EE 240C
Analog-Digital Interface Integrated Circuits

5th Order Modulator (Example)

Overview

- Building and evaluating behavioral models
  - Focus on functionality first
  - Add nonidealities later
  - Beware: changes get more expensive later in the process …

- A 5th-order, 1-Bit ΣΔ modulator example
  - Noise shaping
  - Complex loop filters
  - Stability
  - Voltage scaling

SD Modulator Filter Design

- Procedure
  - Establish requirements
  - Design noise-transfer function, NTF
  - Determine loop-filter, H
  - Synthesize filter
  - Evaluate performance, stability

- References:
  - R. W. Adams and R. Schreier, "Stability Theory for DS Modulators," in Delta-Sigma Data Converters, S. Norsworthy et al. (eds), IEEE Press, 1997, pp. 141-164.
  - S. Pavan, R. Schreier, and G. C. Temes, Understanding Delta-Sigma Data Converters. Wiley-IEEE Press, 2017. Chapter 4.

Modulator Specification

- Example: Audio ADC
  - Dynamic range — DR — 16 Bits
  - Signal bandwidth — B — 20 kHz
  - Nyquist frequency — f_N — 44.1 kHz
  - Modulator order — L — 5
  - Oversampling ratio — $M = f_s / f_N$ — 64
  - Sampling frequency — f_s — 2.822 MHz

- The order L and oversampling ratio M are chosen based on
  - SQNR > 120dB (20dB below thermal noise)
  - Experience (e.g. Figure 4.14 in Adams & Schreier or Figure 4.18, 4.19, 4.20 in Understand Delta-Sigma Data Converters)

Modulator Specification

- SQNR
  - Modulator Order (N in graphs below)
  - Oversampling Ratio (OSR)
  - Number of levels / bits in quantizer

[S. Pavan, R. Schreier, and G. C. Temes, *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, 2017. Chapter 4.]

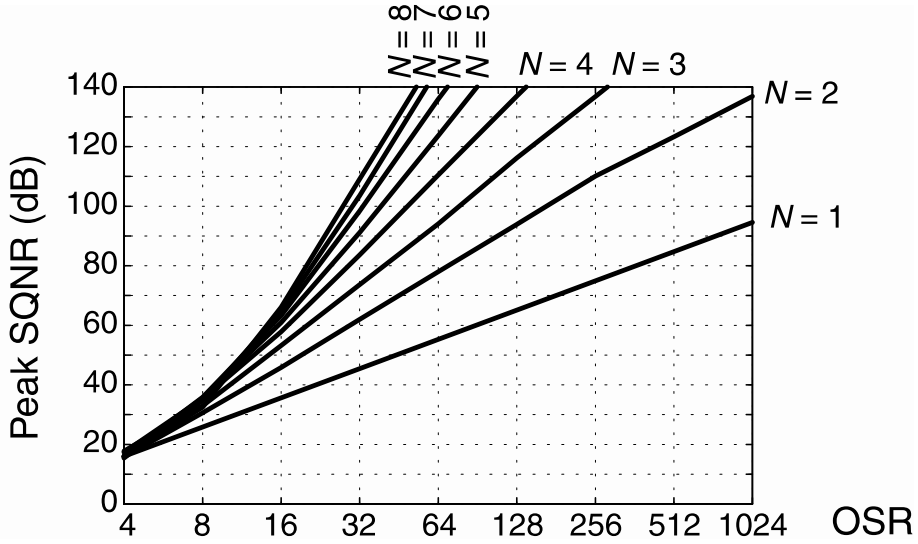


Figure 4.18 Empirical SQNR limit for 1-bit modulators of order N.

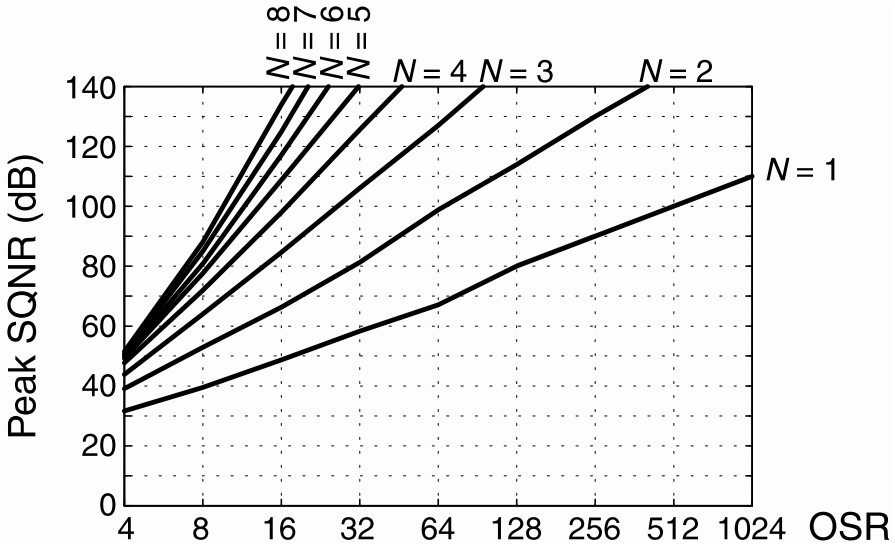
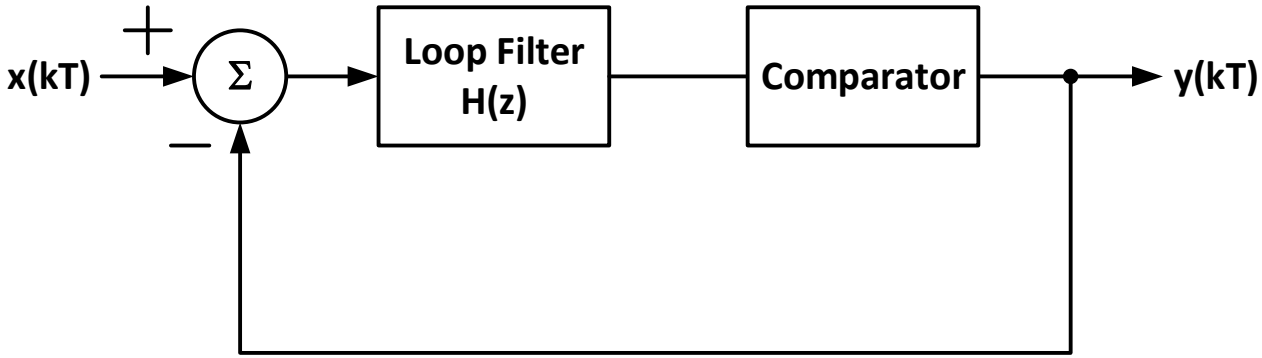


Figure 4.20 Empirical SQNR limit for modulators with 3-bit quantizers of order N.

Modulator Block Diagram



$$\text{STF} = \frac{Y(z)}{X(z)} = \frac{H(z)}{1 + H(z)}$$

$$\text{NTF} = \frac{Y(z)}{E(z)} = \frac{1}{1 + H(z)}$$

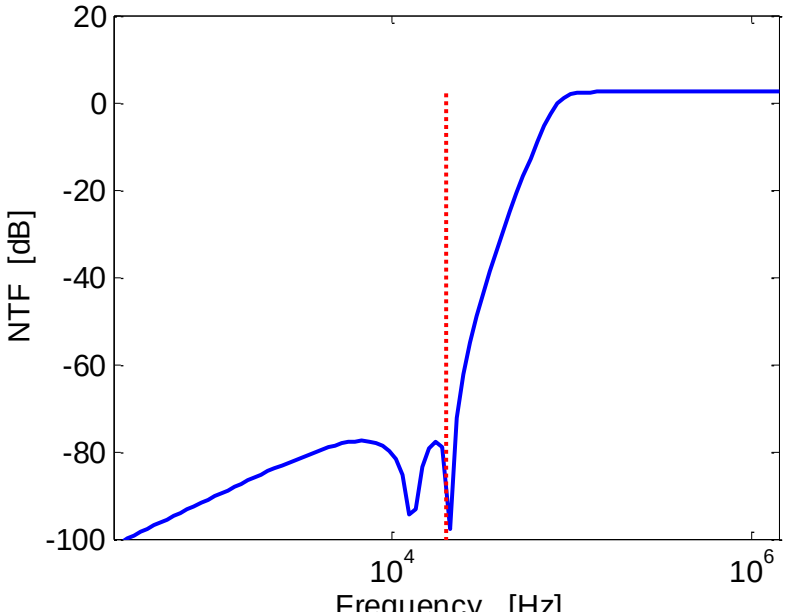
Approach:
Design NTF and solve for H(z)

Noise Transfer Function, NTF(z)

```
% stop-band attenuation Rstop ...
% reduce if design is not stable

Rstop = 80;
[b,a] = cheby2(L, Rstop, 1/M, 'high');

% normalize (for causality)
b = b/b(1);
NTF = filt(b, a, 1/fs);

% check stability (mag < 1.5)
[mag] = bode(NTF, pi*fs)
```

>> mag = 1.32

sigma_delta_L5_design.m



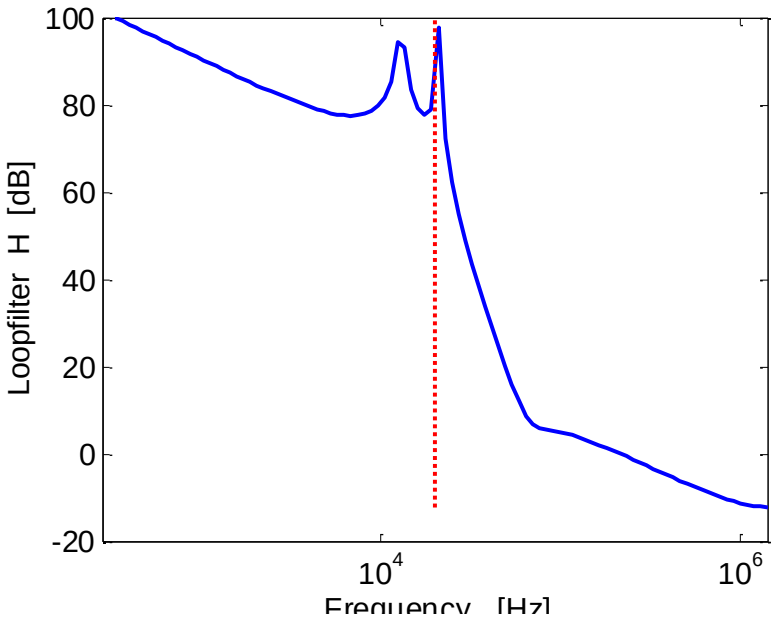
Noise Transfer Function, NTF(z)

- NTF(z) numerator and denominator constant terms equal to 1
  (b(1) = 1 and a(1) = 1)
  → H(z) = 1/NTF(z) – 1 has no constant term in numerator
  → unit delay through loop filter
  → realizable modulator
- Rule-of-thumb for stability of 1-bit modulators (Lee's rule):
  $\|NTF(\omega)\|_\infty = \max |NTF(\omega)| < 1.5$
- In-band NTF attenuation (→ higher SQNR)
  vs out-of-band NTF gain (→ lower maximum stable amplitude)

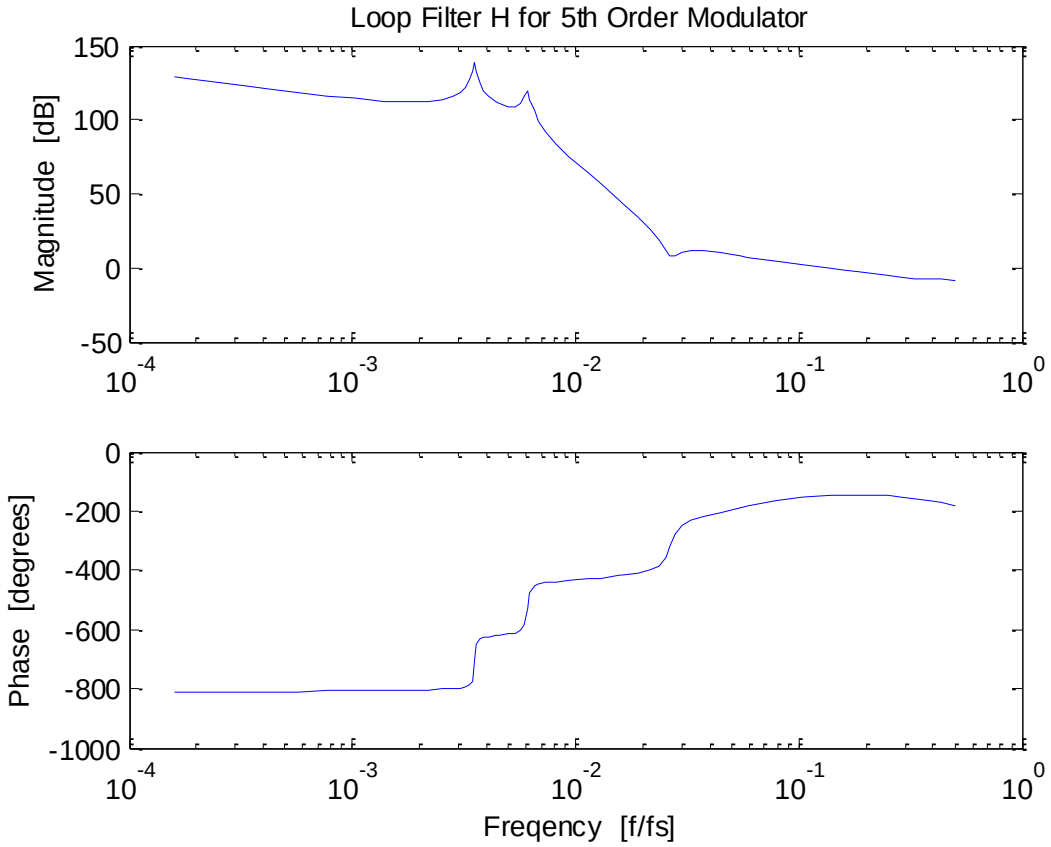
Loop-Filter, H(z)

```
H = inv(NTF) - filt(1, 1, 1/fs);

% check causality … y(1) should be 0
y = impulse(H);
y = y(1)
```

```
>> y =  0
```



5th Order Loop Filter



sigma_delta_L5_H.m

- Lot's of gain in the pass-band
- Remember that NTF ~ 1/H
- H ~ 0dB in stop-band gives quantization noise a place to show up

Modulator Topologies

- CIFB: Cascade of Integrators with Feedback
  - State at the output of the integrators
    → larger unscaled signals at output of integrators
  - Multiple DACs

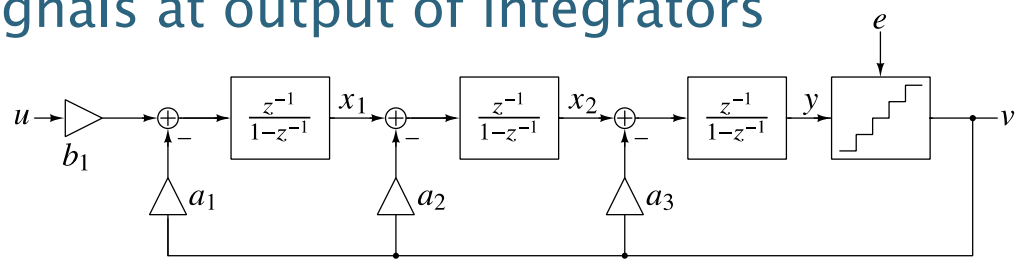
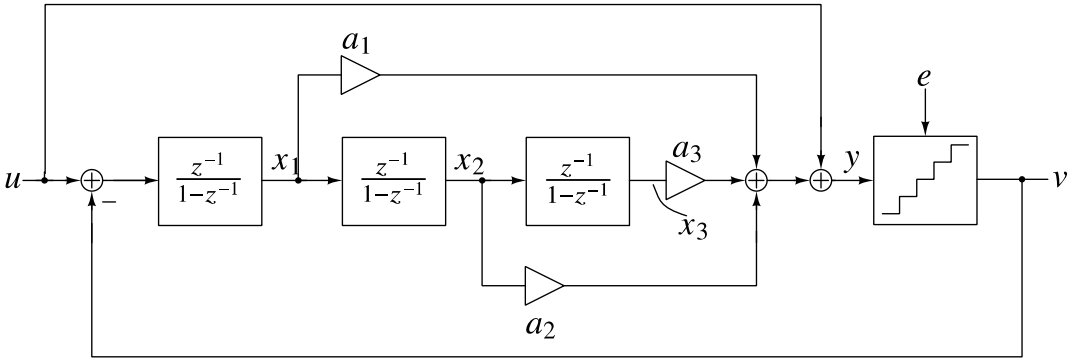


Figure 4.21 A third-order NTF realized as a cascade of integrators with feedback (CIFB) structure. All NTF zeros are at $z = 1$.

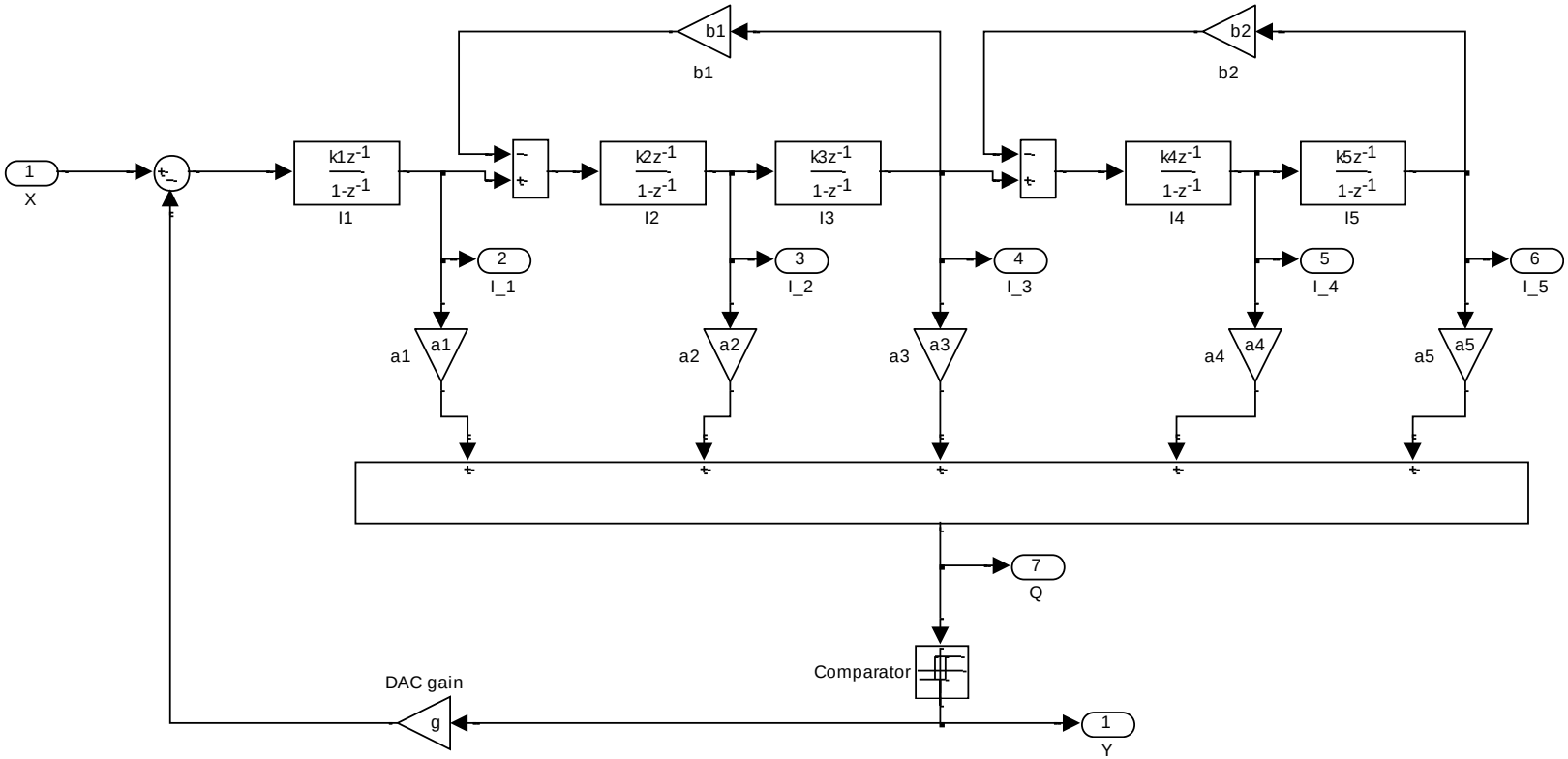
- CIFF: Cascade of Integrators with Feed-forward
  - Only quantization noise in integrators



[S. Pavan, R. Schreier, and G. C. Temes, *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, 2017. Chapter 4.]

