ECE1371 Advanced Analog Circuits
Lecture 1

INTRODUCTION TO DELTA-SIGMA ADCS

Richard Schreier
richard.schreier@analog.com

Trevor Caldwell
trevor.caldwell@utoronto.ca

Course Goals

- **Deepen understanding of CMOS analog circuit design through a top-down study of a modern analog system— a delta-sigma ADC**
- **Develop circuit insight through brief peeks at some nifty little circuits**

  The circuit world is filled with many little gems that every competent designer ought to know.

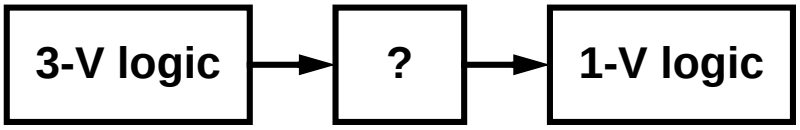
Logistics

- **Format:**
  - **Meet Mondays 1:00-3:00 PM (not Feb 16, Apr 6)**
  - **12 2-hr lectures plus proj. presentation**
- **Grading:**
  - **30% homework**
  - **40% project**
  - **30% exam**
- **References:**
  - **Schreier & Temes, "Understanding ΔΣ …"**
  - **Chan-Carusone, Johns & Martin, "Analog IC …"**
  - **Razavi, "Design of Analog CMOS ICs"**
- **Lecture Plan:**

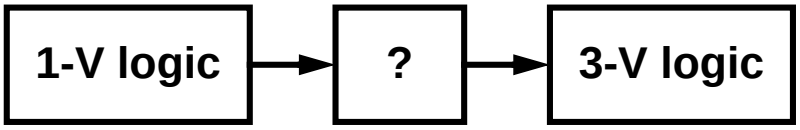
Date	Lecture (M 13:00-15:00)			Ref	Homework
2015-01-05	RS	1	MOD1 & MOD2	ST 2, 3, A	1: Matlab MOD1&2
2015-01-12	RS	2	MOD*N* + ΔΣ Toolbox	ST 4, B	2: ΔΣ Toolbox
2015-01-19	RS	3	Example Design: Part 1	ST 9.1, CCJM 14	3: Sw.-level MOD2
2015-01-26	RS	4	Example Design: Part 2	CCJM 18	
2015-02-02	TC	5	SC Circuits	R 12, CCJM 14	4: SC Integrator
2015-02-09	TC	6	Amplifier Design		
2015-02-16	Reading Week– No Lecture				
2015-02-23	TC	7	Amplifier Design		5: SC Int w/ Amp
2015-03-02	RS	8	Comparator & Flash ADC	CCJM 10	Project
2015-03-09	TC	9	Noise in SC Circuits	ST C	
2015-03-16	RS	10	Advanced ΔΣ	ST 6.6, 9.4	
2015-03-23	TC	11	Matching & MM-Shaping	ST 6.3-6.5, +	
2015-03-30	TC	12	Pipeline and SAR ADCs	CCJM 15, 17	
2015-04-06	Exam			Proj. Report Due Friday April 10	
2015-04-13	Project Presentation				

NLCOTD: Level Translator

VDD1 > VDD2, e.g.



- **VDD1 < VDD2, e.g.**



- **Constraints: CMOS**
  1-V and 3-V devices
  no static current

Highlights
(i.e. What you will learn today)

1. **MOD1: 1st-order $\Delta\Sigma$ modulator**
   Structure and theory of operation
2. **Inherent linearity of binary modulators**
3. **Inherent anti-aliasing of continuous-time modulators**
4. **MOD2: 2nd-order $\Delta\Sigma$ modulator**
5. **Good FFT practice**

0. Background
(Stuff you already know)

- **The SQNR* of an ideal *n*-bit ADC with a full-scale sine-wave input is (6.02*n* + 1.76) dB**
  - **"6 dB = 1 bit."**
- **The PSD at the output of a linear system is the product of the input's PSD and the squared magnitude of the system's frequency response**

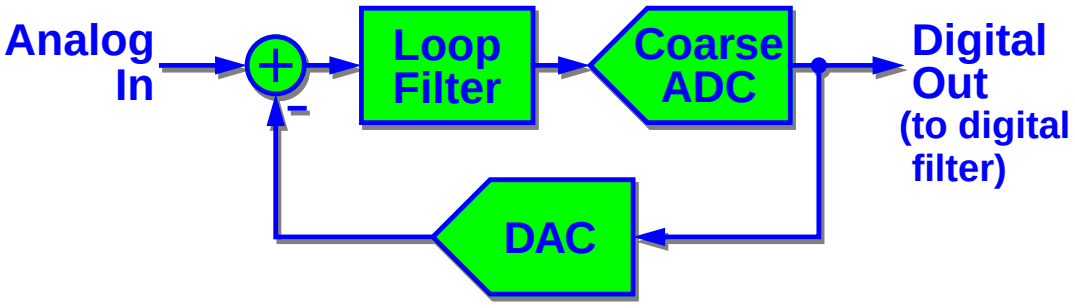
i.e. $X \rightarrow \boxed{H(z)} \rightarrow Y$ $\quad S_{yy}(f) = |H(e^{j2\pi f})|^2 \cdot S_{xx}(f)$

- **The power in any frequency band is the integral of the PSD over that band**

***. SQNR = Signal-to-Quantization-Noise Ratio**

1. What is ΔΣ?

- **ΔΣ is NOT a fraternity**
- **Simplified ΔΣ ADC structure:**



- **Key features: coarse quantization, filtering, feedback and oversampling**
  - **Quantization is often *quite* coarse (1 bit!), but the effective resolution can still be as high as 22 bits.**

What is Oversampling?

- **Oversampling is sampling faster than required by the Nyquist criterion**

  For a lowpass signal containing energy in the frequency range $(0, f_B)$, the minimum sample rate required for perfect reconstruction is $f_s = 2f_B$

- **The *oversampling ratio* is $OSR \equiv f_s/(2f_B)$**

- **For a regular ADC, $OSR \sim 2-3$**

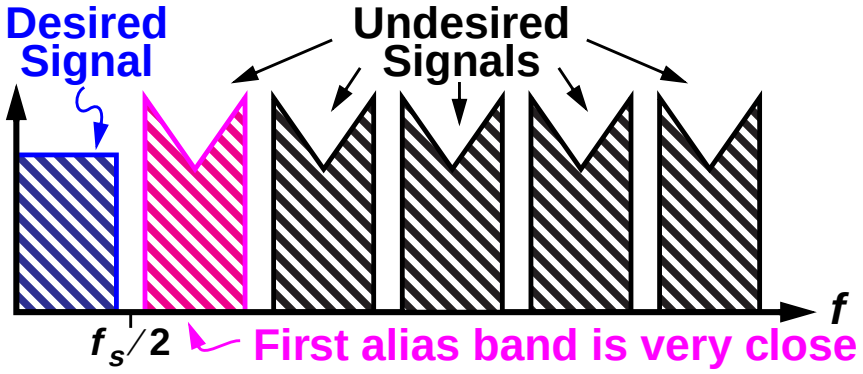
  To make the anti-alias filter (AAF) feasible

- **For a $\Delta\Sigma$ ADC, $OSR \sim 30$**

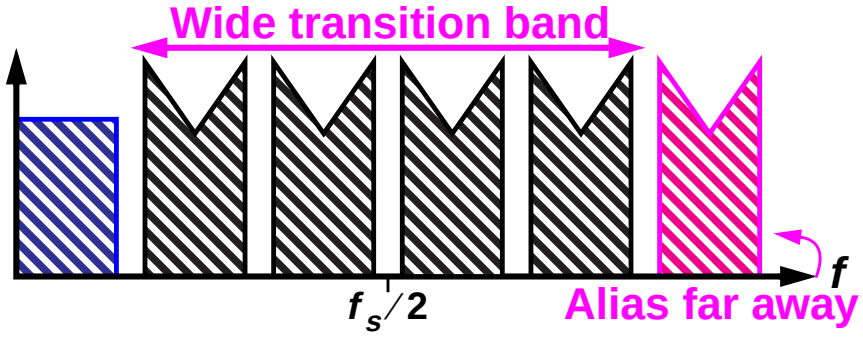
  To get adequate quantization noise suppression. Signals between f_B and ~f_s are removed digitally.

