid-State Circuits*, 2014.]
Frequency synthesis

- Conventional

\[ f_{VCO} = \left( 32 + \frac{9262421}{2^{24}} \right) 122.88 \text{ MHz} = 3.9999999756 \ldots \text{ GHz} \]

- Mixed-radix

\[ f_{VCO} = \left( 32 + \frac{9262421 + \frac{1}{3}}{2^{24}} \right) 122.88 \text{ MHz} = 4.000000000 \text{ GHz} \]
Frequency synthesis
Frequency synthesis

\[
Y_0(z) = \left( N_0 + \frac{N_1 + \frac{N_2}{M_2}}{M_1} \right) + \text{NTF}_1(z)E_{q_1}(z) + \frac{1}{M_1} \text{NTF}_2(z)E_{q_2}(z)
\]
Frequency synthesis

• Dither second and third stages...
Stochastic approach: LSB dither

- “Efficient dithering” [GDGAFVM10]

[Graph showing power spectral density (PSD) vs. frequency with different dithering methods indicated]
Fractional-N frequency synthesizer

![Graph](image_url)
Fractional-N frequency synthesizer

![Graph showing normalized frequency vs. power/frequency (dB/rad/sample)](image)

- X: 0.007843
- Y: -42.94

- X: 0.0002451
- Y: -189.3
Simplified model

Requantizer
Simplified model

- Diagram showing a simplified model of a phase-locked loop (PLL).
  - f_{ref} input to PFD
  - f_{div} input to Multi Modulus Divider
  - x[n] input to Divider Controller
  - y[n] output from Divider Controller
  - f_{out} output from VCO
  - Charge Pump
  - Loop Filter

- Graphs showing sample number vs. power/frequency and frequency vs. power/frequency.
Simplified model

Simplified model
Simplified model

\[ x[n] \xrightarrow{\text{Requantizer}} y[n] + s[n] - t[n] + g[n] \]

\[ \frac{1}{M} \]

\[ Z^{-1} \]

\( G(s) \)

\( G(f) \)
Spurs due to the PFD-CP nonlinearity

Diagram showing the process of Spurs due to the PFD-CP nonlinearity, including the blocks for Requantizer, Memoryless Nonlinearity, and the transfer functions G(s) and G(f).
Acknowledgements

• FIRB Italian National Research Program RBAP06L4S5
• Irish Research Council for Science, Engineering & Technology (IRCSET)
• Science Foundation Ireland Grants 02/IN1/I45 and 08/IN1/I1854

peter.kennedy@ucc.ie
References


References


References


References


References


References


References