Figure 4.29 A low distortion CIFF structure, accomplished using input feedforward.

Modulator Topology



sigma_delta_L5_sim.mdl

Rounded Filter Coefficients

```
a1=1;
a2=1/2;
a3=1/4;
a4=1/8;
a5=1/8;
```

```
k1=1;
k2=1;
k3=1/2;
k4=1/4;
k5=1/8;
```

```
b1=1/1024;
b2=1/16-1/64;
```

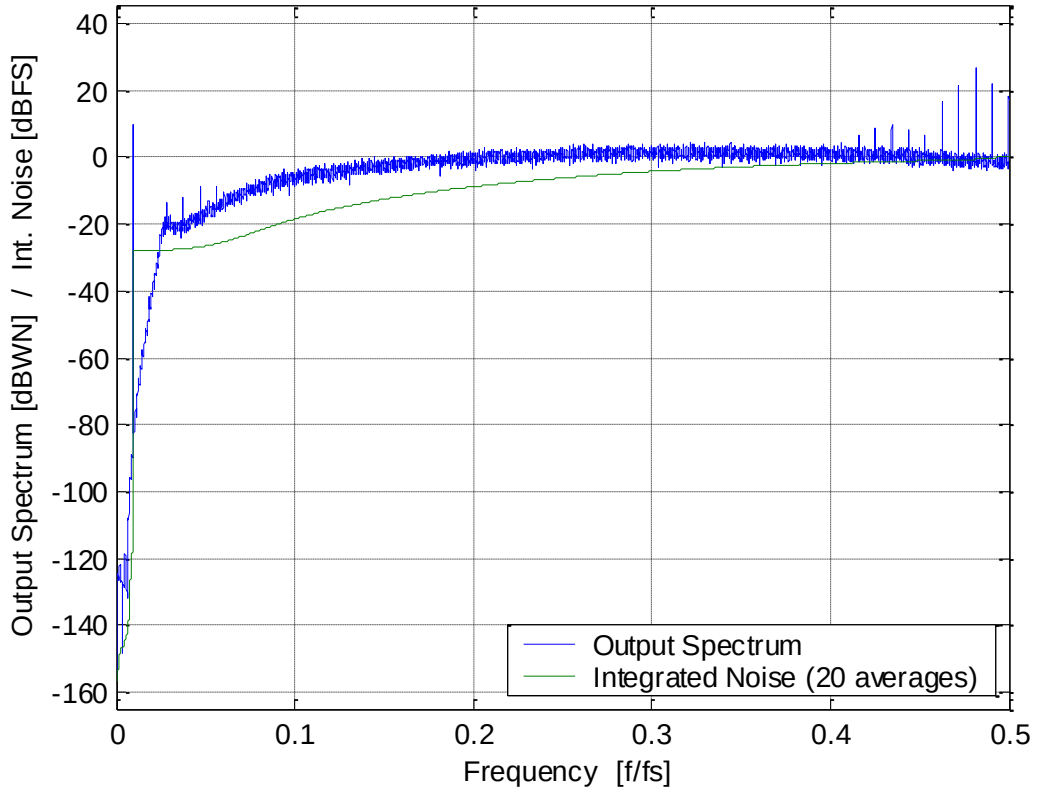
```
g =1;
```

Ref: Nav Sooch, Don Kerth, Eric Swanson, and Tetsuro Sugimoto, “Phase Equalization System for a Digital-to-Analog Converter Using Separate Digital and Analog Sections”, U.S. Patent 5061925, 1990, figure 3 and table 1.

EE 240C
Analog-Digital Interface Integrated Circuits

Noise Shaping

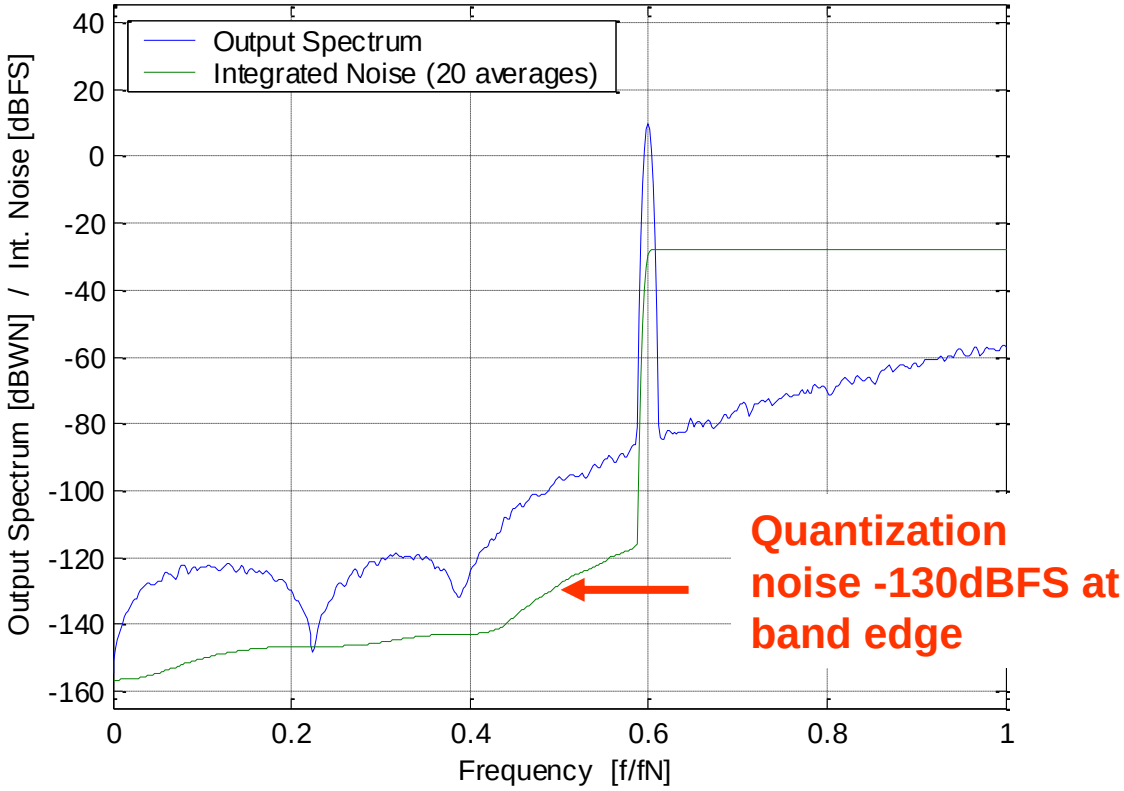
5th Order Noise Shaping



sigma_delta_L5.m

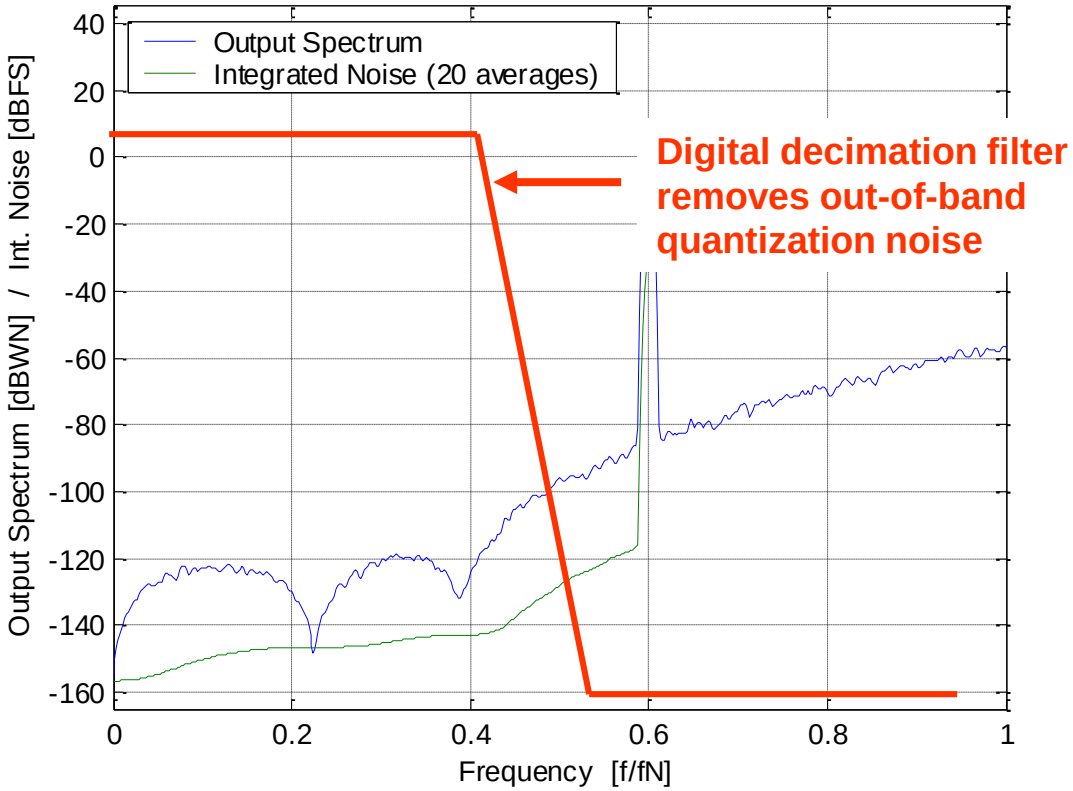
- Mostly quantization noise, except at low frequencies
- Let's zoom …

5th Order Noise Shaping



sigma_delta_L5.m

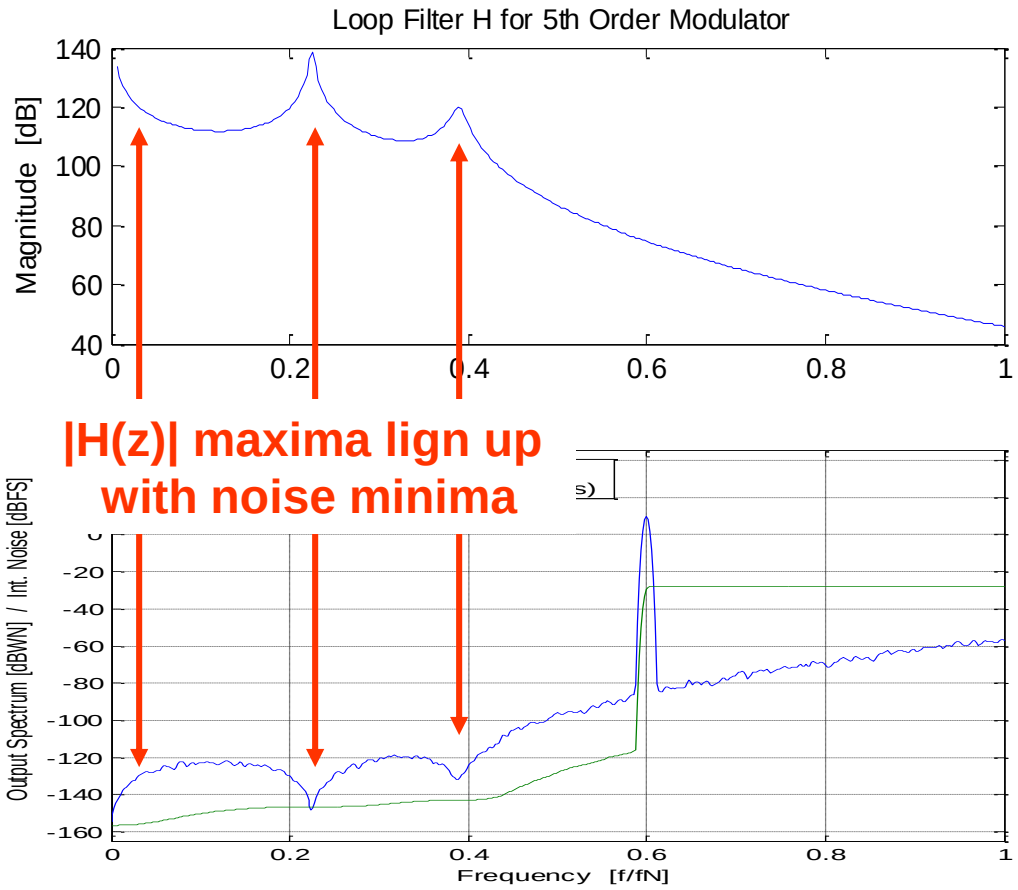
5th Order Noise Shaping



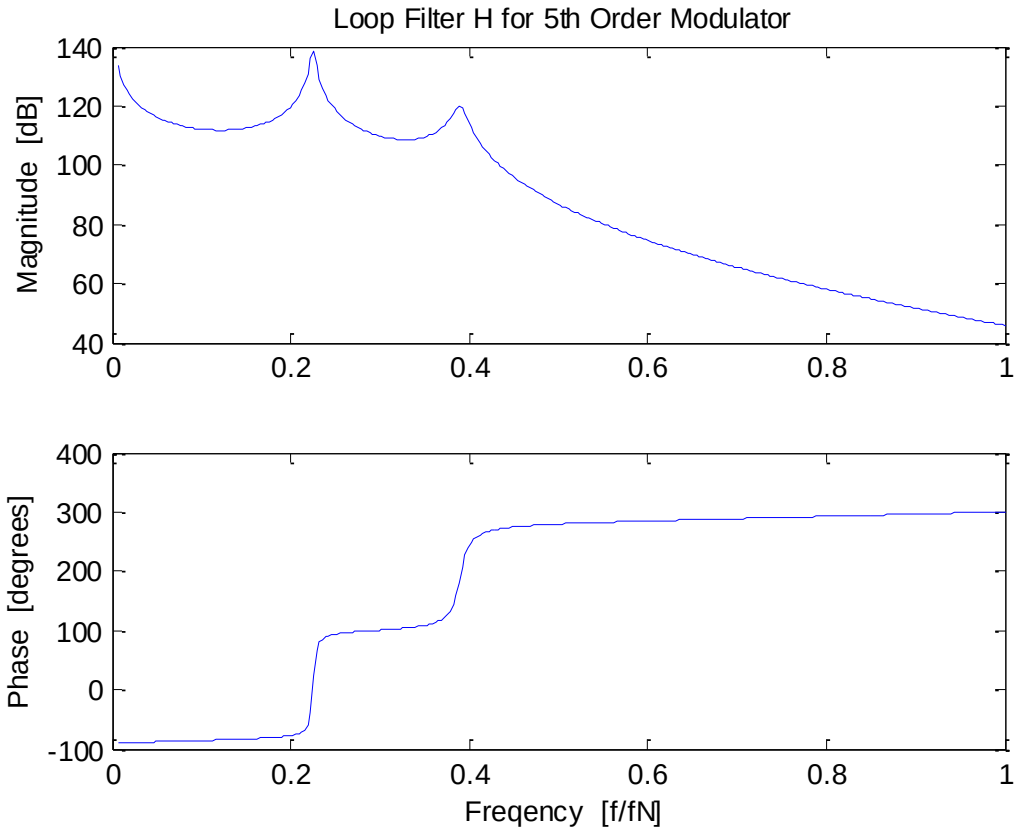
sigma_delta_L5.m

- SQNR > 120dB
- Sigma-delta modulators are usually designed for negligible quantization noise
- Other error sources dominate, e.g. thermal noise

In-Band Noise Shaping



In-Band Noise Shaping

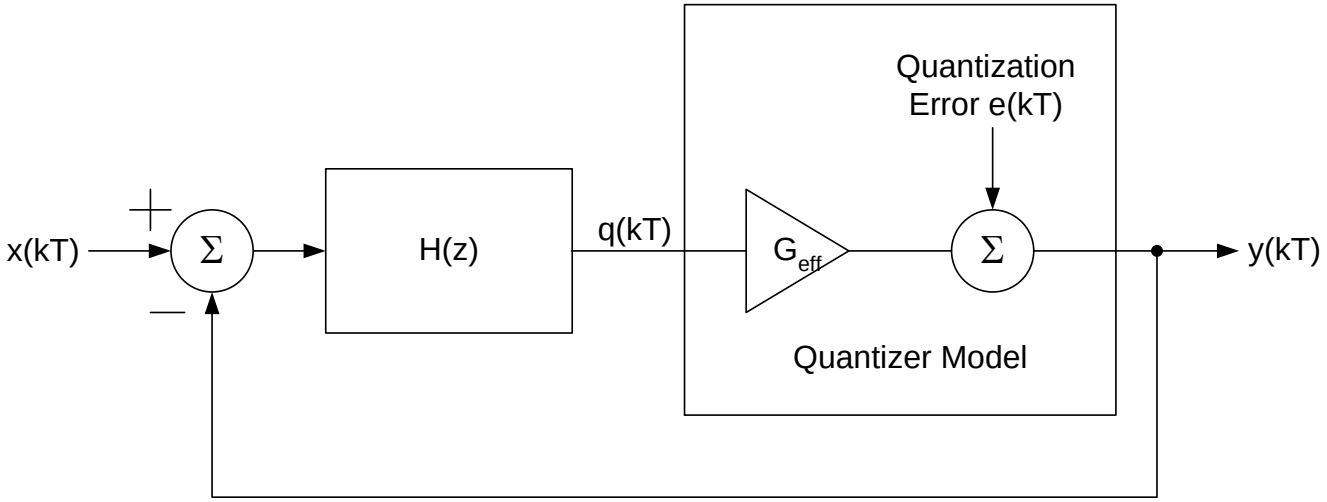


- Positive phase jumps indicates poles of H(z) slightly outside unit circle
- Is the modulator stable?
- Let's analyze …