Oversampling Simplifies AAF

OSR ~ 1:

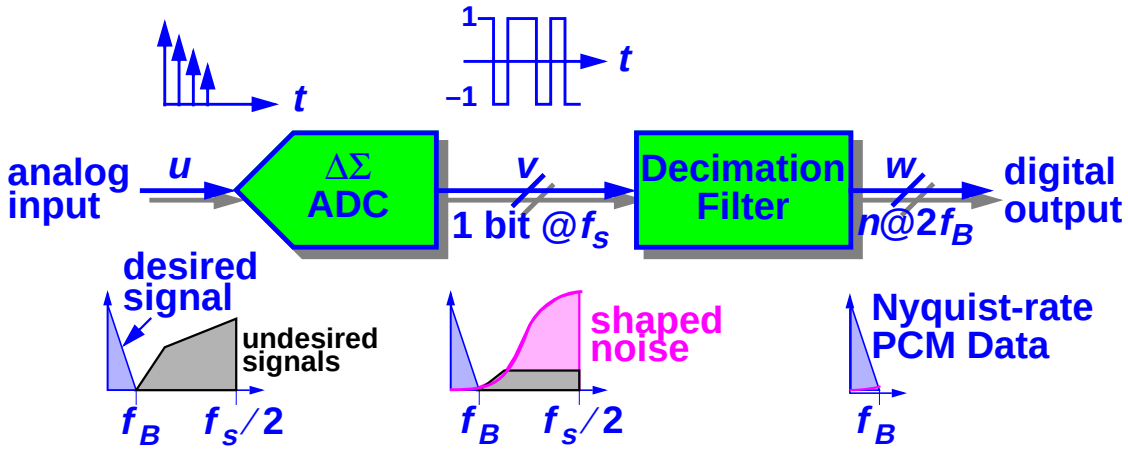


OSR = 3:

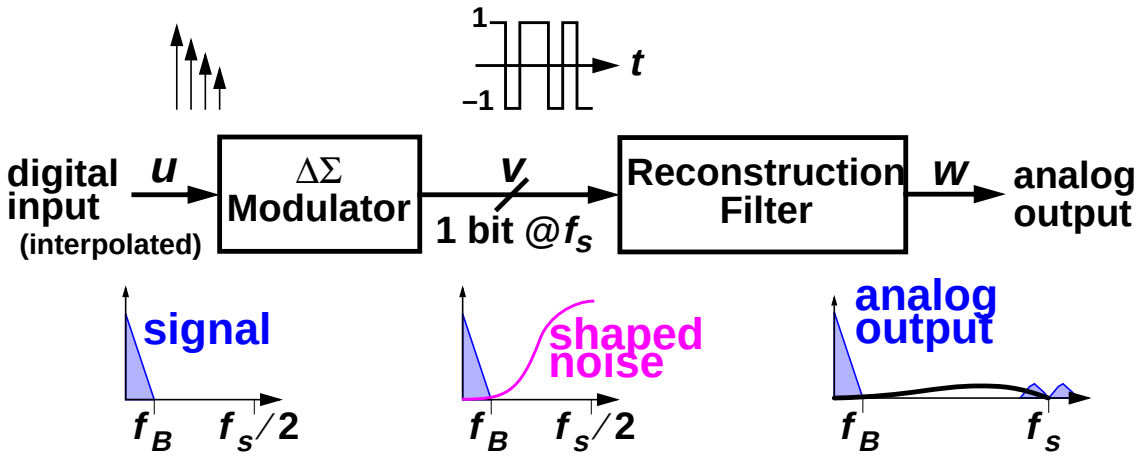


How Does A ΔΣ ADC Work?

- Coarse quantization ⇒ lots of quantization error. So how can a ΔΣ ADC achieve 22-bit resolution?
- A ΔΣ ADC spectrally separates the quantization error from the signal through *noise-shaping*



A ΔΣ DAC System



- Mathematically similar to an ADC system

  Except that now the modulator is digital and drives a low-resolution DAC, and that the out-of-band noise is handled by an analog reconstruction filter.

Why Do It The ΔΣ Way?

- **ADC: Simplified Anti-Alias Filter**

  Since the input is oversampled, only very high frequencies alias to the passband.
  A simple RC section often suffices.

  If a continuous-time loop filter is used, the anti-alias filter can often be eliminated altogether.

- **DAC: Simplified Reconstruction Filter**

  The nearby images present in Nyquist-rate reconstruction can be removed digitally.

+ **Inherent Linearity**

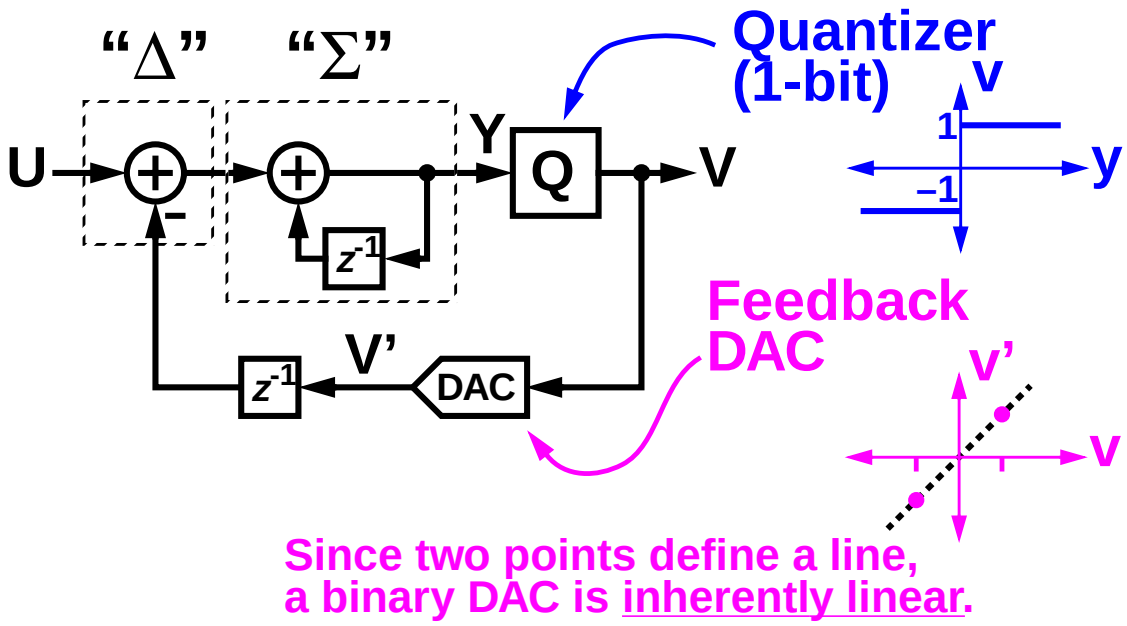
  Simple structures can yield very high SNR.

+ **Robust Implementation**

  ΔΣ tolerates sizable component errors.

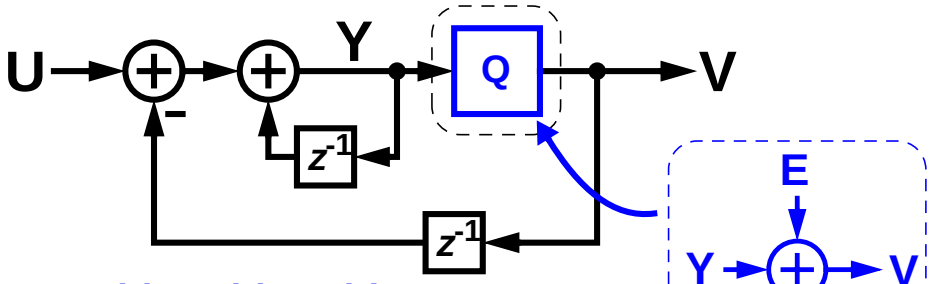
2. MOD1: 1st-Order ΔΣ Modulator

[Ch. 2 of Schreier & Temes]



MOD1 Analysis

- **Exact analysis is intractable for all but the simplest inputs, so treat the quantizer as an additive noise source:**



$V(z) = Y(z) + E(z)$

$Y(z) = (U(z) – z^{-1}V(z)) / (1–z^{-1})$

$\Rightarrow (1–z^{-1})\, V(z) = U(z) – z^{-1}V(z) + (1–z^{-1})E(z)$

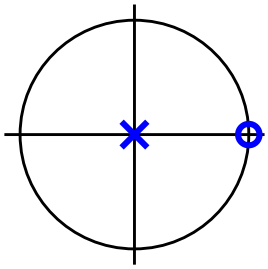
$$V(z) = U(z) + (1–z^{-1})E(z)$$

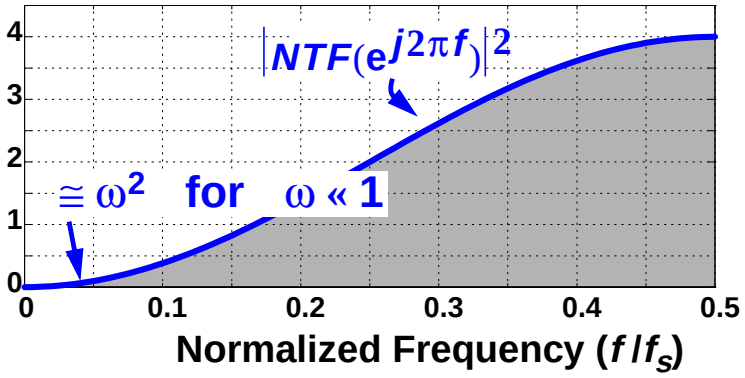
The Noise Transfer Function (NTF)

- **In general, V(z) = STF(z)•U(z) + NTF(z)•E(z)**
- **For MOD1, NTF(z) = $1–z^{-1}$**

  The quantization noise has spectral shape!

Poles & zeros:





- **The total noise power increases, but the noise power at low frequencies is reduced**

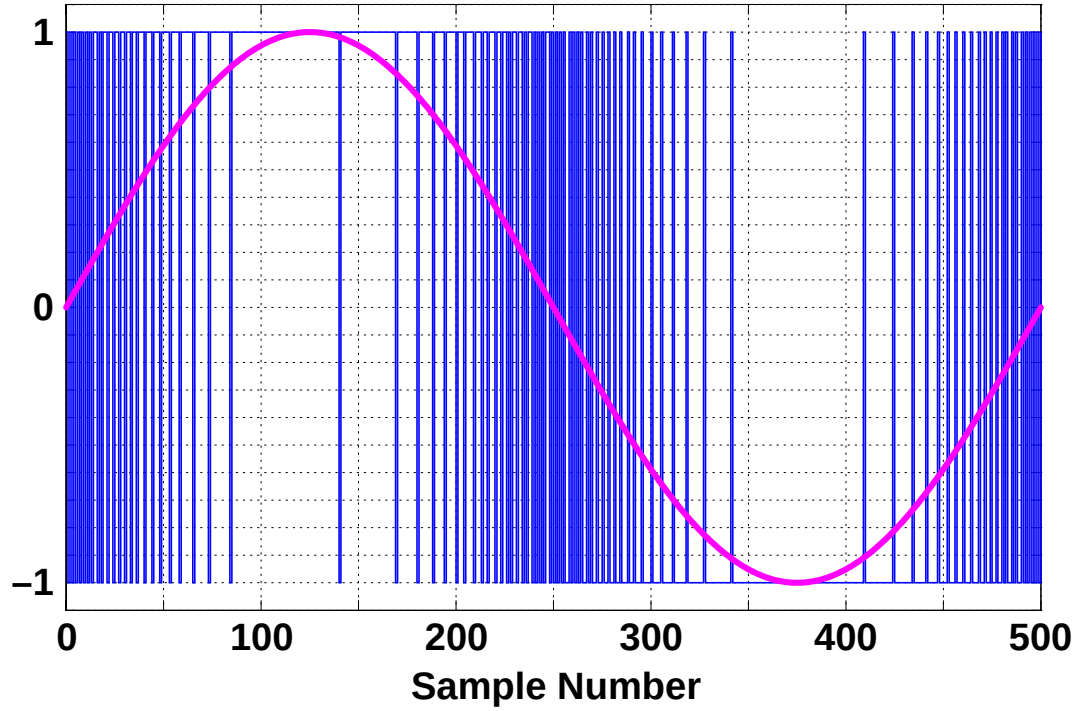
In-band Quant. Noise Power

- Assume that e is white with power σ_e^2
  i.e. $S_{ee}(\omega) = \sigma_e^2/\pi$
- The in-band quantization noise power is

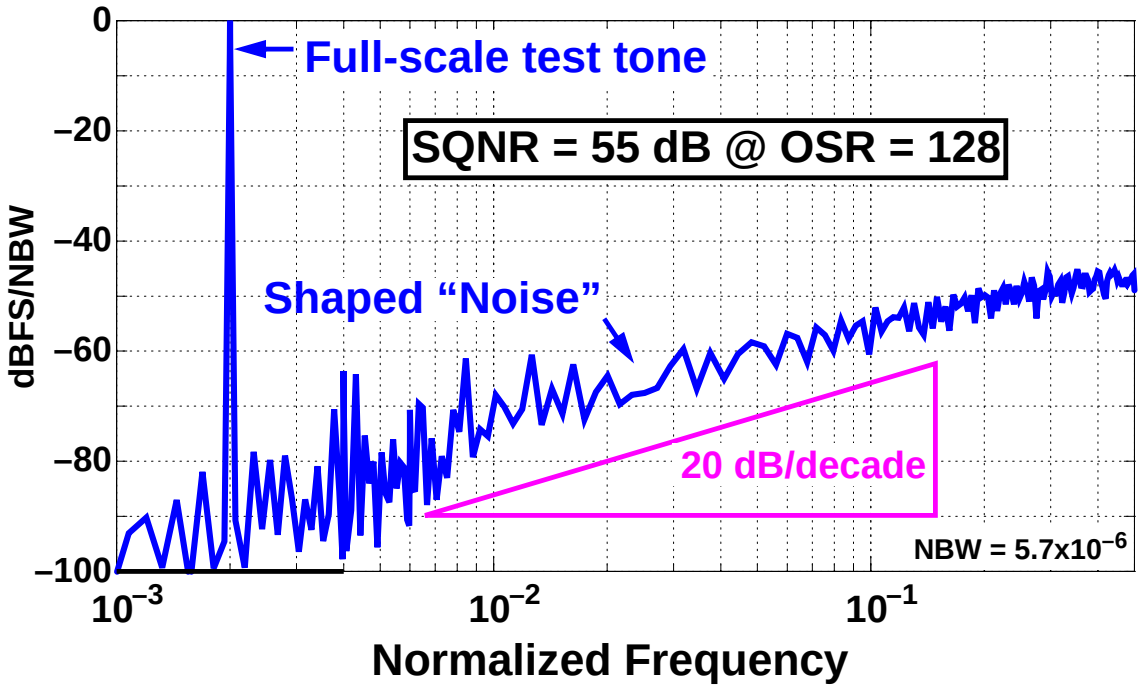
$$IQNP = \int_0^{\omega_B} |H(e^{j\omega})|^2 S_{ee}(\omega)\, d\omega \cong \frac{\sigma_e^2}{\pi} \int_0^{\omega_B} \omega^2\, d\omega$$

- Since $OSR \equiv \frac{\pi}{\omega_B}$, $IQNP = \frac{\pi^2 \sigma_e^2}{3}(OSR)^{-3}$
- For MOD1, an octave increase in *OSR* increases SQNR by 9 dB
  "1.5-bit/octave SQNR-OSR trade-off."

A Simulation of MOD1— Time



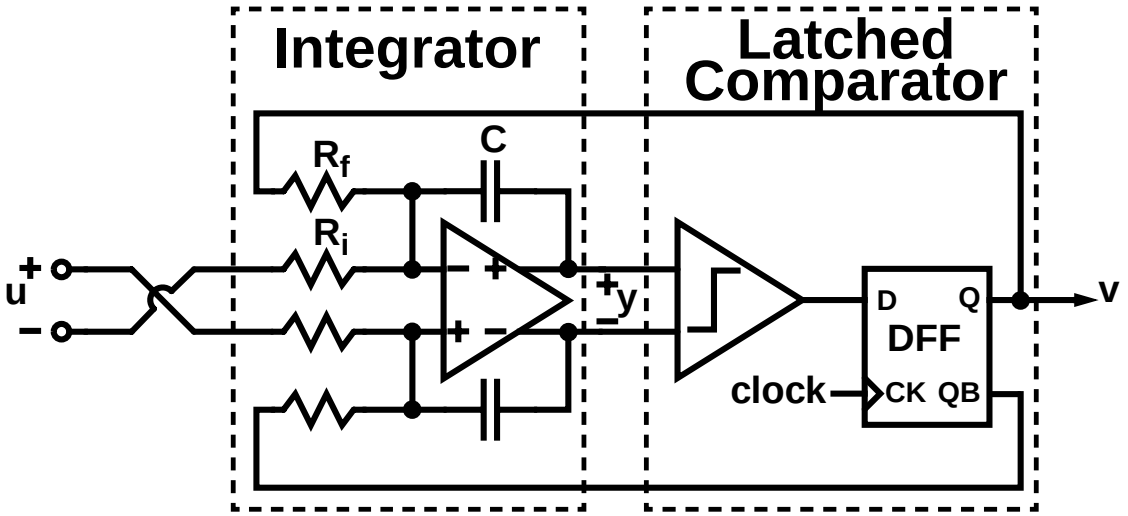
A Simulation of MOD1— Freq.