EE 240C
Analog-Digital Interface Integrated Circuits

Stability and Voltage Scaling

Stability Analysis

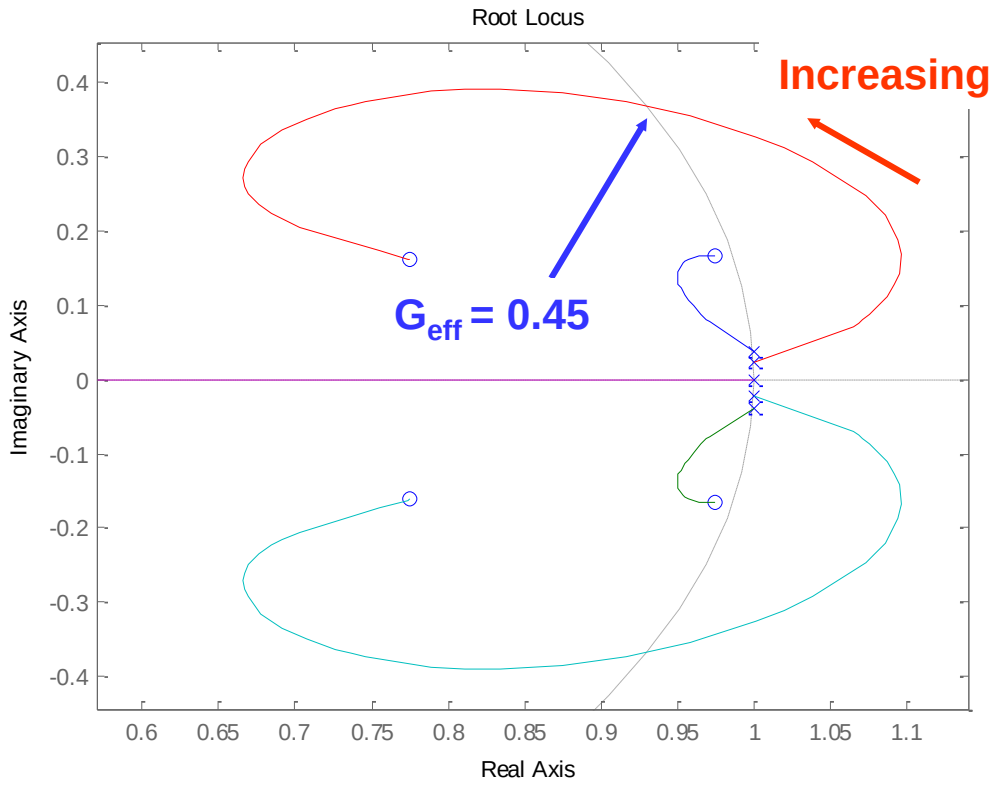


- Approach: linearize quantizer and use linear system theory!
- Effective quantizer gain

$$G_{eff}^2 = \overline{y^2} \Big/ \overline{q^2}$$

- Obtain G_{eff} from simulation

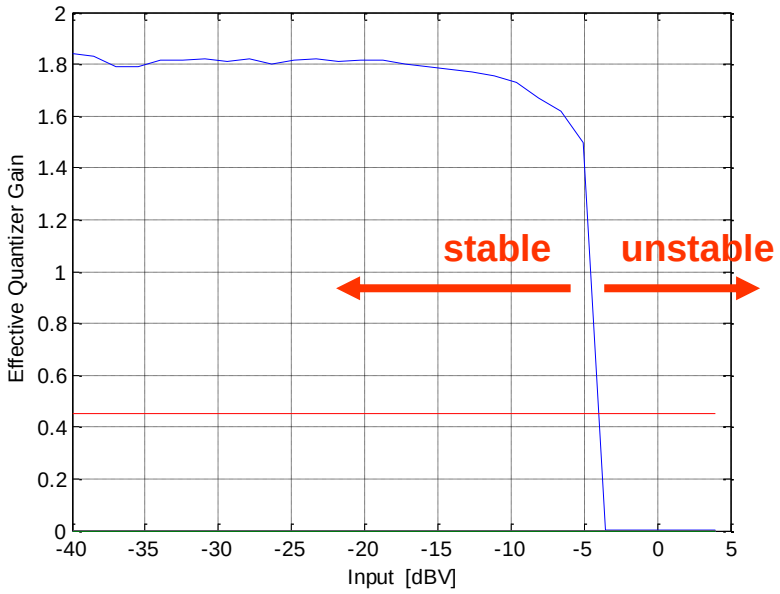
Modulator Root-Locus



sigma_delta_L5_H.m

- As G_{eff} increases, poles of STF move from
  - poles of H(z) (G_{eff} = 0) to
  - zeros of H(z) (G_{eff} = ∞)
- Pole-locations inside unit-circle correspond to stable modulator
- G_{eff} > 0.45 for stability

Effective Quantizer Gain, G_{eff}

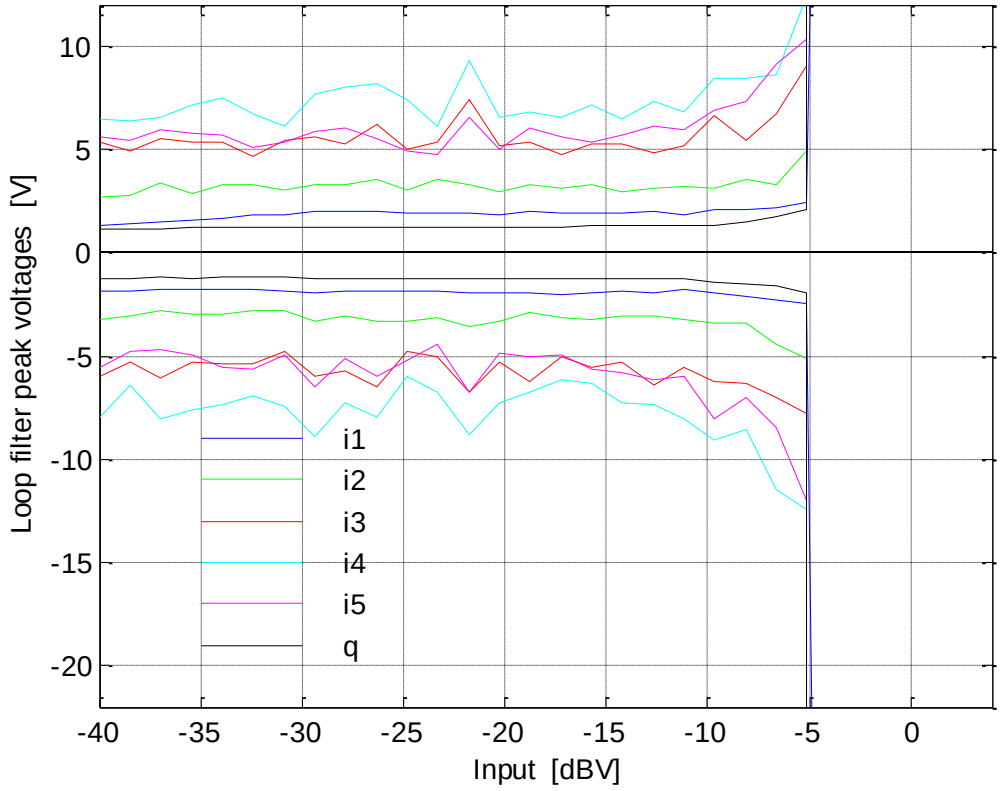


- Large inputs → comparator input grows
- Output is fixed (±1)
- → G_{eff} drops
- → modulator unstable for large inputs

- Solution:
  - Limit input amplitude
  - Detect instability (long sequence of +1 or -1) and reset integrators
  - Note: signals grow slowly for nearly stable systems → use long simulations

sigma_delta_L5_peaks.m

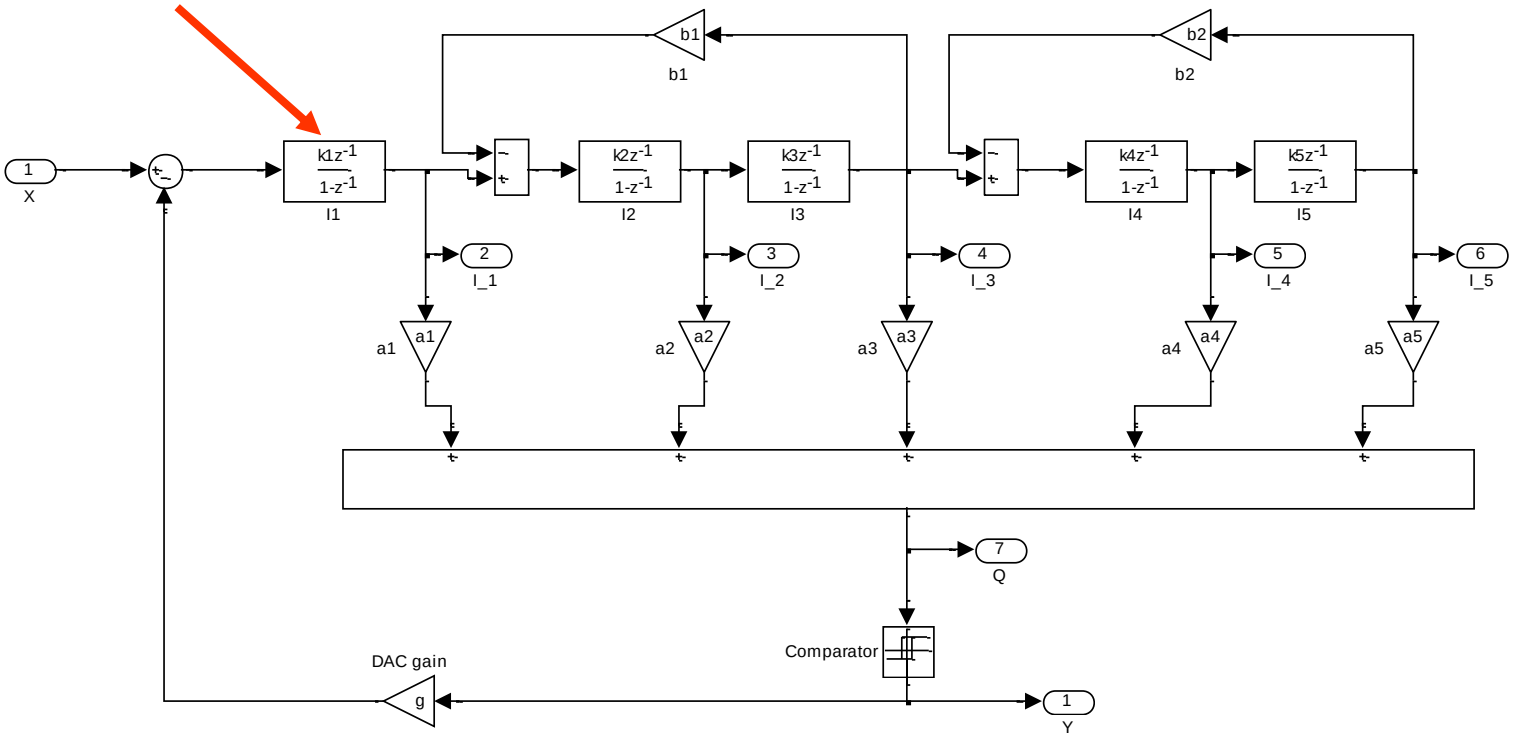
Loop Voltages



- Internal signal amplitudes are week function of input level (except near overload)
- Exceed supply voltage
- Solutions:
  - Reduce V_{ref} ??
  - Scaling

5th Order Modulator – Scaling

Only the sign of Q matters: choose k_1 without changing the 1-Bit data at all

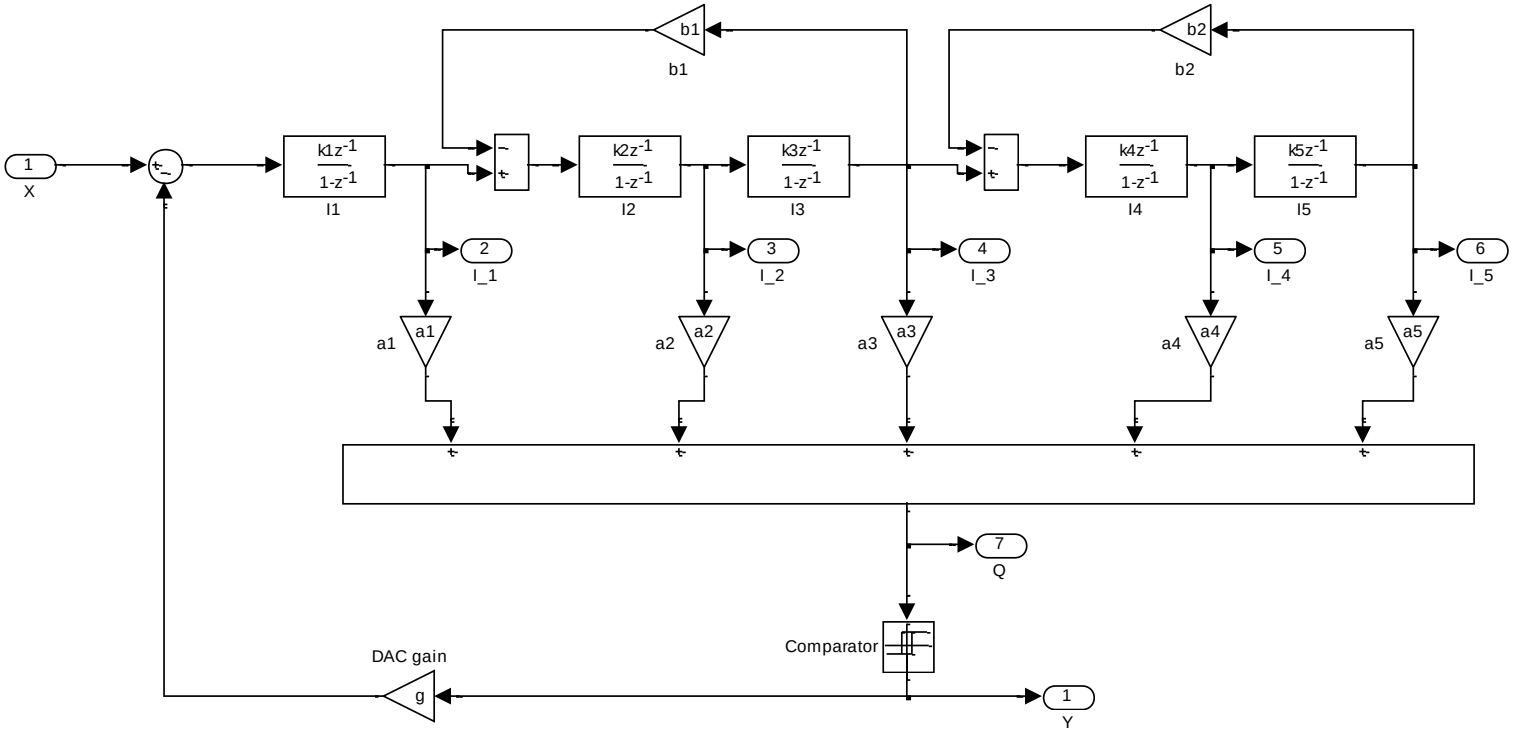


sigma_delta_L5_sim.mdl

Scaling Example

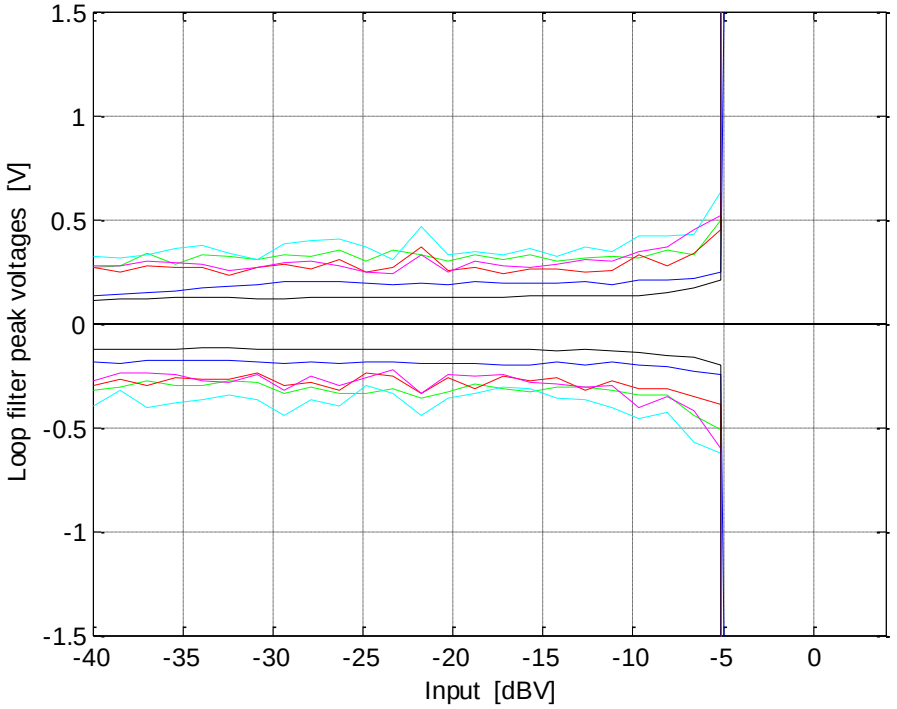
Integrator 3 Output times S:

K3 * S, b1 /S, a3 / S, K4 / S, b2 * S



sigma_delta_L5_sim.mdl

Voltage Scaling



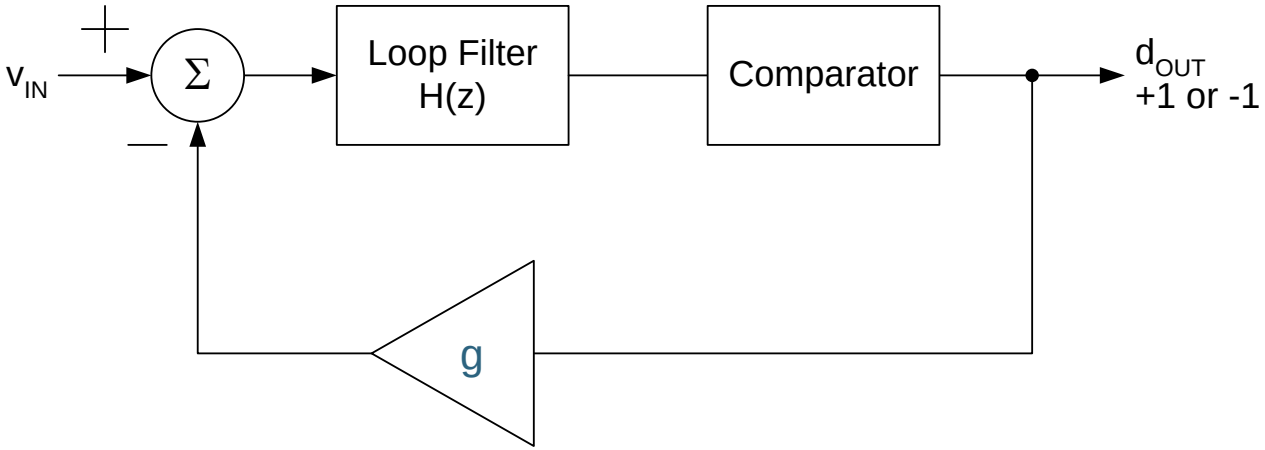

```
k1=1/10;
k2=1;
k3=1/4;
k4=1/4;
k5=1/8;
a1=   1;
a2=1/2;
a3=1/2;
a4=1/4;
a5=1/4;
b1=1/512;
b2=1/16-1/64;
g  =1;
```

- Integrator output range is fine now
- But: maximum input signal limited to -5dB (-7dB with safety) – fix?

Input Range Scaling

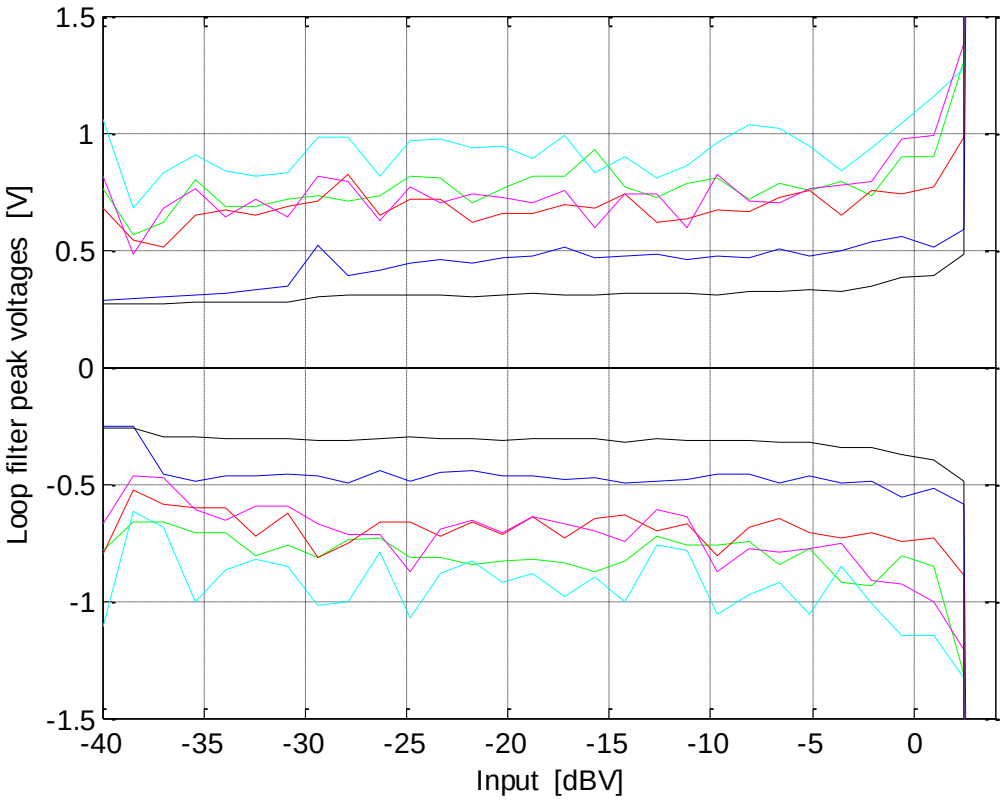
Increasing the DAC levels by g reduces the analog to digital conversion gain:

$$\frac{D_{OUT}(z)}{V_{IN}(z)} = \frac{H(z)}{1+gH(z)} \cong \frac{1}{g}$$



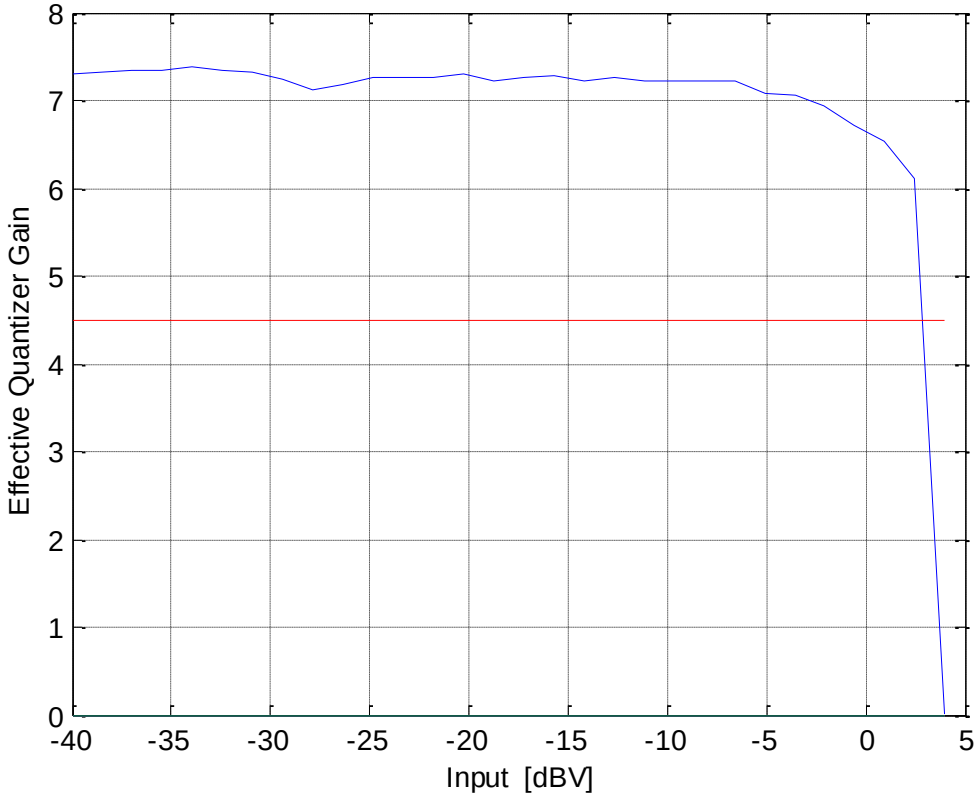
Increasing v_{IN} & DAC level (g) by the same factor leaves 1-Bit data unchanged

Scaled Modulator Model



g = 2.5;

Scaled Model Overload



2dB safety margin for stability

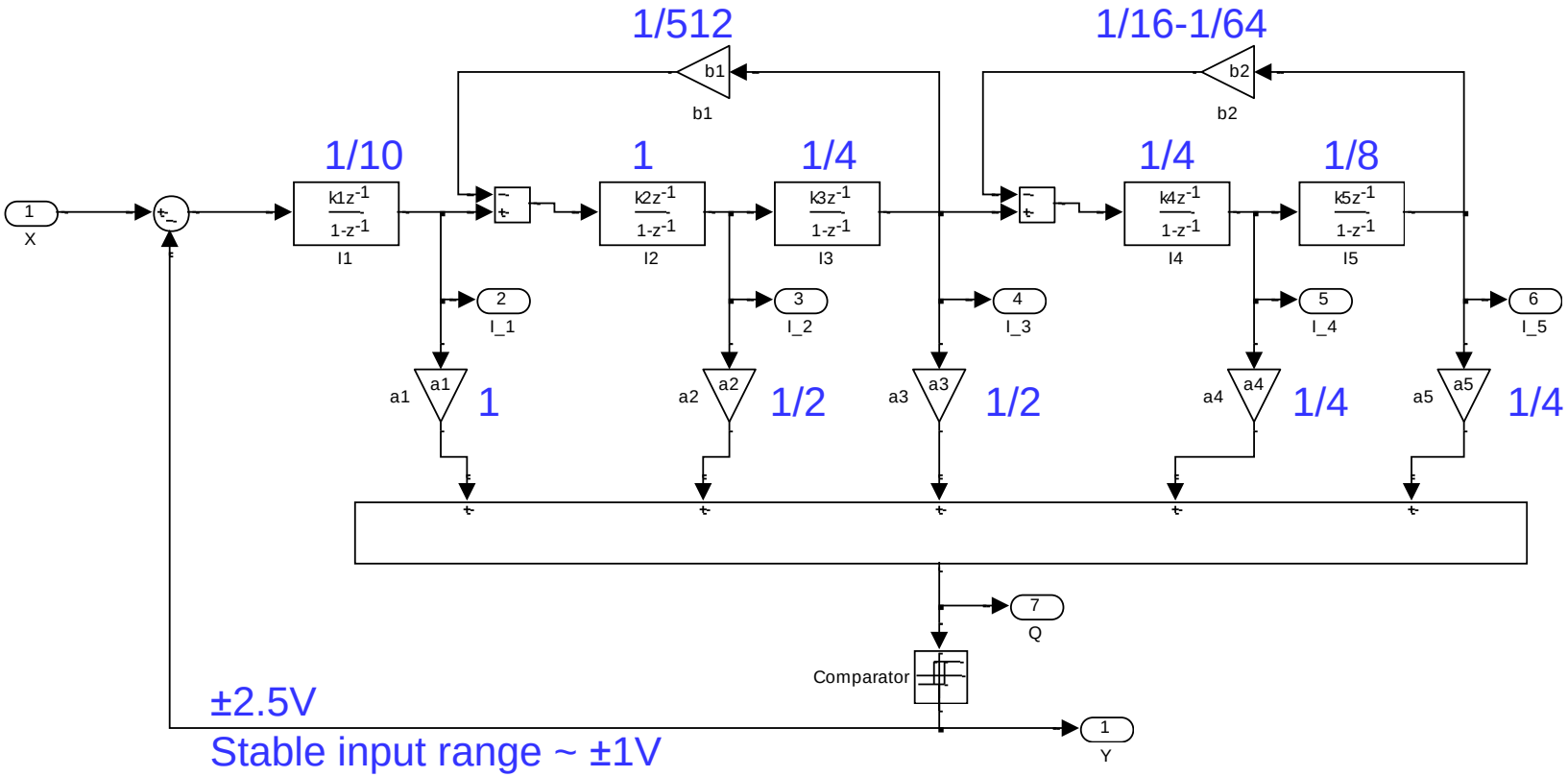
NTF Design

- SQNR is determined by NTF design choices
  - Filter order and shape (e.g. zeros)
  - In-band quantization noise attenuation vs maximum stable amplitude
- Filter shape influence circuit topology
  - Zeros in loop filter → resonator structures
- Manual iteration to maximize SQNR
- Or use MATLAB Delta Sigma Toolbox
  - `NTF = synthesizeNTF(order=3,osr=64,opt=0,H_inf=1.5,f0=0)`
  - `NTF = synthesizeChebyshevNTF(order=3,OSR=64,opt=1,H_inf=1.5,f0=0)`

EE 240C
Analog-Digital Interface Integrated Circuits

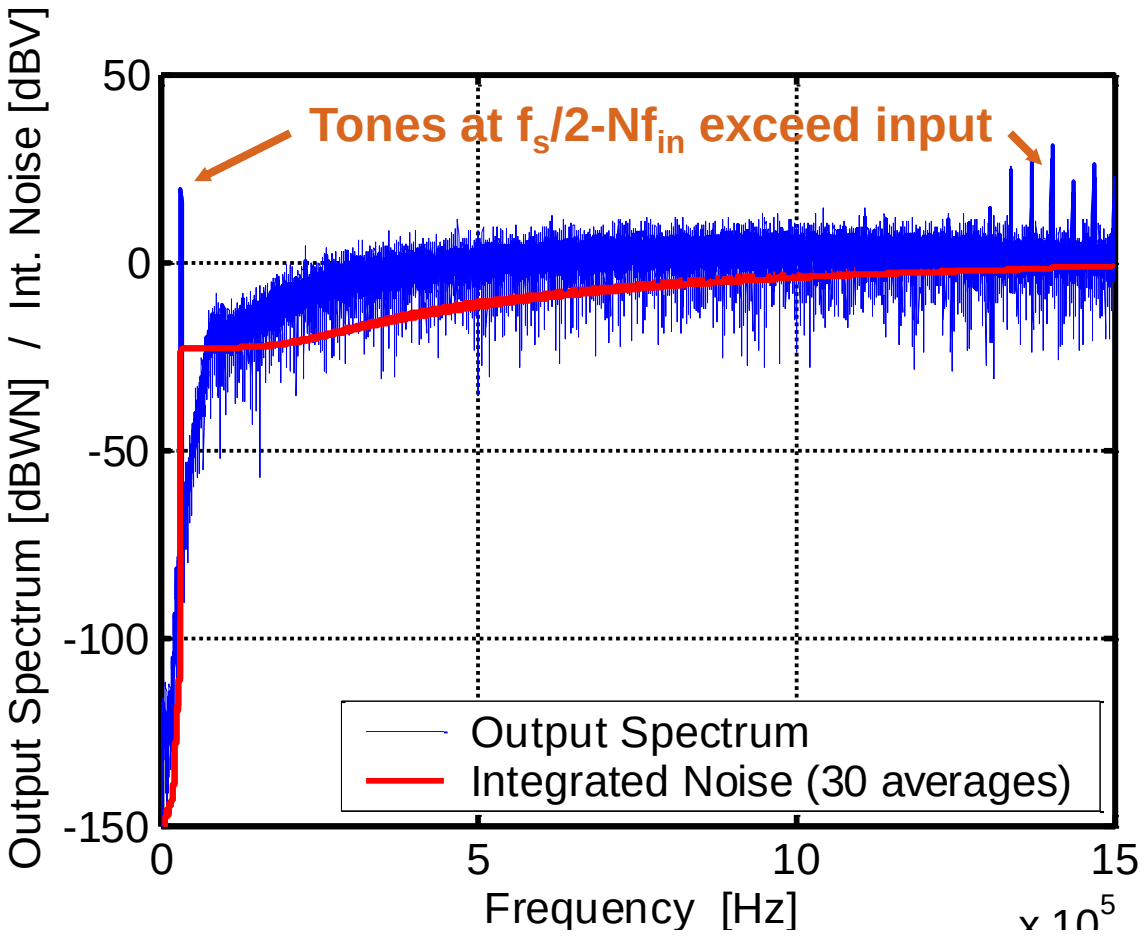
Limit Cycles

5th Order Modulator



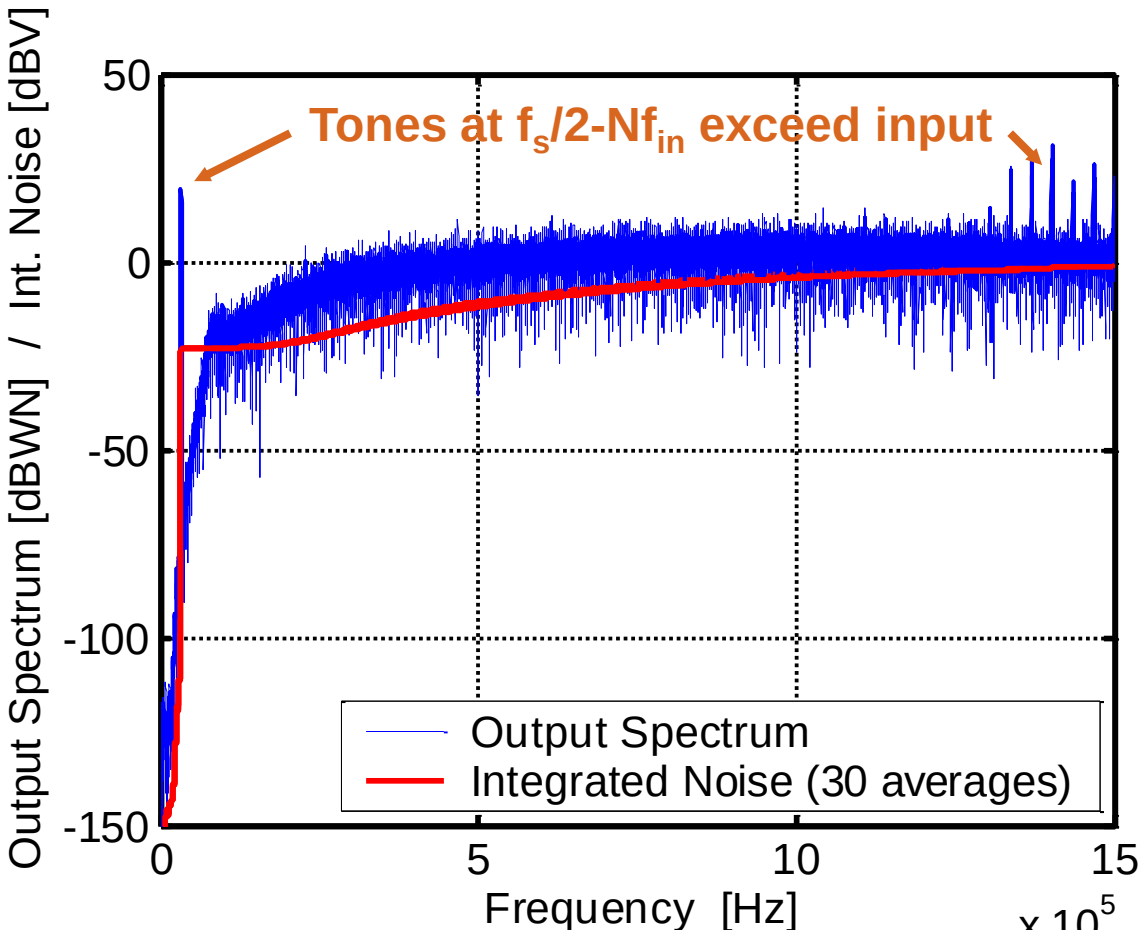
sigma_delta_L5_sim.mdl

Quantization Noise Tones



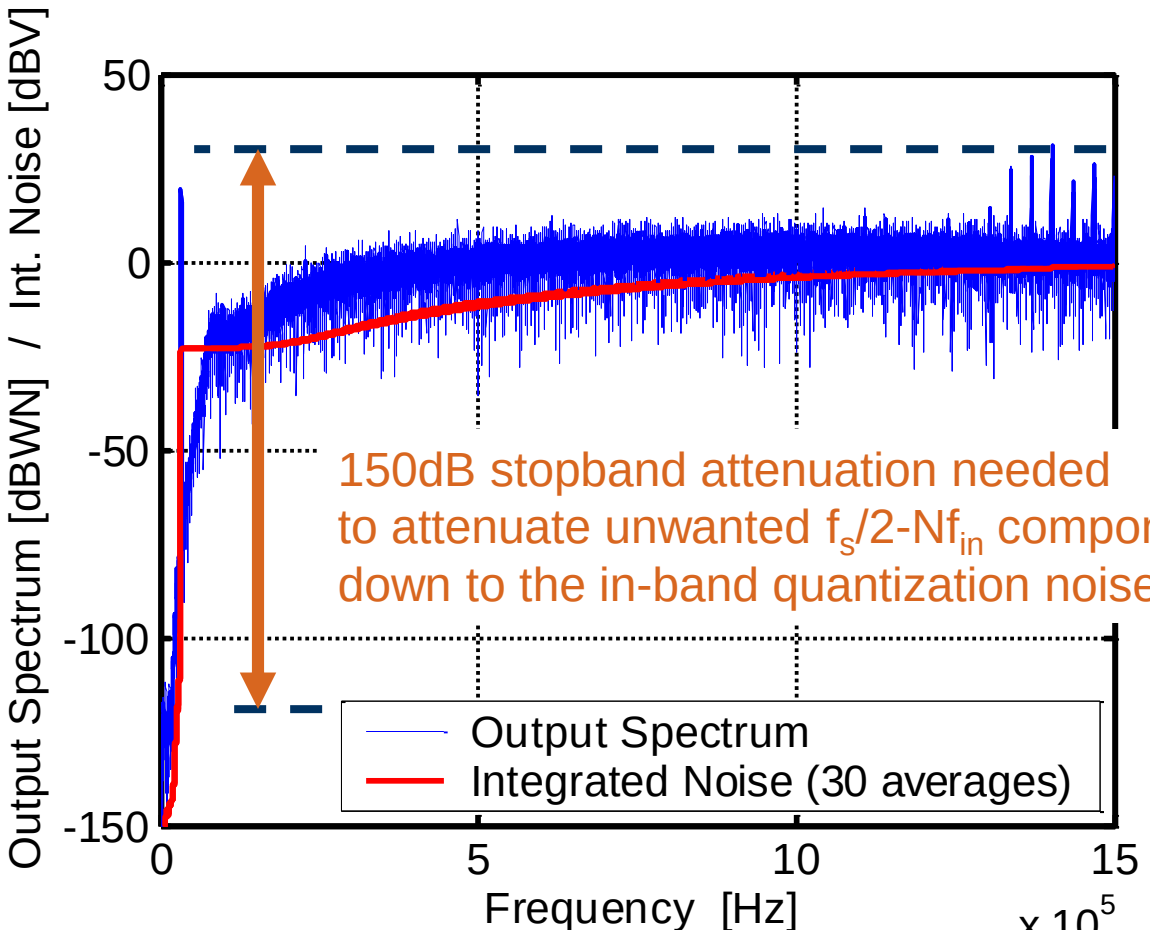
Input: 0.1V, sinusoid
2^{15} point DFT
30 averages