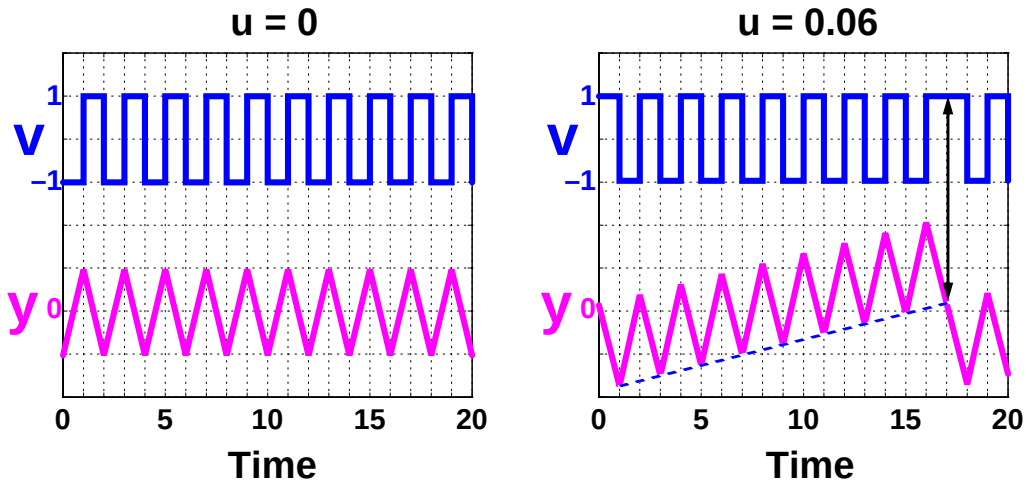
CT Implementation of MOD1

- R_i/R_f sets the full-scale; C is arbitrary

  Also observe that an input at f_s is rejected by the integrator— *inherent anti-aliasing*



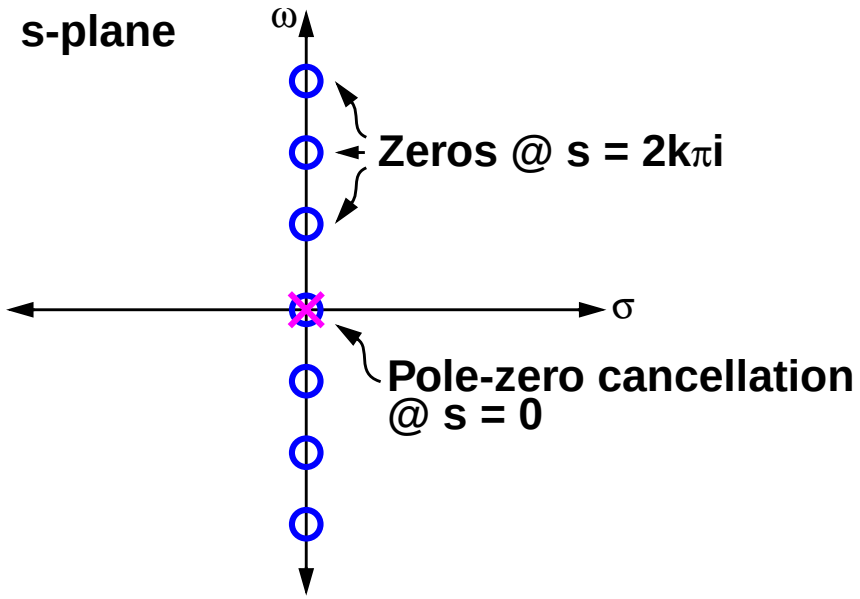
MOD1-CT Waveforms



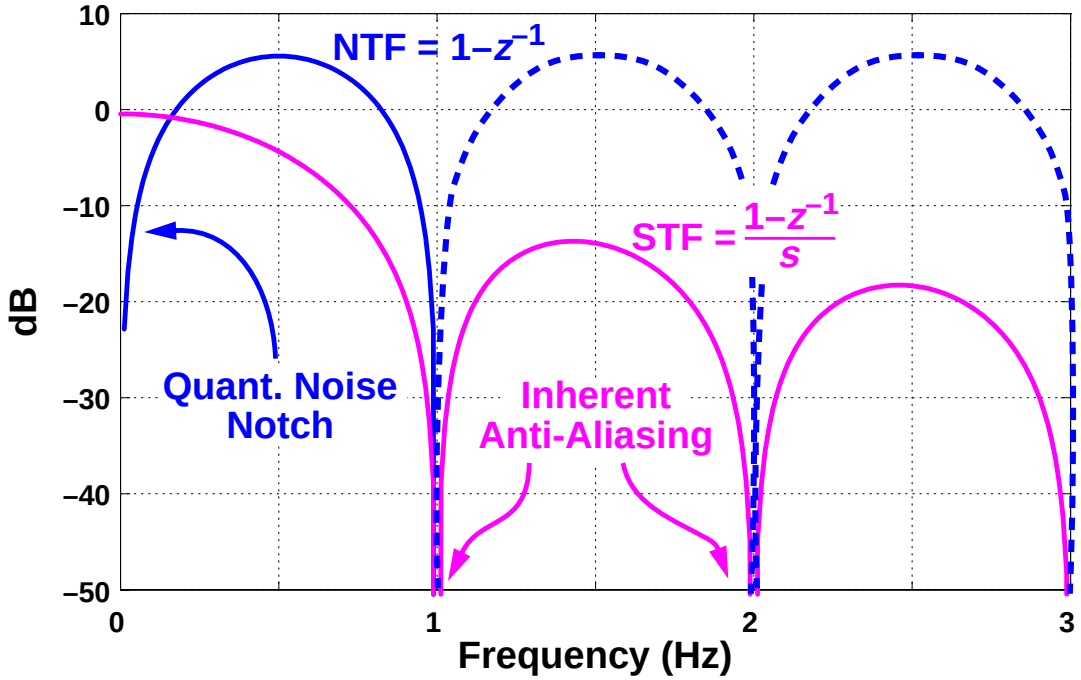
- With *u*=0, *v* alternates between +1 and –1
- With *u*>0, *y* drifts upwards; *v* contains consecutive +1s to counteract this drift

MOD1-CT STF = $\frac{1 - z^{-1}}{s}$

Recall $z = e^{s}$



MOD1-CT Frequency Responses

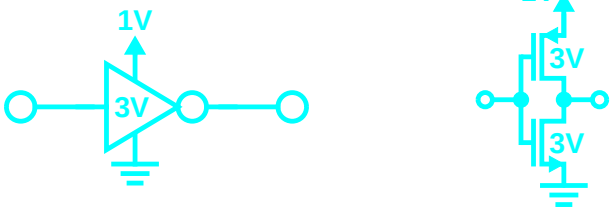


Summary

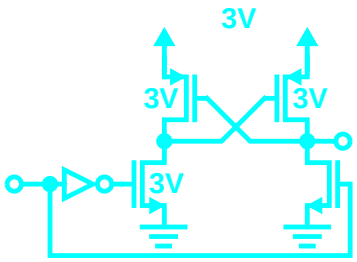
- $\Delta\Sigma$ works by spectrally separating the quantization noise from the signal
  Requires oversampling. $OSR \equiv f_s/(2f_B)$.
- Noise-shaping is achieved by the use of *filtering* and *feedback*
- A binary DAC is *inherently linear,* and thus a binary $\Delta\Sigma$ modulator is too
- MOD1 has $NTF(z) = 1 - z^{-1}$
  $\Rightarrow$ Arbitrary accuracy for DC inputs.
  1.5 bit/octave SQNR-OSR trade-off.
- MOD1-CT has *inherent anti-aliasing*

NLCOTD

3V → 1V:



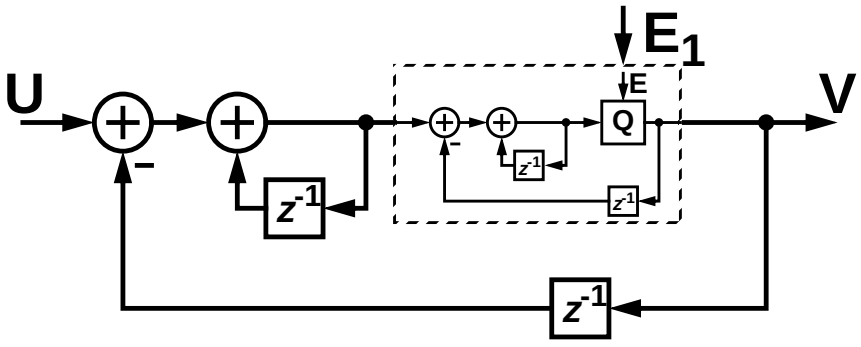
1V → 3V:



3. MOD2: 2nd-Order ΔΣ Modulator

[Ch. 3 of Schreier & Temes]

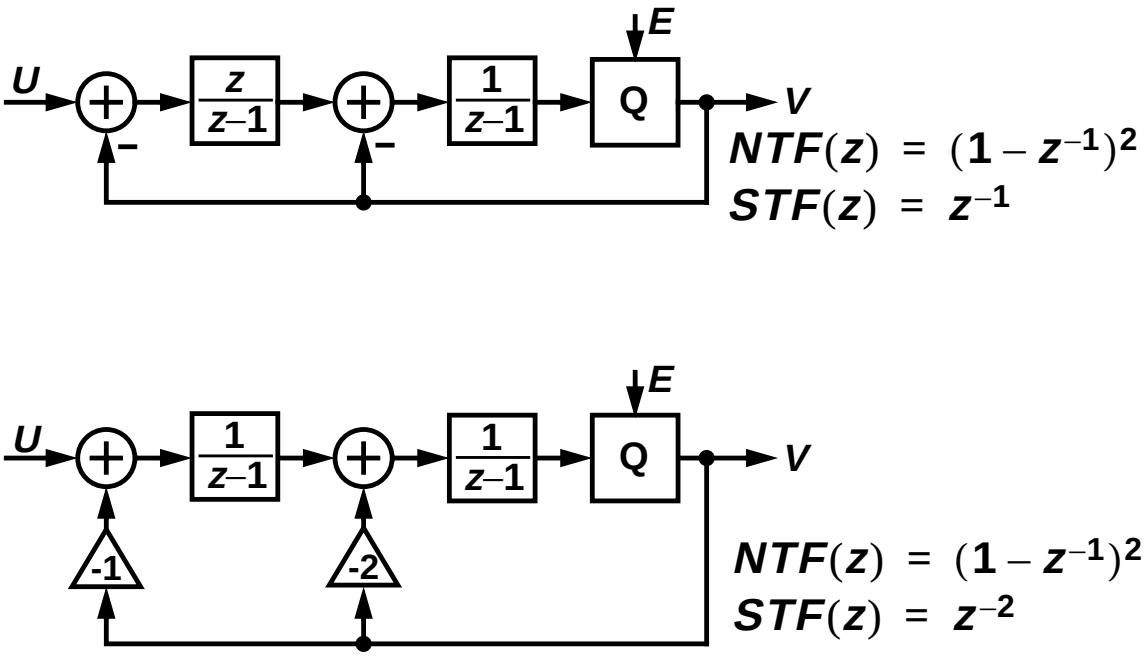
- **Replace the quantizer in MOD1 with another copy of MOD1 in a recursive fashion:**



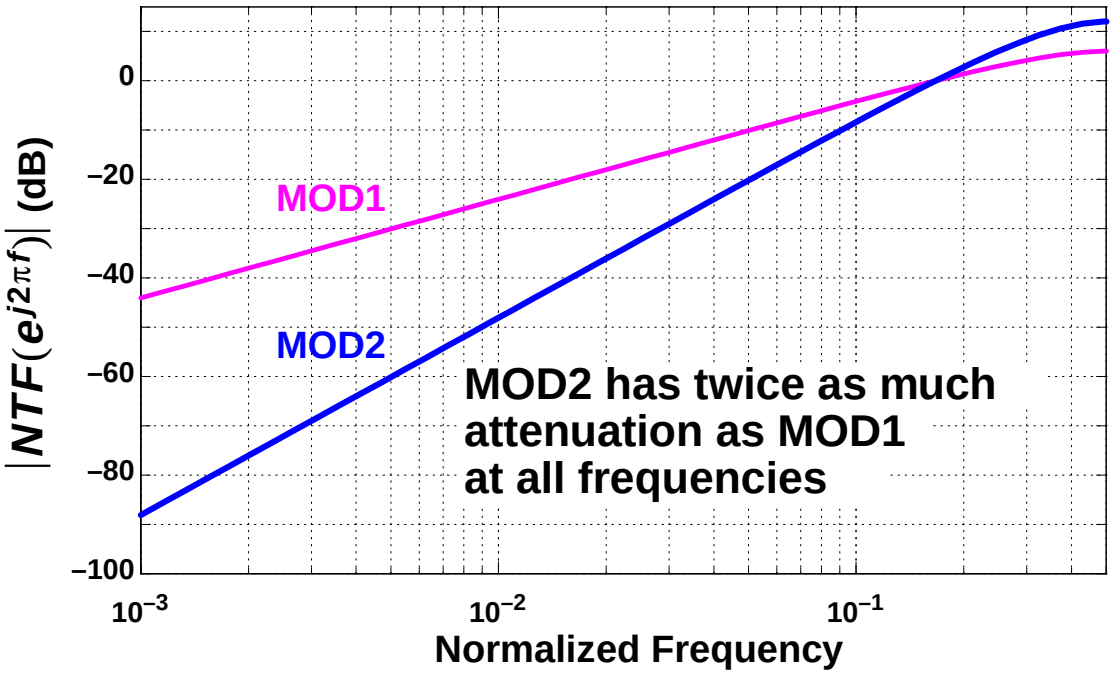
$$V(z) = U(z) + (1-z^{-1})E_1(z),\ E_1(z) = (1-z^{-1})E(z)$$

$$\Rightarrow V(z) = U(z) + (1-z^{-1})^2 E(z)$$

Simplified Block Diagrams



NTF Comparison



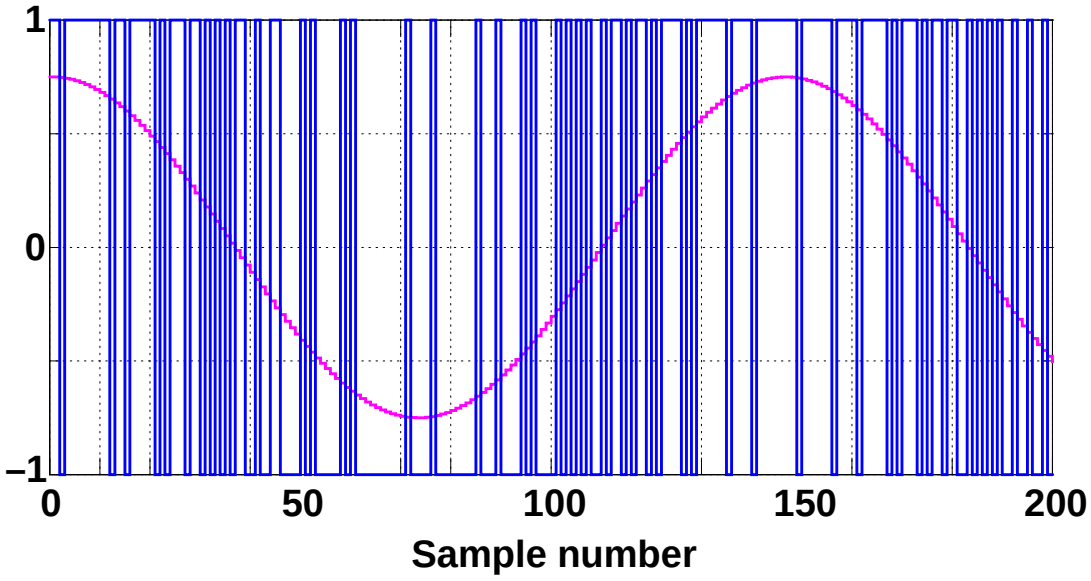
In-band Quant. Noise Power

- For MOD2, $|H(e^{j\omega})|^2 \approx \omega^4$
- As before, $IQNP = \int_0^{\omega_B} |H(e^{j\omega})|^2 S_{ee}(\omega)\, d\omega$ and $S_{ee}(\omega) = \sigma_e^2/\pi$
- So now $IQNP = \frac{\pi^4 \sigma_e^2}{5}(OSR)^{-5}$ ✳