Quantization Noise Tones



Input: 0.1V, sinusoid
2^{15} point DFT
30 averages

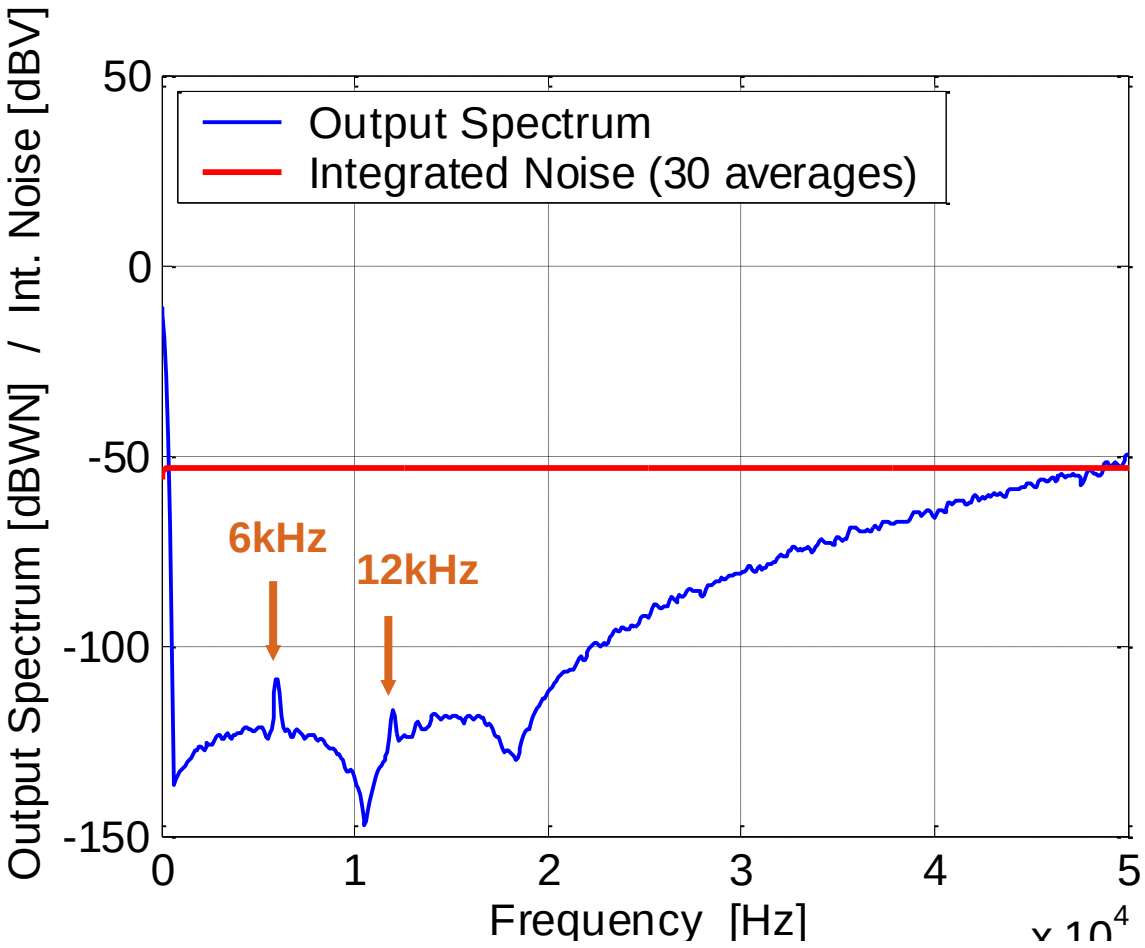
Quantization Noise Filter Requirements



150dB stopband attenuation needed
to attenuate unwanted $f_s/2-Nf_{in}$ components
down to the in-band quantization noise level → digital filter

Input: 0.1V, sinusoid
2^{15} point DFT
30 averages

DC Inputs



2mV DC input
(1V full-scale)

Simulation technique:

A random 1st sample randomizes the noise from DC input and enables averaging. Otherwise the small tones are not visible.

Limit Cycles

- Representing a DC term with a –1/+1 pattern … e.g.

$$\tfrac{1}{11} \rightarrow \Big\{ \underbrace{\underbrace{-1\ +1}_{1}\ \underbrace{-1\ +1}_{2}\ \underbrace{-1\ +1}_{3}\ \underbrace{-1\ +1}_{4}\ \underbrace{-1\ +1}_{5}}_{\langle 0 \rangle}\ +1 \Big\}_{\langle 1/11 \rangle}$$

- Spectrum

$$\frac{f_s}{11} \quad 2\frac{f_s}{11} \quad 3\frac{f_s}{11} \quad \ldots$$

[Eric Swanson]

Limit Cycles

- Fundamental

$$f_\delta = f_s \frac{V_{DC}}{V_{DAC}}$$

$$= 3\text{MHz} \frac{2\text{mV}}{1\text{V}}$$

$$= \underline{6\text{kHz}}$$

- "Tone velocity" (useful for debugging)

$$\frac{df_\delta}{dV_{DC}} = \frac{f_s}{V_{DAC}}$$

$$= \underline{3\text{kHz/mV}}$$

ΣΔ Tones

- Tones follow the noise shape

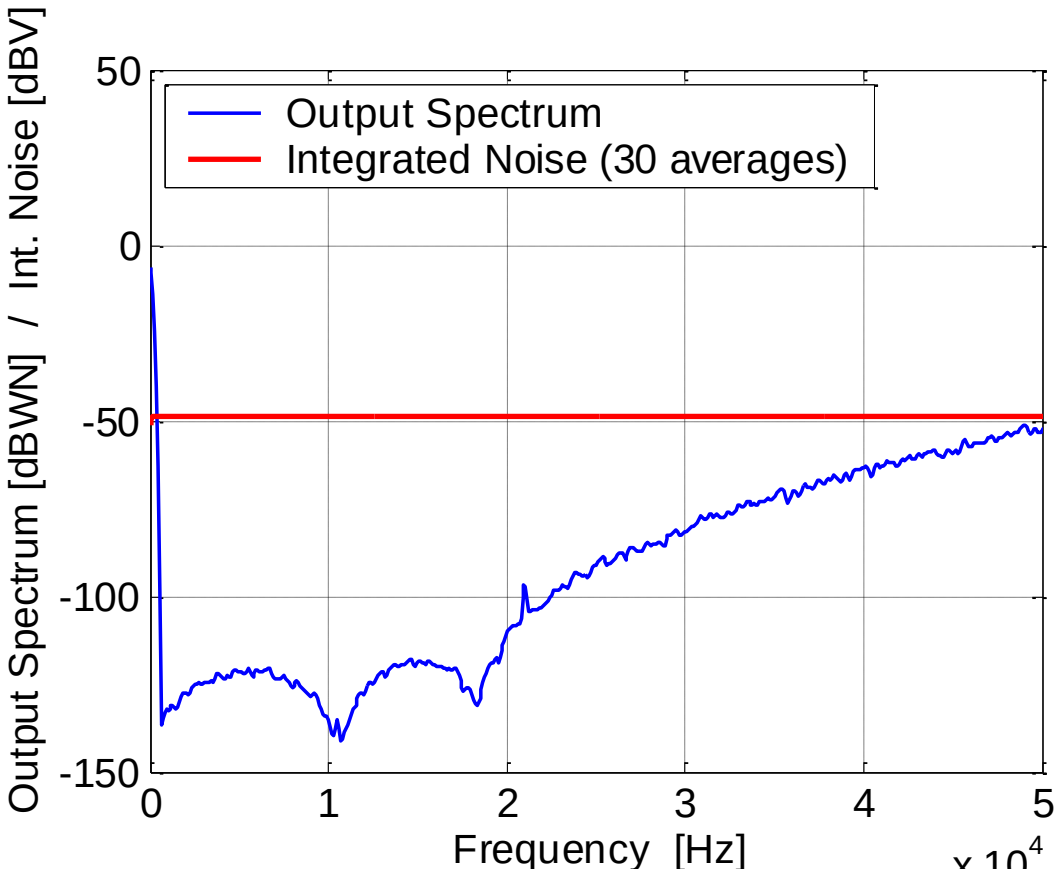
- The fundamental of a tone that falls into a "quantization noise null" disappears …

$$V_{DC} = V_{FB} \frac{f_\delta}{f_s}$$

$$= 1\text{V} \frac{10.5\text{kHz}}{3\text{MHz}}$$

$$= \underline{3.5\text{mV}}$$

ΣΔ Tones



3.5mV DC input

- High ΣΔ loop gain at 10.5 KHz suppresses limit cycle tone
- Tone at 2·10.5 KHz = 21 KHz still visible

ΣΔ Tones

- In-band tones look like signals
- Big problem in some applications
  - E.g. audio → tones below the quantization noise floor can be audible
  - Harmonics below the noise floor in the frequency domain combine to periodic time domain artifacts above the noise floor
- Tones near $f_s/2$ can be aliased down into the signal band
  - Since they are often strong, even a small alias can be a big problem
- Dither can be used to reduce or eliminate in-band tones

EE 240C
Analog-Digital Interface Integrated Circuits

Dither

Dither

- DC inputs can of course be represented by many possible bit patterns
- Including some that are random but still average to the DC input
- The spectrum of such a sequence has no tones
- How can we get a $\Sigma\Delta$ modulator to produce such "randomized" sequences?

Dither

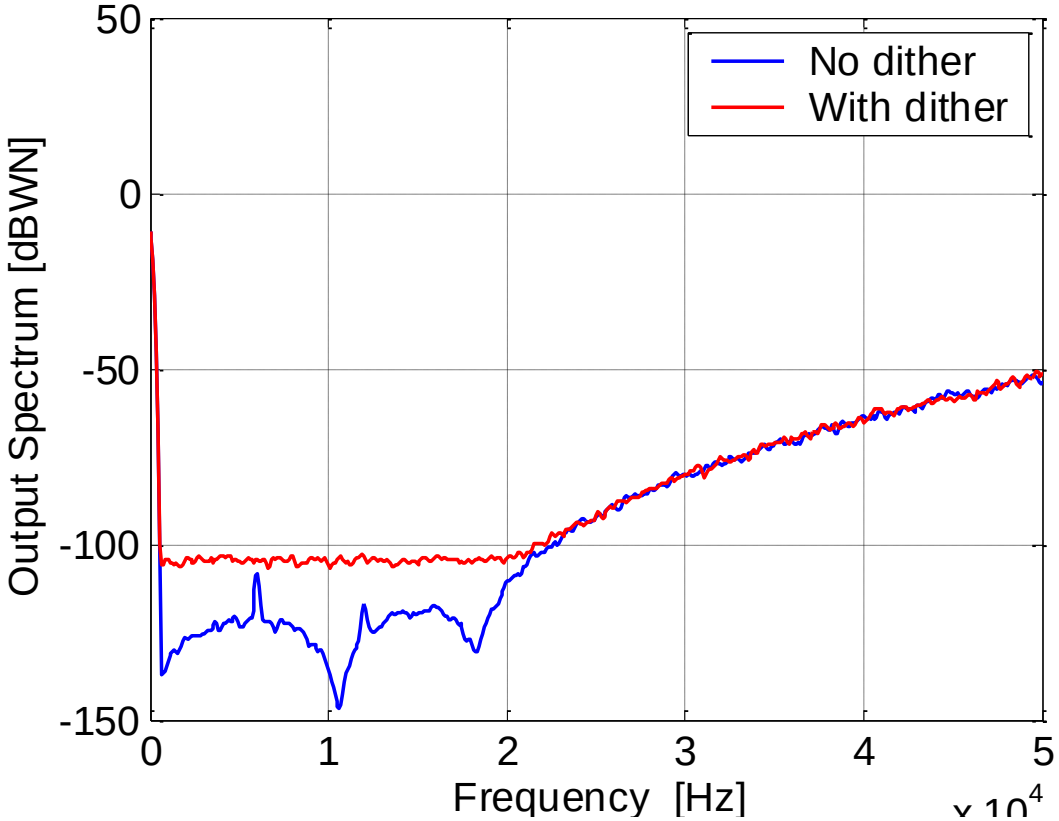
- The target DR for our audio ΣΔ is 16 Bits, or 98dB
- Let's choose the sampling capacitor such that it limits the dynamic range:

$$DR = \frac{\frac{1}{2}(V_{FS})^2}{k_B T / C}$$

$$C = DR \frac{k_B T}{\frac{1}{2}(V_{FS})^2}$$

$$= 10^{9.8} \frac{k_B T}{\frac{1}{2}(1\text{V})^2} = \underline{50.5\text{pF}} \quad \rightarrow \quad \sqrt{\overline{v_n^2}} = \sqrt{\frac{k_B T}{C}} = \underline{9\mu\text{V}}$$

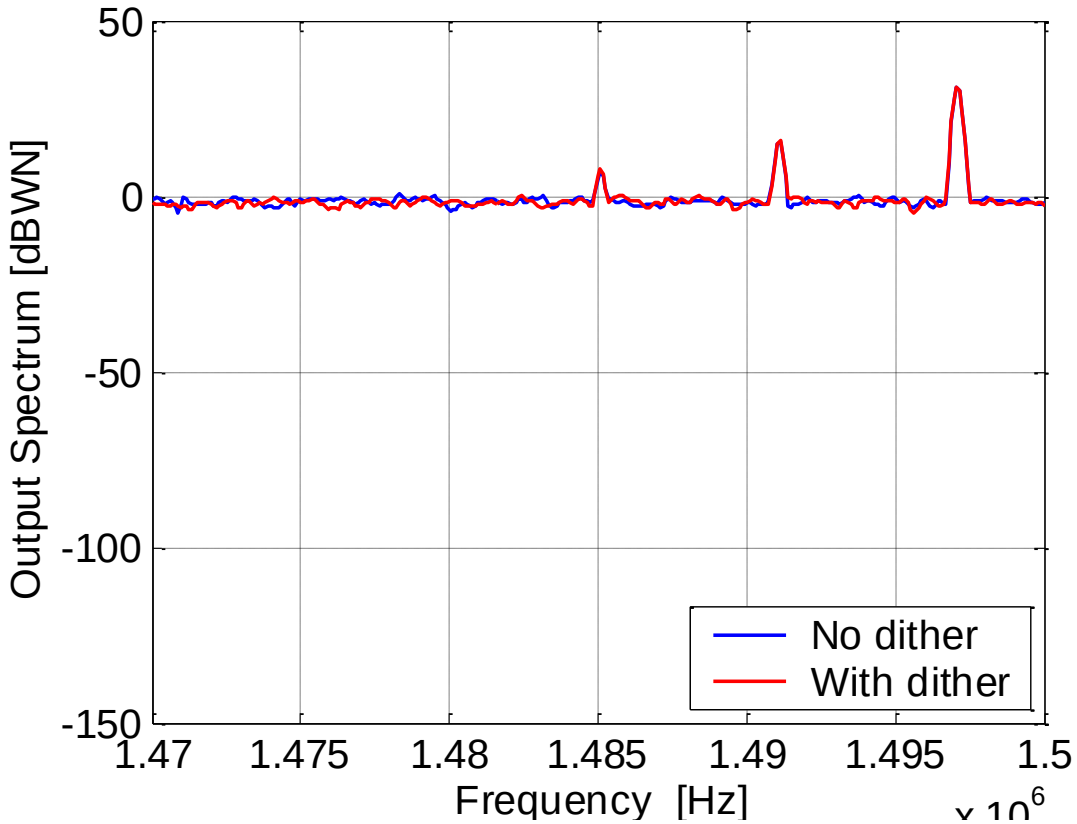
Dither



2mV DC input

- Tones disappear
- Courtesy of the “excessive” SQNR of this design
- Note: they are not just buried
- How can we tell?

Dither



Dither at an amplitude which buries the in-band tones has virtually no effect on tones near $f_s/2$

EE 240C
Analog-Digital Interface Integrated Circuits

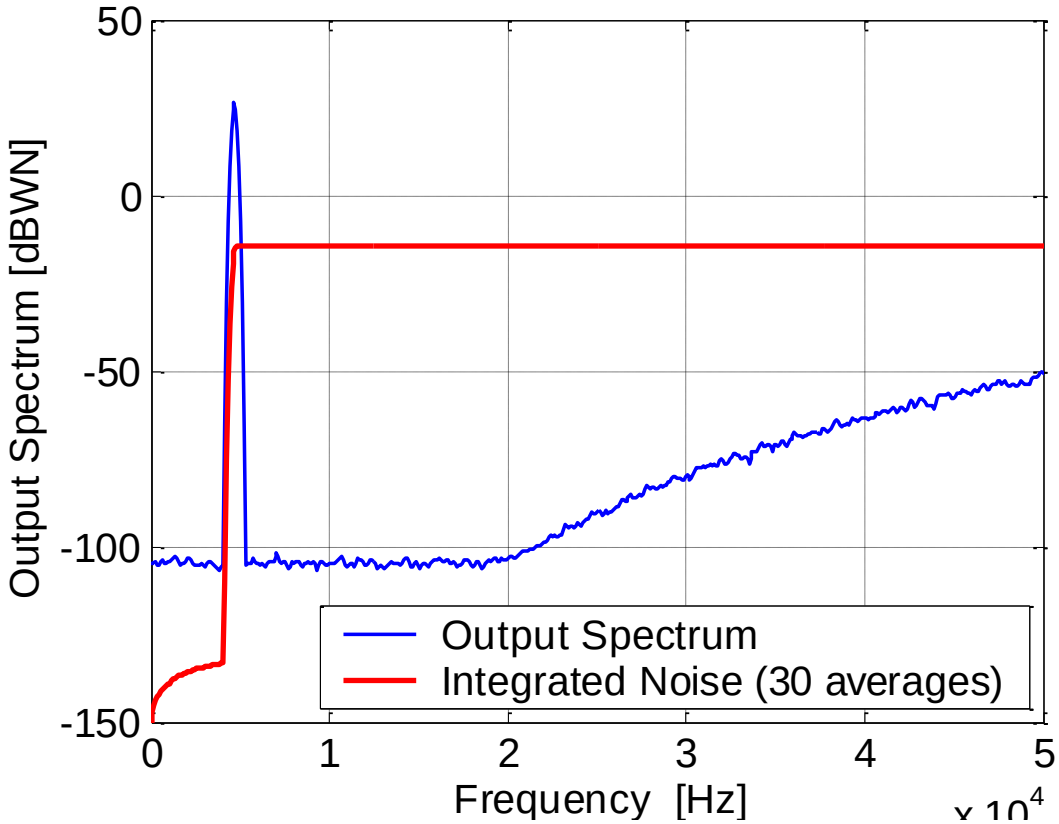
Full-Scale Inputs

Full-Scale Inputs

- With practical levels of thermal noise added, let's try a 5kHz sinusoidal input near full-scale

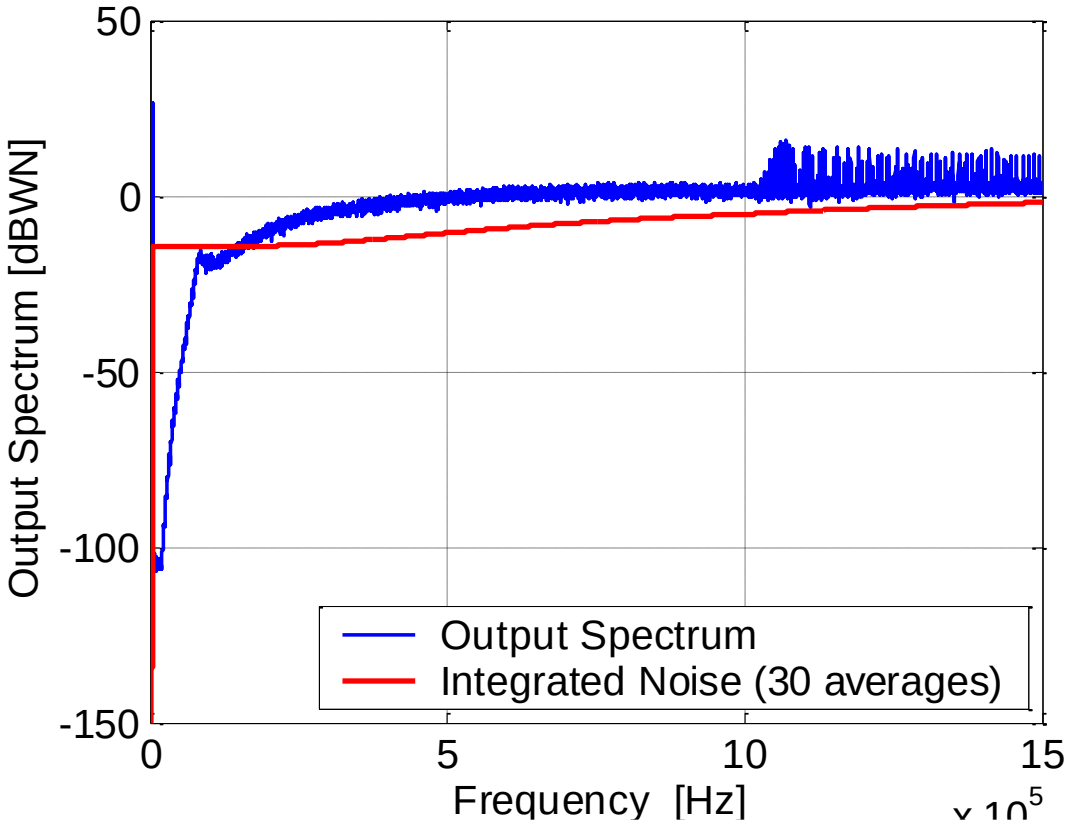
- No distortion is visible in the spectrum
  - 1-Bit modulators are intrinsically linear
  - But tones exist at high frequencies
    → to the oversampled modulator, a sinusoidal input looks like two "slowly" alternating DCs … hence giving rise to limit cycles

Full-Scale Inputs



No distortion
"linear" 1-Bit DAC

Full-Scale Inputs

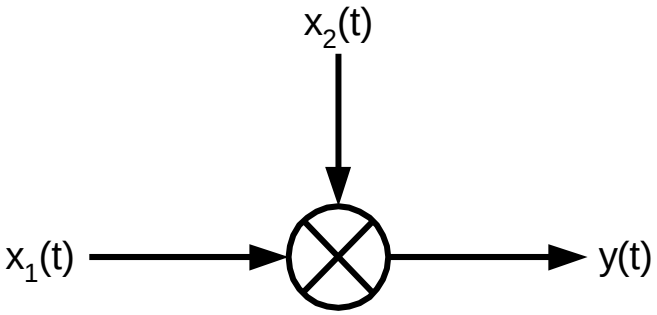


Tones near $f_s/2$

Avoid mixing into signal band

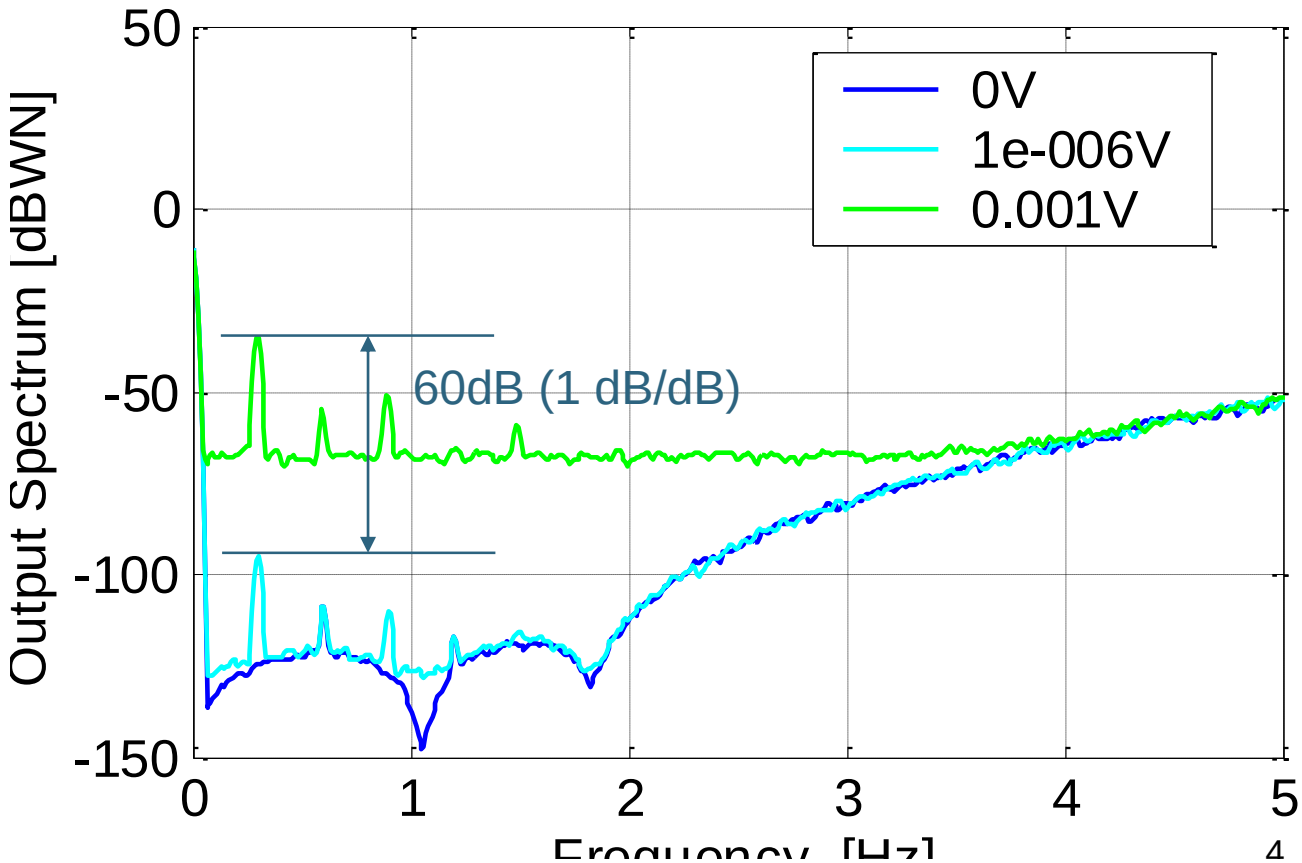
Why would this happen?

AM Modulation



$$x_1(t) = X_1 \cos(\omega_1 t)$$
$$x_2(t) = X_2 \cos(\omega_2 t)$$
$$x_1(t) \times x_2(t) = \frac{X_1 X_2}{2}\left[\cos(\omega_1 t + \omega_2 t) + \cos(\omega_1 t - \omega_2 t)\right]$$

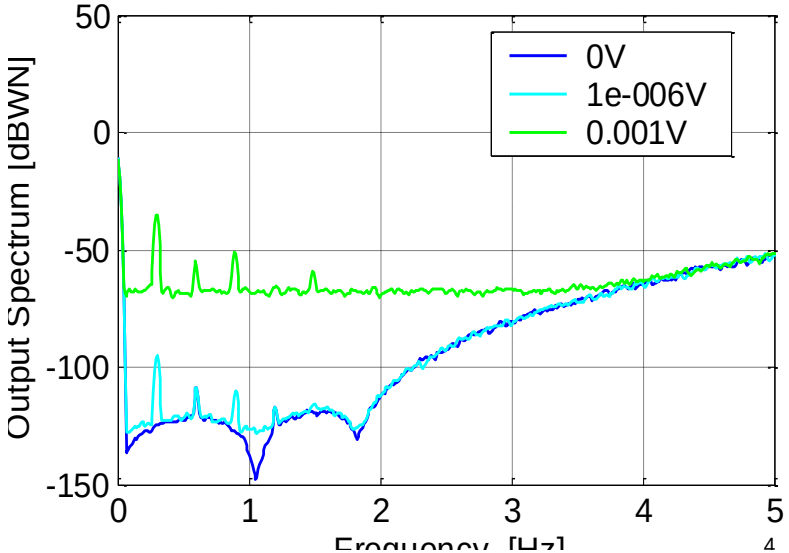
V_{ref} Interference

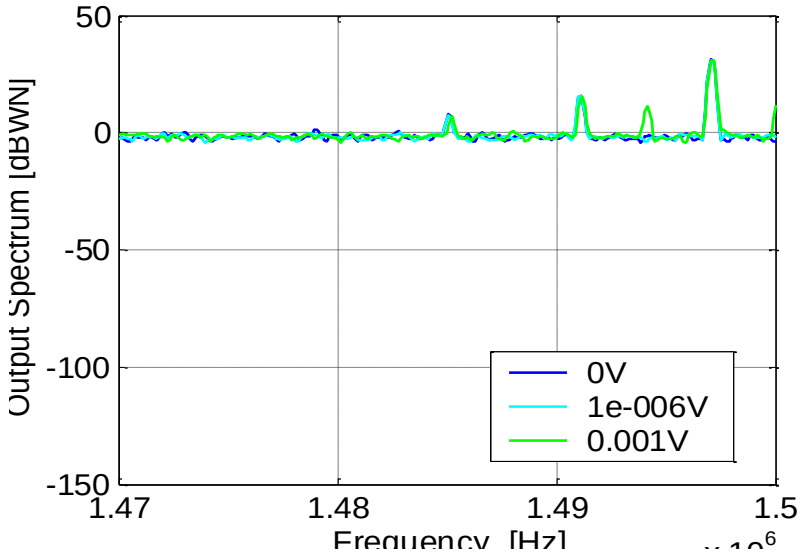


1μV interference suffices to create strong in-band tones

1mV interference also rises the noise floor

V_{ref} Interference





Symmetry of the spectra at $f_s/2$ and DC confirm that this is AM modulation

EE 240C
Analog-Digital Interface Integrated Circuits

Decimation Filters

Decimation filters for ΣΔ ADCs

- Digital decimation filters
  - Aliasing in the analog domain
  - Aliasing in the digital domain
  - Coefficient precision and gain scaling

- Digital arithmetic throughput calculations
  - One-stage decimation
  - Linear phase implications
  - Multi-stage decimation

Ref: R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital

Signals – A Tutorial Review", Proc. IEEE, 69, pp. 300-331, March 1981.

ΣΔ Analog-to-Digital Converters

- A ΣΔ Analog-to-Digital Converter (ΣΔ ADC) combines
  - An analog ΣΔ modulator which produces an oversampled output stream of 1-bit digital samples
  - A digital decimation filter which takes the 1-bit modulator output as its input and
    - Filters out out-of-band quantization noise
    - Filters out unwanted out-of-band signals present in the modulator's analog input
    - Lowers the sampling frequency to a value closer to 2X the highest frequency of interest

Decimation Filters for ΣΔ ADCs

- Commercial DSPs aren't designed to handle 1-bit input samples at oversampled data rates
  - A 400Mip DSP only executes 133 instructions per 3MHz sample
- DSPs are designed to handle 16+ bit wide data words at Nyquist-like sampling frequencies
- ΣΔ decimation filters bridge the speed/resolution gap

Aliasing in the Analog Domain

- An analog filter preceding the ΣΔ modulator is required to reject aliases that fold into the signal band
- Example:
  - f_s = 3MHz, B = 20kHz
    → 2.98MHz aliases to 20kHz
  - 1st order RC LPF with 30kHz cutoff has only 40dB attenuation at 3MHz
  - Is this sufficient?
    - Depends on application
    - Microphones produce negligible output at 3MHz

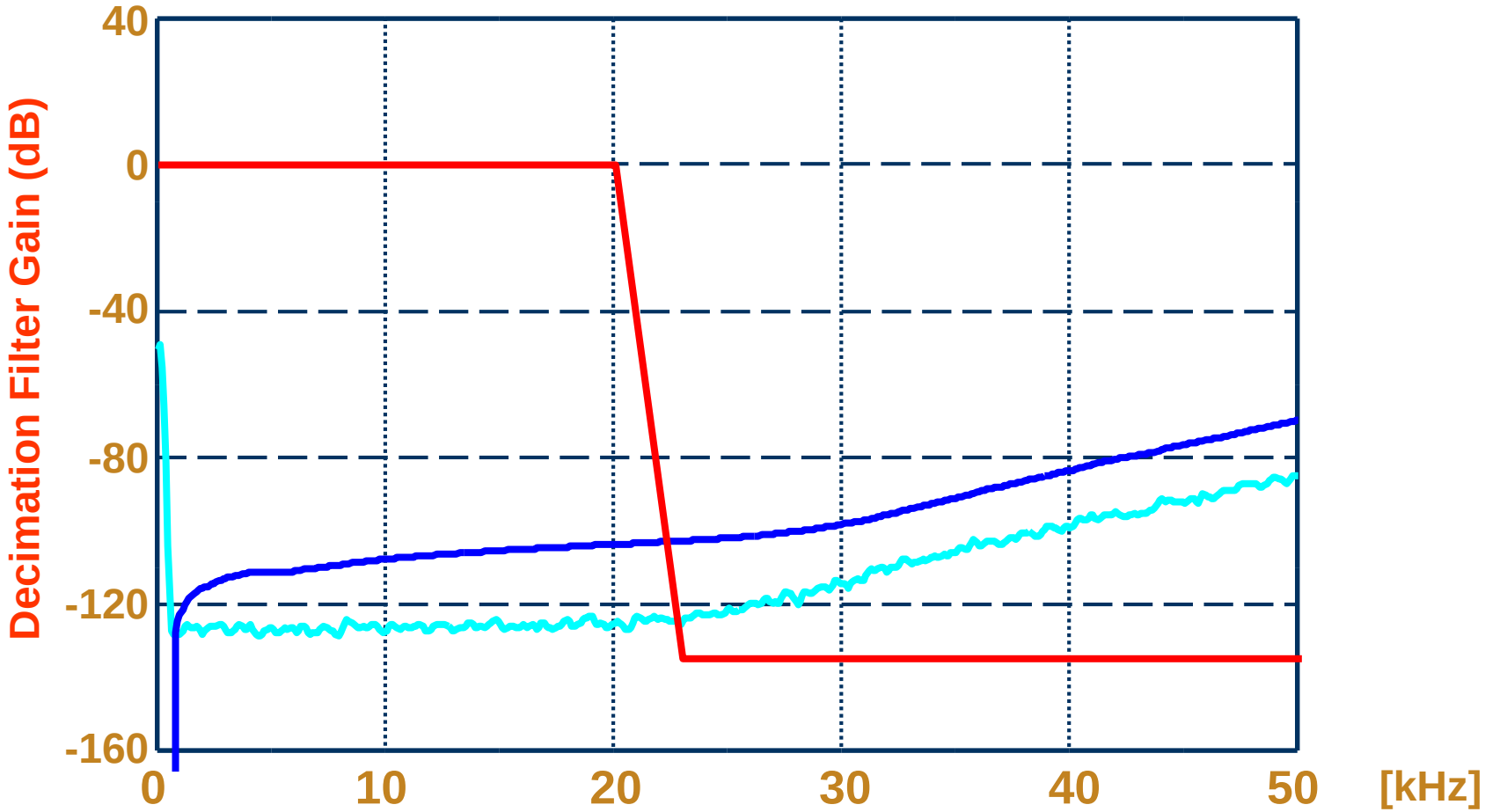
EE 240C
Analog-Digital Interface Integrated Circuits

Digital Decimation Filter

Aliasing in the Digital Domain

- The digital decimation filter following the SD modulator rejects quantization noise and out-of-band signal components to well below the noise floor
- Example:
  - f_s = 3MHz
  - $f_N = f_s/64$ = 46.875kHz
  - 135dB attenuation from for frequencies > $f_N/2$
  - Digital filters can readily achieve this
  - Filter coefficient precision:
    - Rule of thumb: 6dB/bit attenuation
    - 135/6 = 22.5 bit → use 24 bit coefficients