  With binary quantization to ±1,
  $\Delta = 2$ and thus $\sigma_e^2 = \Delta^2/12 = 1/3$.
- "An octave increase in *OSR* increases MOD2's SQNR by 15 dB (2.5 bits)"

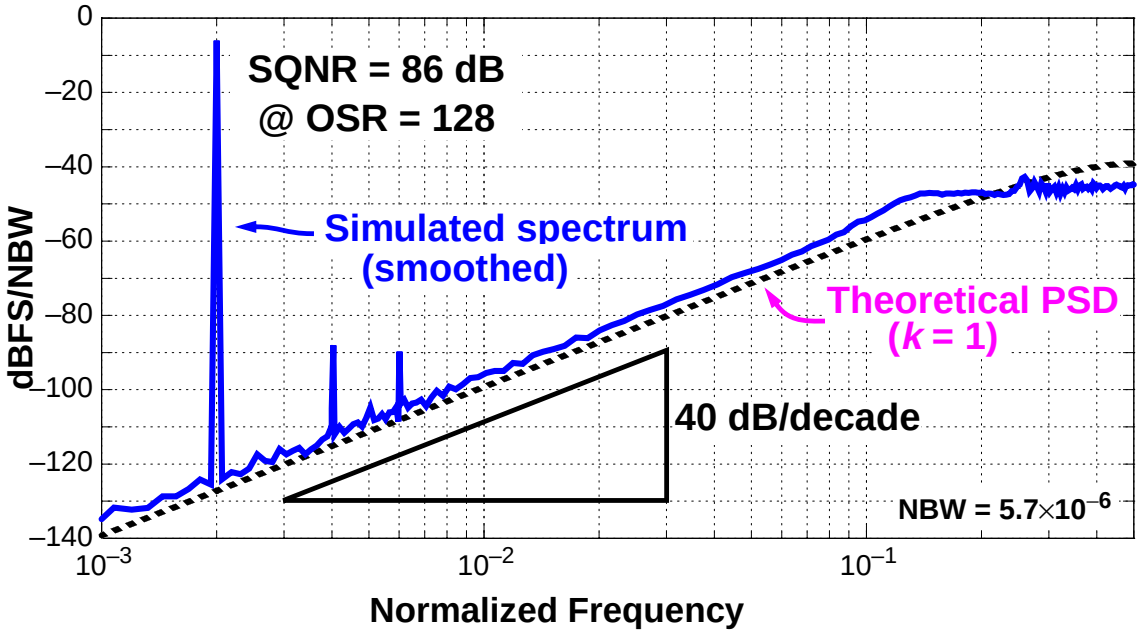
Simulation Example

Input at 75% of FullScale



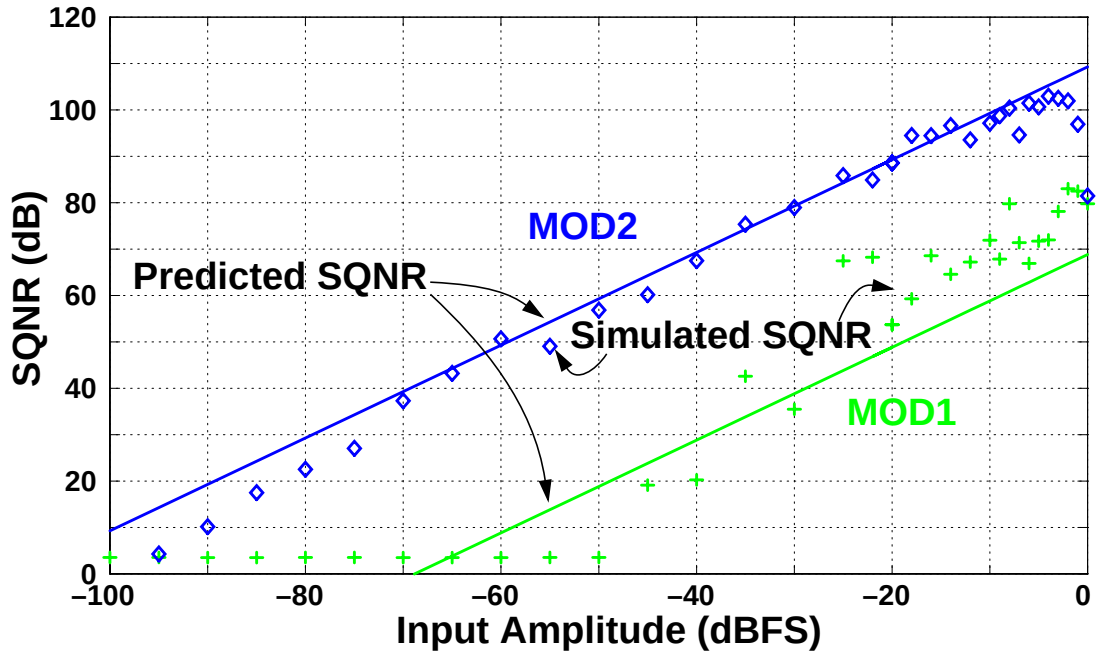
Simulated MOD2 PSD

Input at 50% of FullScale

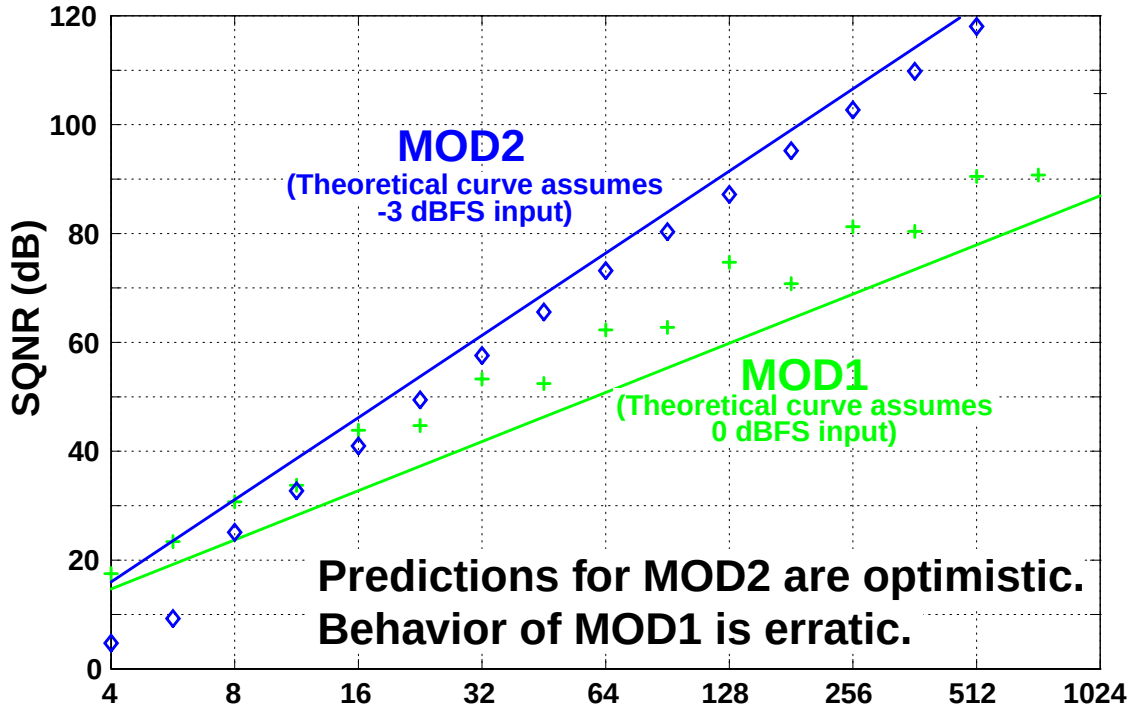


SQNR vs. Input Amplitude

MOD1 & MOD2 @ OSR = 256

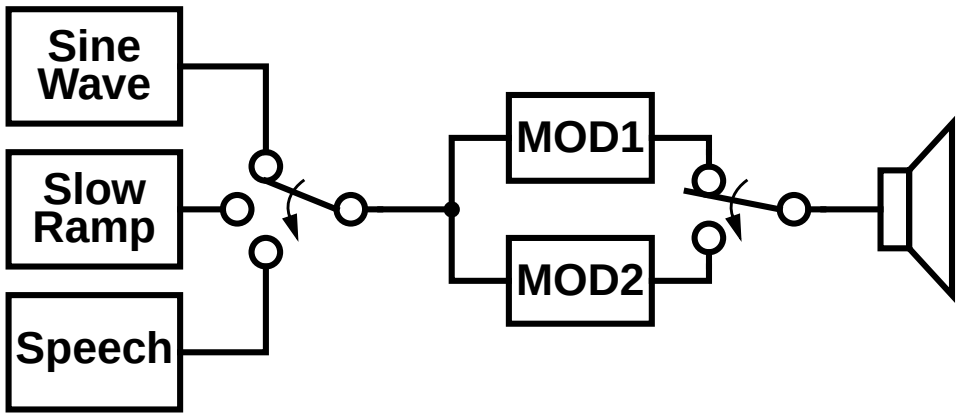


SQNR vs. OSR



Audio Demo: MOD1 vs. MOD2

[dsdemo4]