Target Filter Response



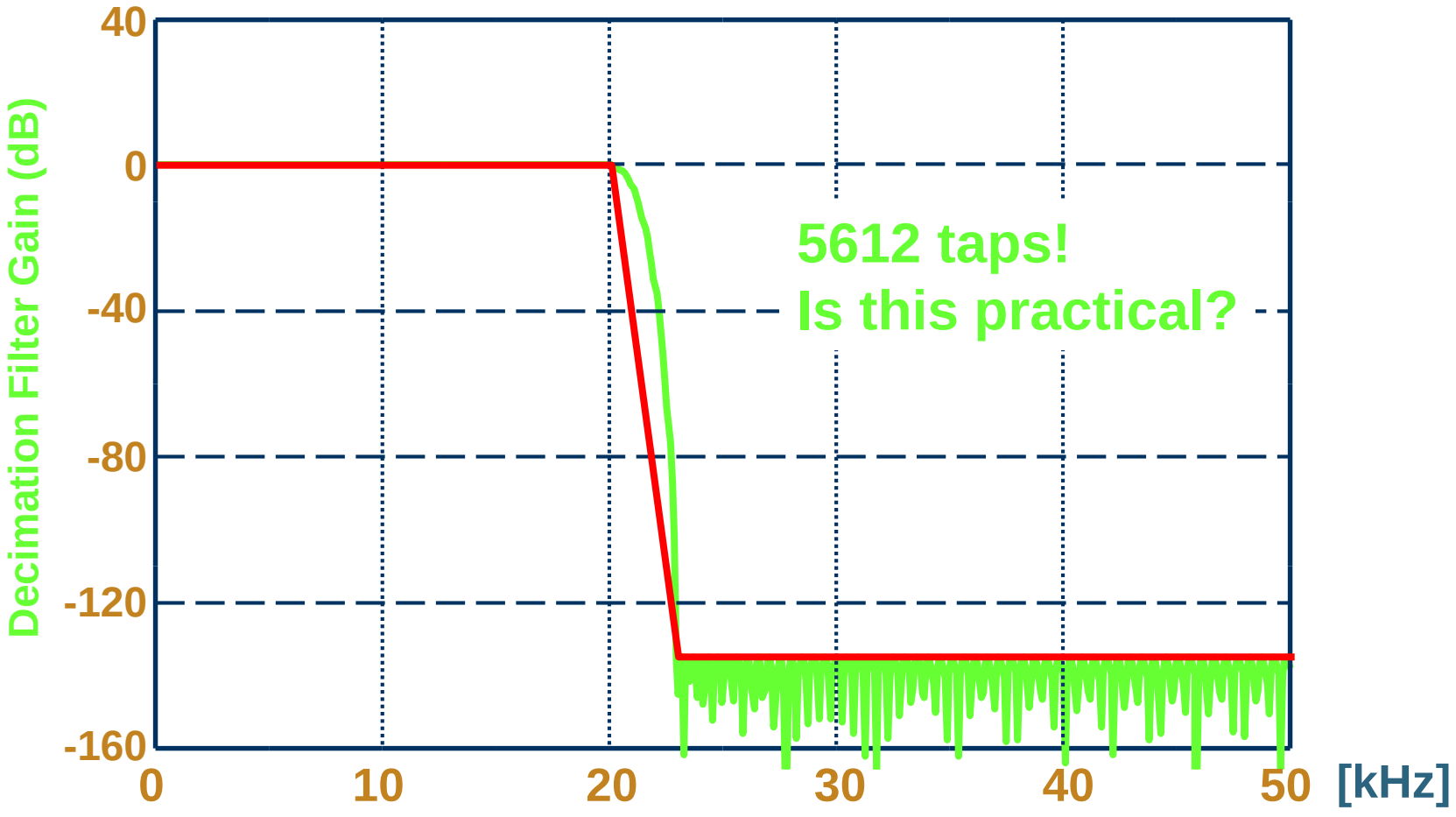
[Eric Swanson]

Decimation Filter Synthesis

- We will design several increasingly more efficient filters

- "Filter #1"
  - 0.00±0.01dB gain from 0–20kHz
  - 135dB stopband attenuation from 23–2977kHz
  - Linear phase
  - Synthesize with Parks–McClellan algorithm
    - MATLAB "remez"
    - 5612 tap FIR filter

Filter #1 Target & Actual Response



[Eric Swanson]

Filter #1

- A classical 5612-tap, f_S=3MHz FIR filter would require a 5612*3MHz = 16.8GHz multiply-accumulate (MAC) rate
- Optimizations:
  1. Decimation by 64 discards 63 out of 64 outputs:
     - No point to compute discarded samples
     - MAC rate reduced to 263MHz (16.8 GHz / 64)
  2. Linear phase filter coefficients are symmetrical
     - Cut coefficient ROM size in half to 2806
     - Perform addition before multiplication
       → 132MHz MAC rate (16.8 GHz / 64 / 2)
  3. Modulator output is 1-bit signal
     - Discard multiplier altogether, only accumulator is needed

Filter #1

- The second key factor that makes this FIR filter unusual is that it needs no hardware multiplier at all
  - Input data is only 1-bit wide
  - The "multiplier" merely adds or subtracts coefficients from the accumulator

- 263MHz begins to seem reasonable, but we can use another simple trick to reduce power further …

Coefficient Symmetry

- Linear phase filter coefficients are symmetric around the middle of the impulse response
- We'd never waste ROM to store all 5612 coefficients when only 2806 are unique
- A 5612x1b data memory allows us to exploit coefficient symmetry to reduce "multiply"-accumulate rates by another 2X …

Filter #2

- Filter complexity is a strong function of the ratio of transition band width to sampling rate
- Transition bands:
  - Filter #1: 20 … 24kHz
  - Filter #2: 20 … $f_N/2$–20kHz = 26kHz
    - A 25kHz tone aliases to f_N–25kHz = 21kHz
    - Attenuation << 135dB but 21kHz is not audible …
    - Additional quantization noise is negligible
    - Remez returns 2406 taps (instead of 5612)
    - MAC rate drops to
      132MHz * 2406 / 5612 = 57MHz

EE 240C
Analog-Digital Interface Integrated Circuits

Digital Filter Implementation Considerations

FIR Arithmetic Throughput

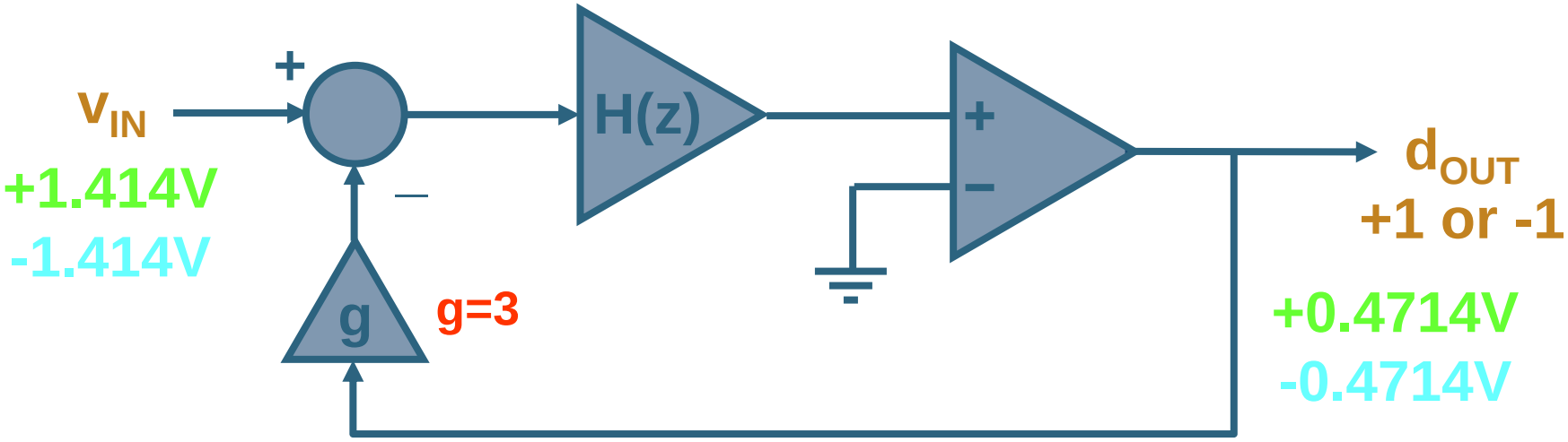
- FIR filters with 1-bit input data don't need traditional hardware multipliers
  - Use add/subtract/do nothing accumulators
- How wide should these accumulators be?
  - What coefficient precision is needed?
  - What output resolution should we use?
  - Let's look at a Filter #2 implementation …

FIR Implementation

- Digital filters usually come with bit-width' that are multiples of 4
- Rounding to 16 Bits would lower the SNR below 98dB
- Let's try a 20-bit filter for our 16-bit ADC
  - $2^{20}=1048576$
  - Each LSB is 1ppm of the ADC input range
- Let's look at the mapping of a 1Vrms full scale sine-wave to digital output values
  - Before we set filter gain levels, we need to review modulator outputs

Modulator Outputs

Positive and negative peaks of a 1Vrms full-scale sinewave correspond to levels shown below:



[Eric Swanson]

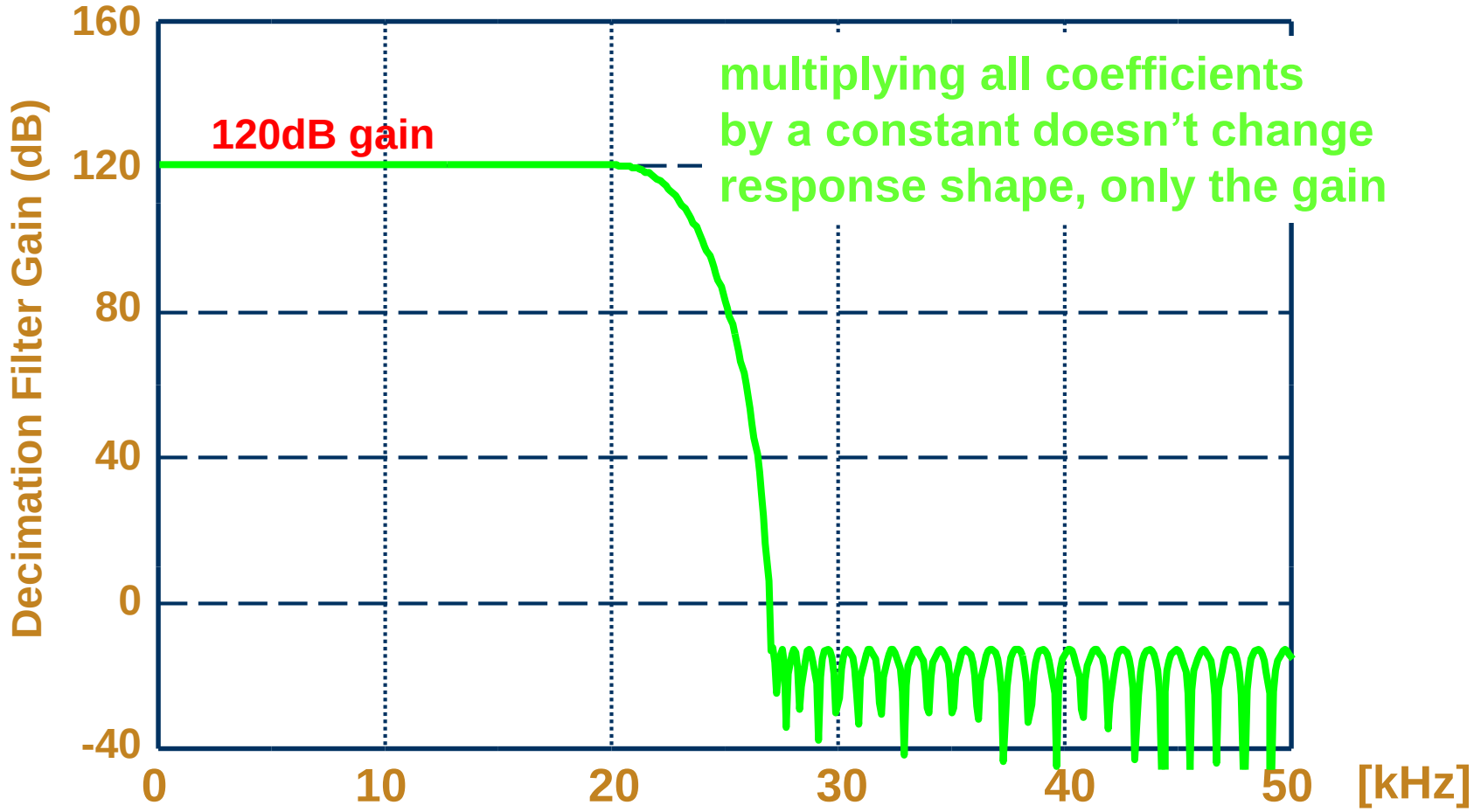
Decimation Filter Gain

- "Gain scaling" in the decimation filter maps the ±0.4714 modulator average output at signal peaks to the 20-bit digital full-scale range of $\pm 2^{19}$
  - Ideal decimation filter dc gain is $2^{19}/0.4714=1112000=120.9\text{dB}$
  - To allow for offsets, etc., we'll use a slightly smaller gain of $2^{20}=120.4\text{dB}$

- An FIR filter's dc gain equals the sum of its coefficients
  - Let's adjust Filter #2's coefficients accordingly …

Ref: Nav Sooch, "Gain Scaling of Oversampled Analog-to-Digital Converters", U.S. Patent 4851841, 1989.

Filter #2 Response



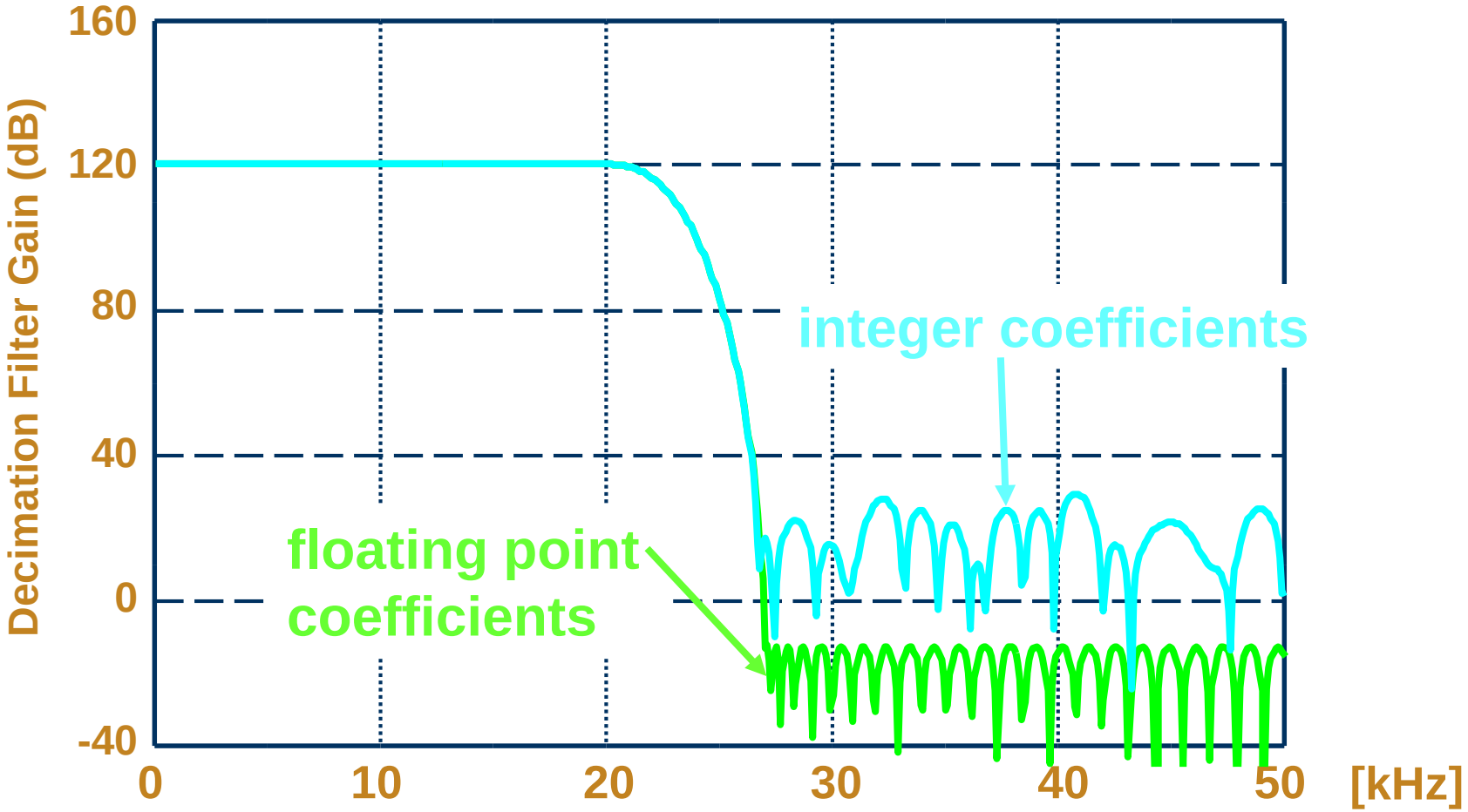
[Eric Swanson]

Filter #2 Response

- The gain adjustment is correct, but coefficients are still floating point

- Rounding these coefficients to the nearest integer using MATLAB's round() function yields the following response …

Filter #2 Responses



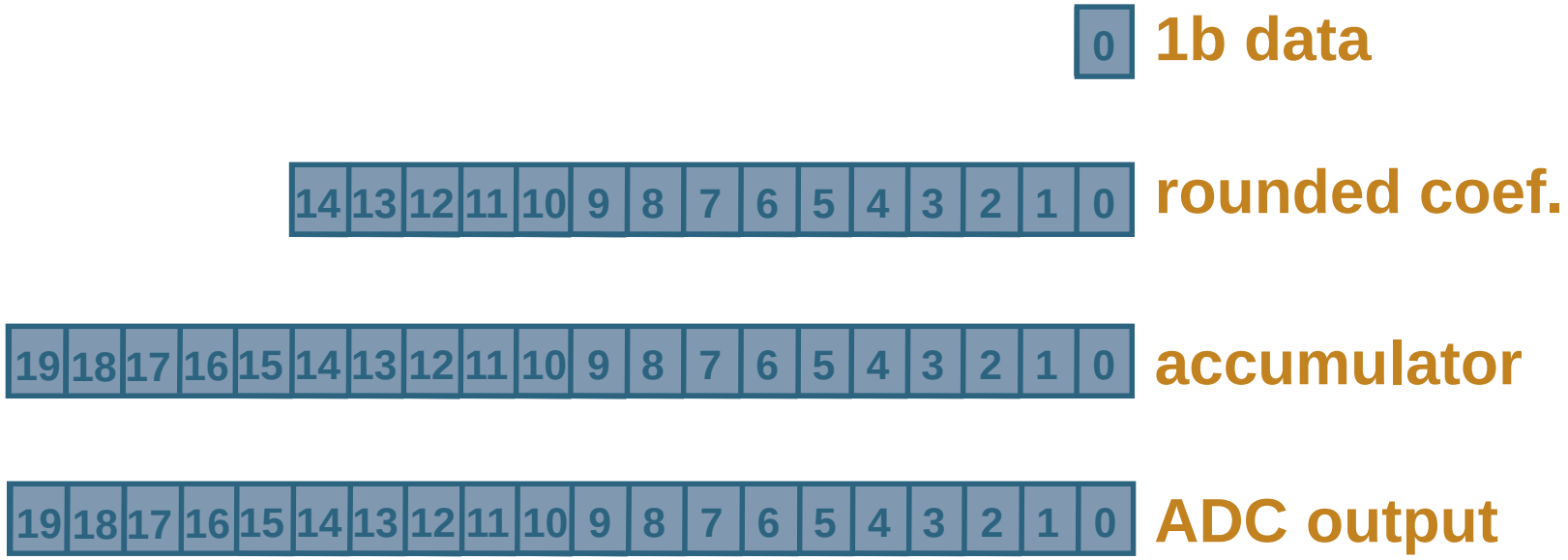
[Eric Swanson]

Filter #2 Responses

- The stopband attenuation drops from 135dB to about 90dB
- Problem is obviously coefficient precision
- Check the integer coefficients
  - The biggest one is +15715
  - The smallest one is -3332
  - That's only 14-15b of coefficient precision, commensurate with ~90dB attenuation
- When 2406 coefficients sum to 2^{20}, the biggest coefficient is pretty small

Filter #2 Bit Map

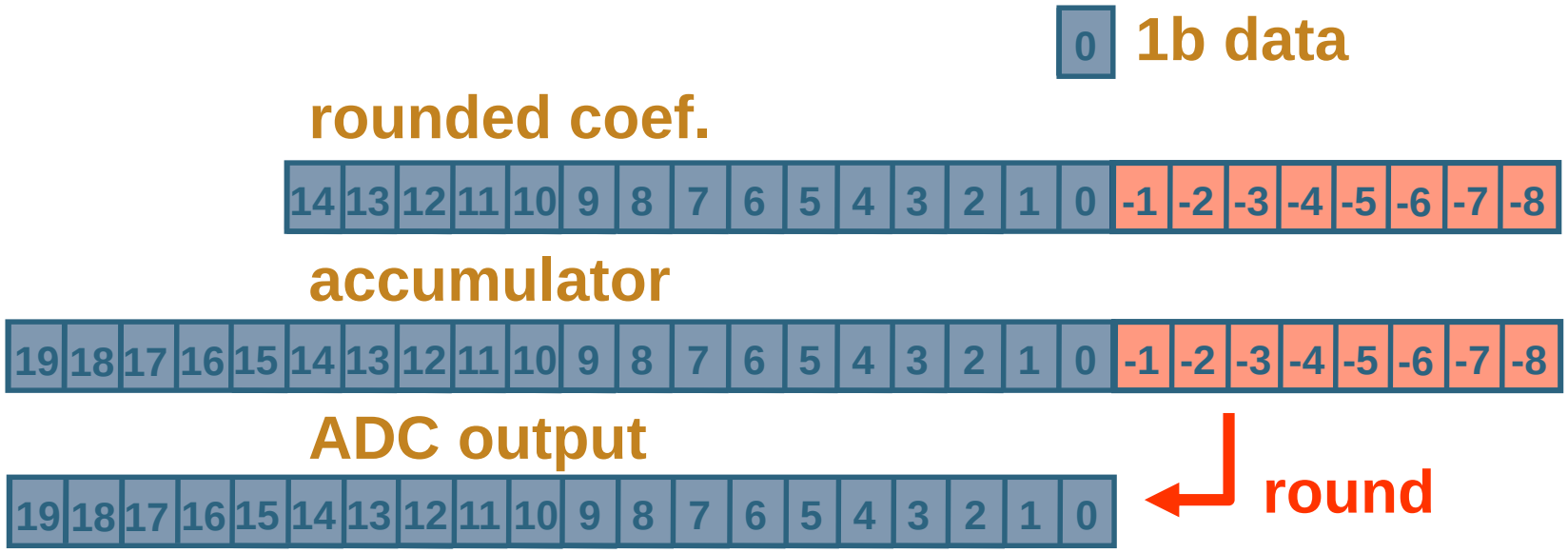
Let's look at the digital scaling in our defective filter :



[Eric Swanson]

Filter #2 Bit Map

To add coefficient resolution, we'll add 8 coefficient bits below the 2^0 point:



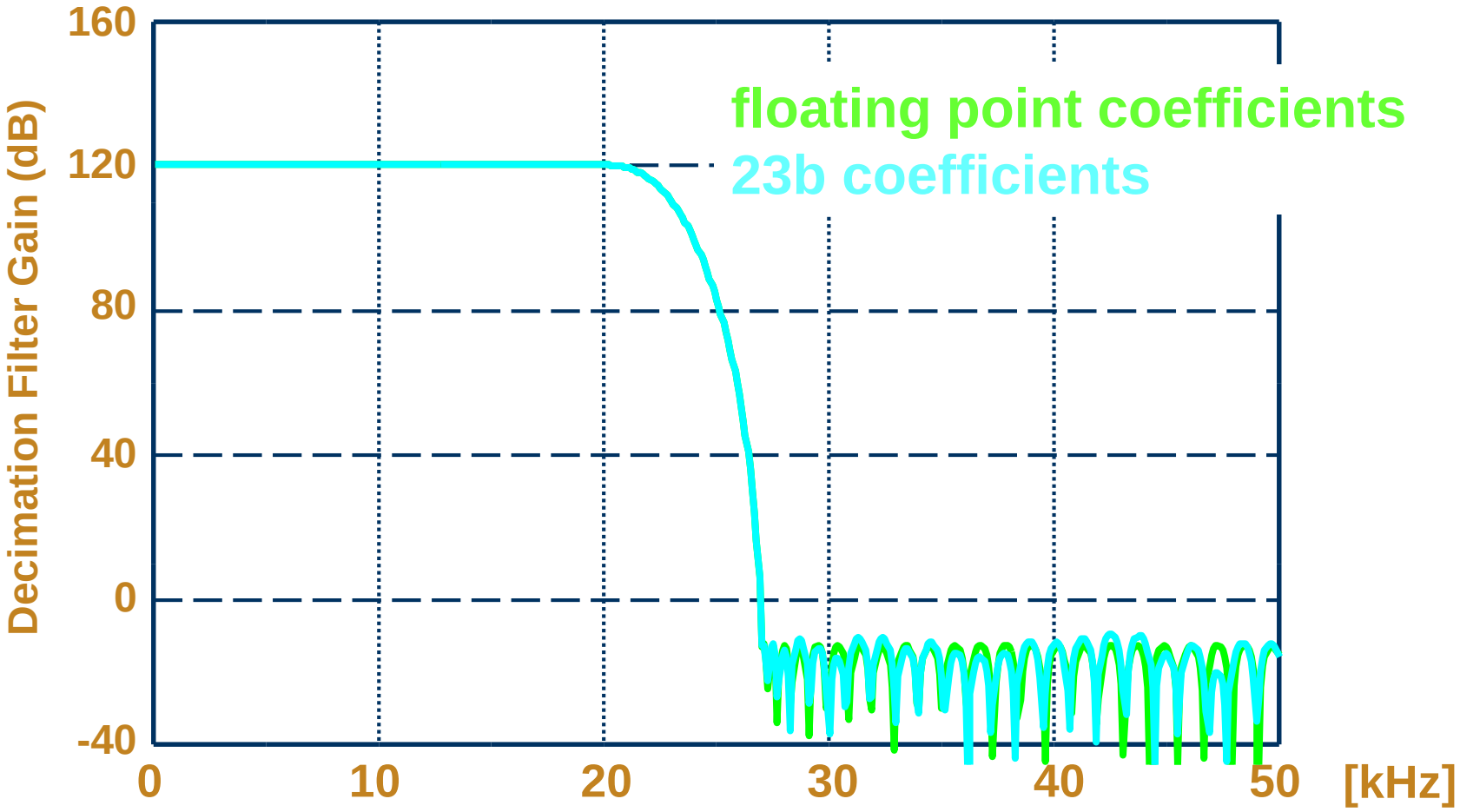
Filter #2 Bit Map

- Higher-precision coefficients are produced with
  coef = round(256*coef)/256 operation

- The 23b fixed point coefficient magnitude response appears on the following slide …

- Rounding of the 28b accumulator to produce the 20b ADC result adds 20b quantization noise
  - At -122dBFS, that's insignificant for a 103dB dynamic range ADC

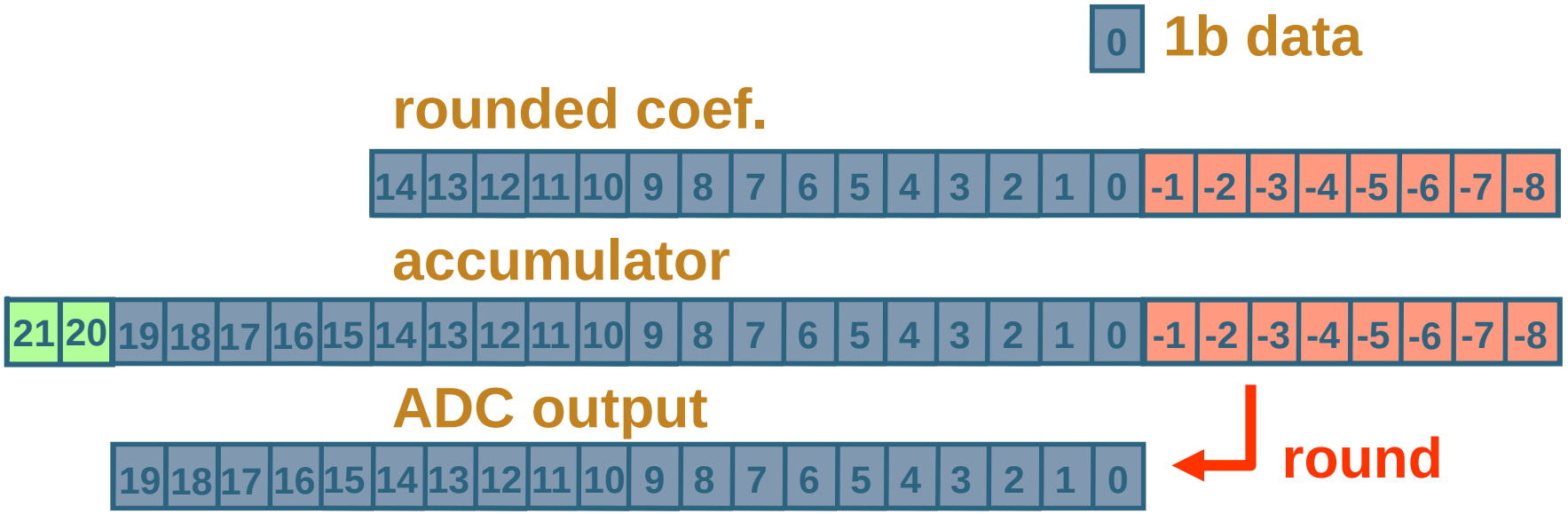
Filter #2 Responses



[Eric Swanson]

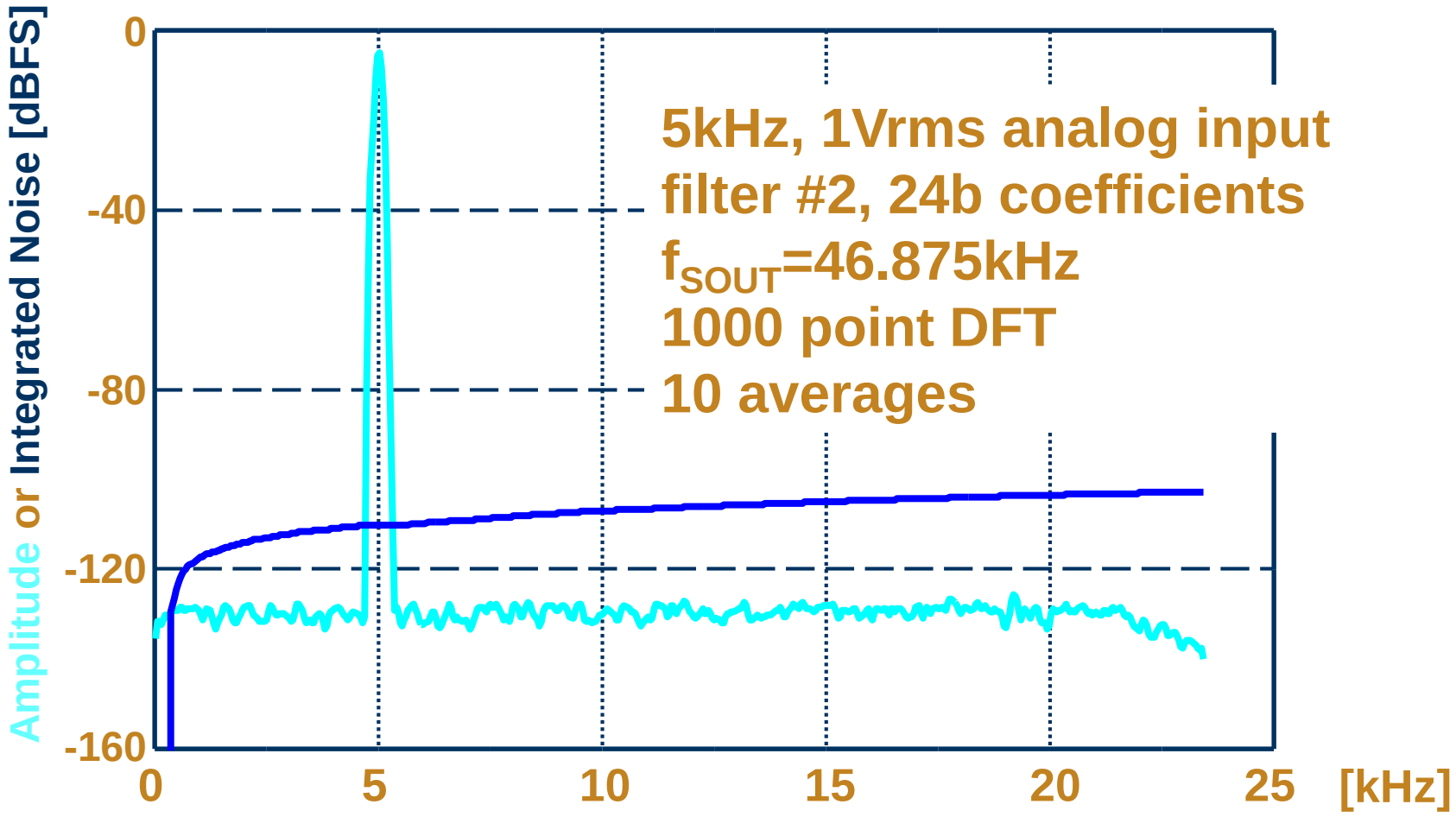
Filter #2 Bit Width

The green accumulator bits (20 and 21) provide complete overload protection:



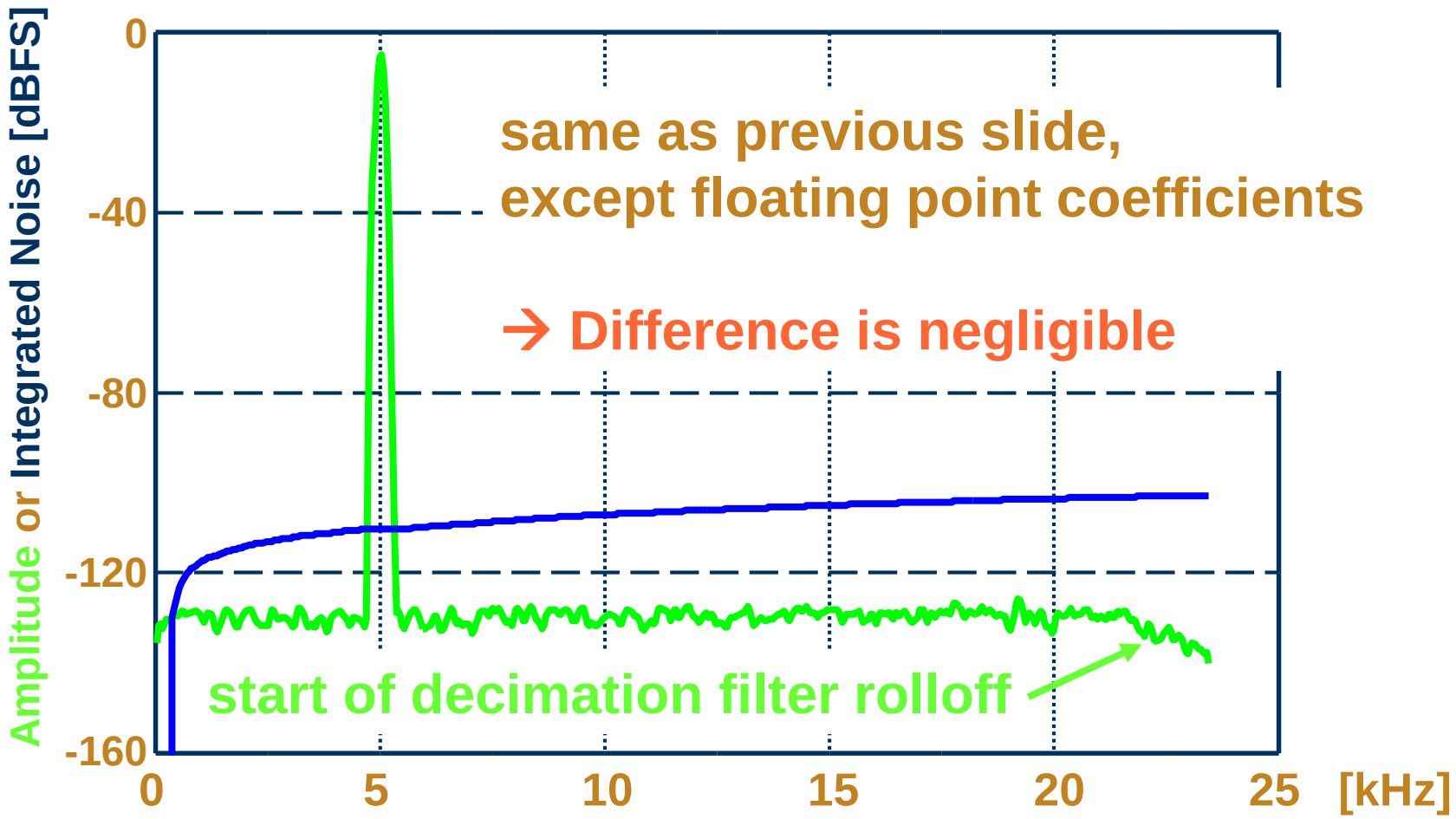
[Eric Swanson]

ΣΔ ADC Output DFT: finite precision



[Eric Swanson]

ΣΔ ADC Output DFT: floating point



[Eric Swanson]