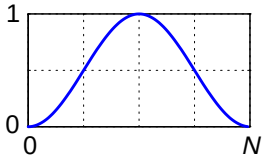


MOD1 + MOD2 Summary

- **$\Delta\Sigma$ ADCs rely on filtering and feedback to achieve high SNR despite coarse quantization**
  - **They also rely on digital signal processing.**
  - **$\Delta\Sigma$ ADCs need to be followed by a digital decimation filter and $\Delta\Sigma$ DACs need to be preceded by a digital interpolation filter.**
- **Oversampling eases analog filtering requirements**
  - **Anti-alias filter in an ADC; image filter in a DAC.**
- **Binary quantization yields inherent linearity**
- **MOD2 is better than MOD1**
  - **15 dB/octave vs. 9 dB/octave SQNR-OSR trade-off.**
  - **Quantization noise more white.**
  - **Higher-order modulators are even better.**

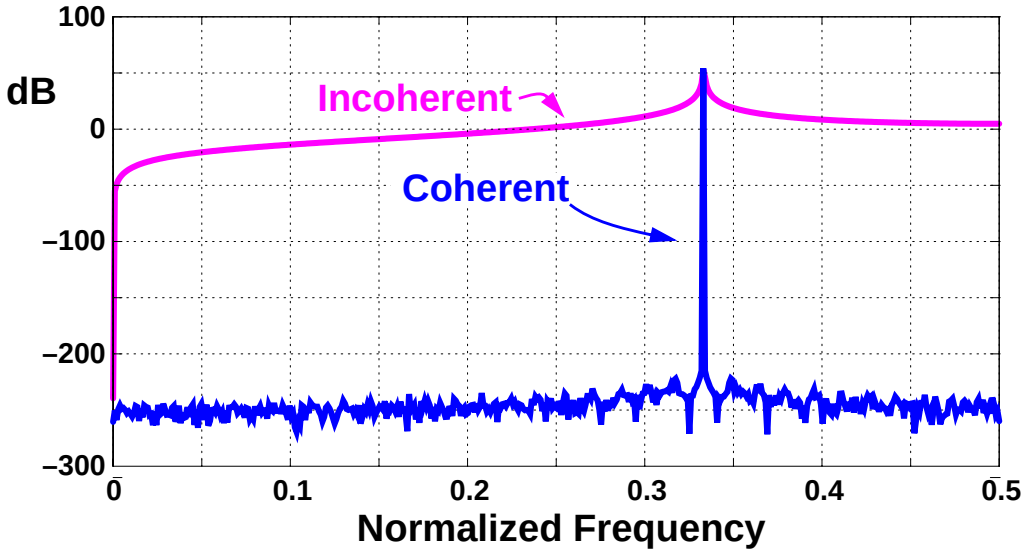
4. Good FFT Practice
[Appendix A of Schreier & Temes]

- **Use coherent sampling**
  - **I.e. have an integer number of cycles in the record.**
- **Use windowing**
  - **A Hann window $w(n) = (1 - \cos(2\pi n/N))/2$ works well.**



- **Use enough points**
  - **Recommend $N = 64 \cdot OSR$.**
- **Scale (and smooth) the spectrum**
  - **A full-scale sine wave should yield a 0-dBFS peak.**
- **State the noise bandwidth**
  - **For a Hann window, $NBW = 1.5/N$.**

Coherent vs. Incoherent Sampling

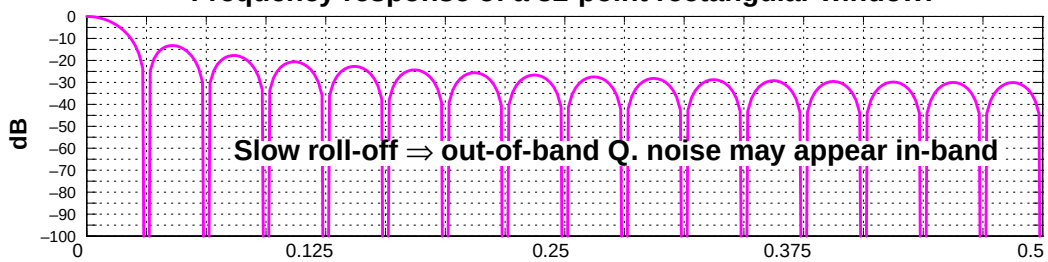


- **Coherent sampling: only one non-zero FFT bin**
- **Incoherent sampling: "spectral leakage"**

Windowing

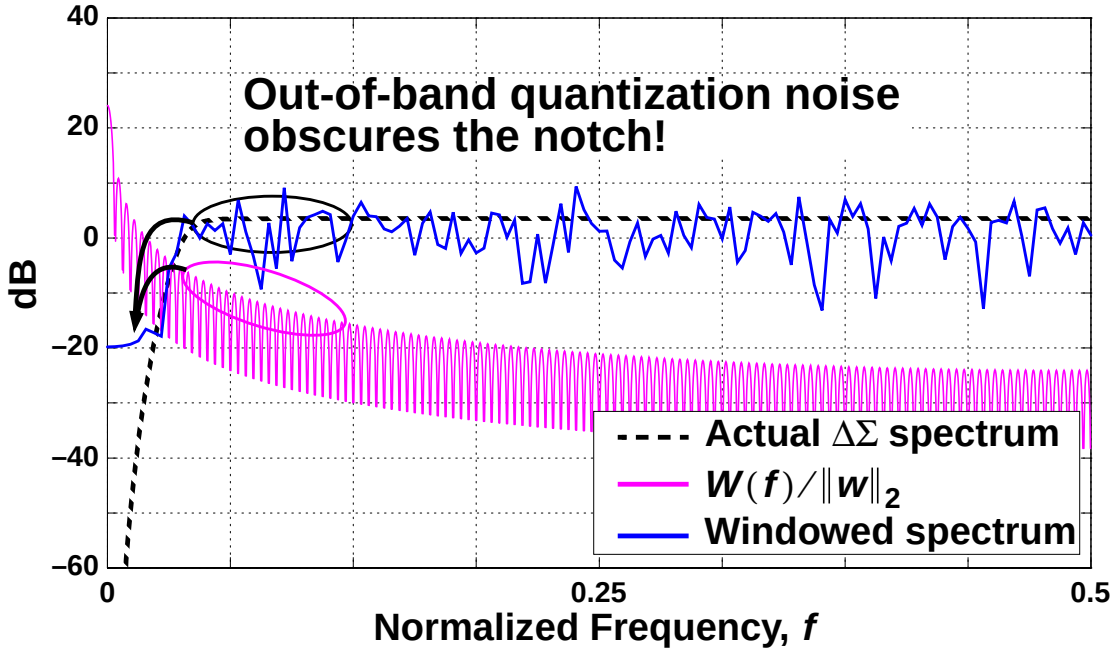
- **$\Delta\Sigma$ data is usually not periodic**
  Just because the input repeats does not mean that the output does too!
- **A finite-length data record = an infinite record multiplied by a *rectangular window*:**
  $w(n) = 1, 0 \le n < N$
  Windowing is unavoidable.
- **"Multiplication in time is convolution in frequency"**

Frequency response of a 32-point rectangular window:

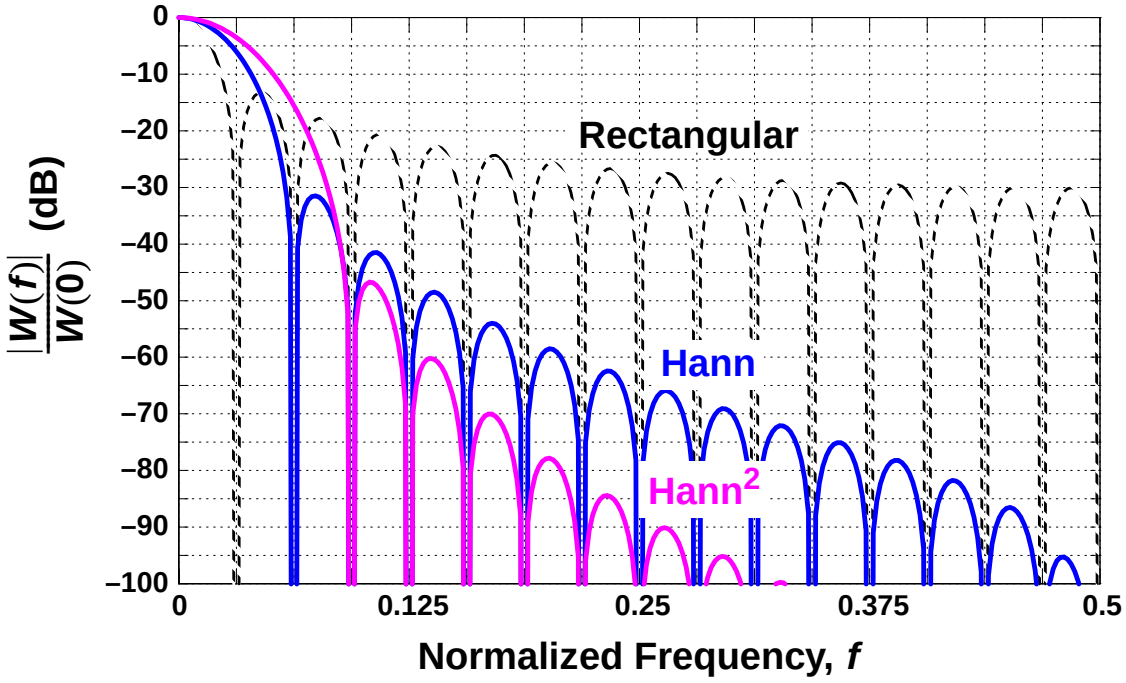


Example Spectral Disaster

Rectangular window, $N = 256$



Window Comparison ($N = 16$)



Window Properties

Window	Rectangular	Hann†	Hann^2
$w(n)$, $n = 0, 1, \ldots, N-1$ ($w(n) = 0$ otherwise)	1	$\frac{1 - \cos\frac{2\pi n}{N}}{2}$	$\left(\frac{1 - \cos\frac{2\pi n}{N}}{2}\right)^2$
Number of non-zero FFT bins	1	3	5
$\|w\|_2^2 = \sum w(n)^2$	N	$3N/8$	$35N/128$
$W(0) = \sum w(n)$	N	$N/2$	$3N/8$
$NBW = \frac{\|w\|_2^2}{W(0)^2}$	$1/N$	$1.5/N$	$35/18N$

†. MATLAB's "hann" function causes spectral leakage of tones located in FFT bins unless you add the optional argument "periodic."

Window Length, N

- Need to have enough in-band noise bins to

1 Make the number of signal bins a small fraction of the total number of in-band bins

<20% signal bins $\Rightarrow$ >15 in-band bins $\Rightarrow$ $N > 30 \cdot OSR$

2 Make the SNR repeatable

$N = 30 \cdot OSR$ yields std. dev. ~1.4 dB.

$N = 64 \cdot OSR$ yields std. dev. ~1.0 dB.

$N = 256 \cdot OSR$ yields std. dev. ~0.5 dB.

- $N = 64 \cdot OSR$ is recommended

FFT Scaling

- **The FFT implemented in MATLAB is**

$$X_M(k+1) = \sum_{n=0}^{N-1} x_M(n+1)e^{-j\frac{2\pi kn}{N}}$$

- **If $x(n) = A\sin(2\pi fn/N)$†, then**

$$|X(k)| = \begin{cases} \frac{AN}{2} & , k = f \text{ or } N-f \\ 0 & , \text{otherwise} \end{cases}$$

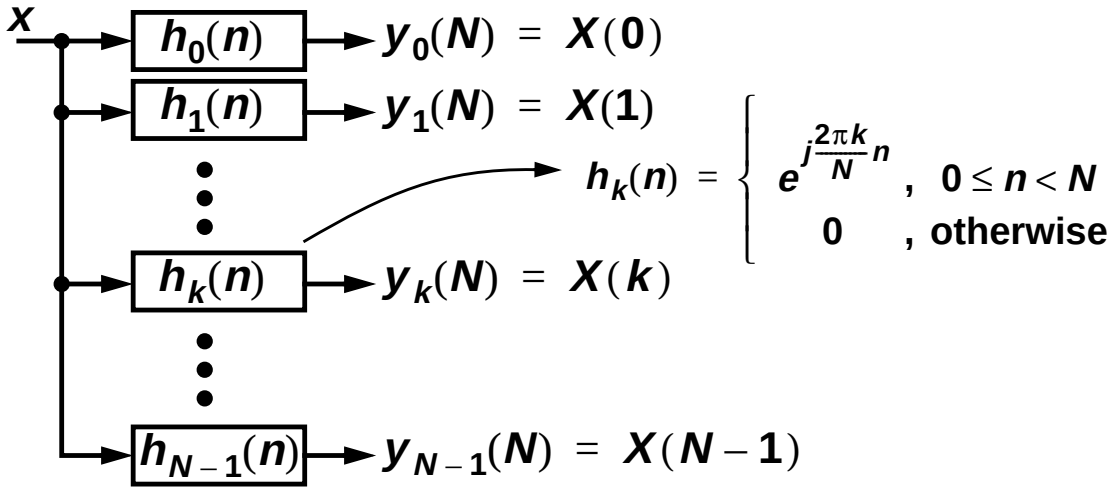
$\Rightarrow$ **Need to divide FFT by $(N/2)$ to get A.**

---

†. **f is an integer in $(0, N/2)$. I've defined $X(k) \equiv X_M(k+1)$, $x(n) \equiv x_M(n+1)$ since Matlab indexes from 1 rather than 0.**

The Need For Smoothing

- **The FFT can be interpreted as taking 1 sample from the outputs of N complex FIR filters:**

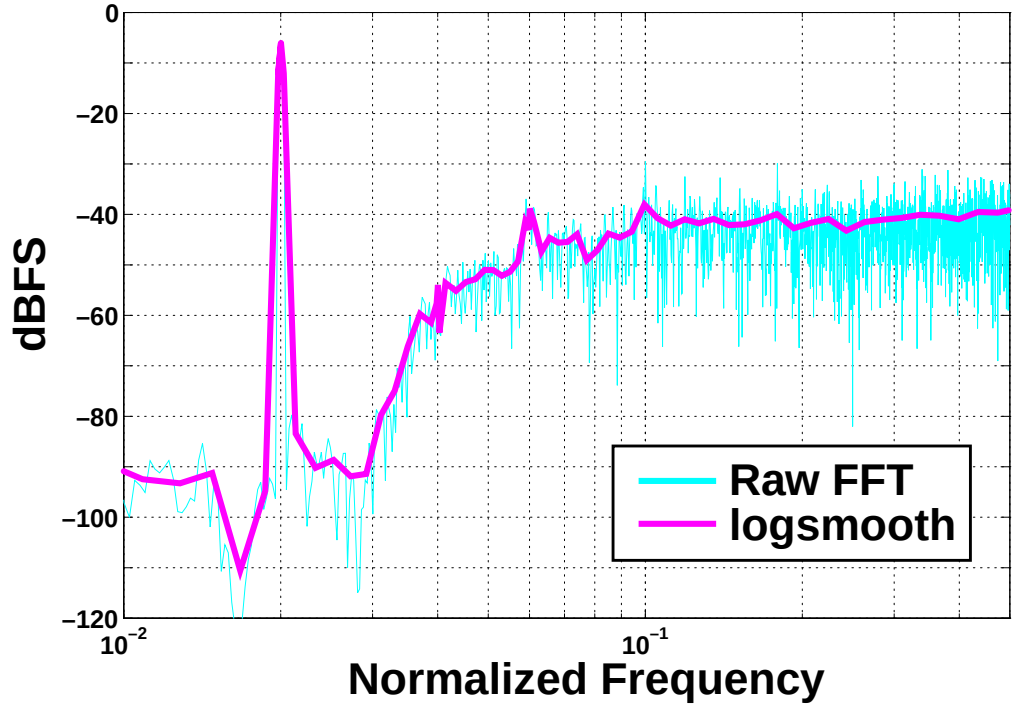


$\Rightarrow$ **an FFT yields a high-variance spectral estimate**

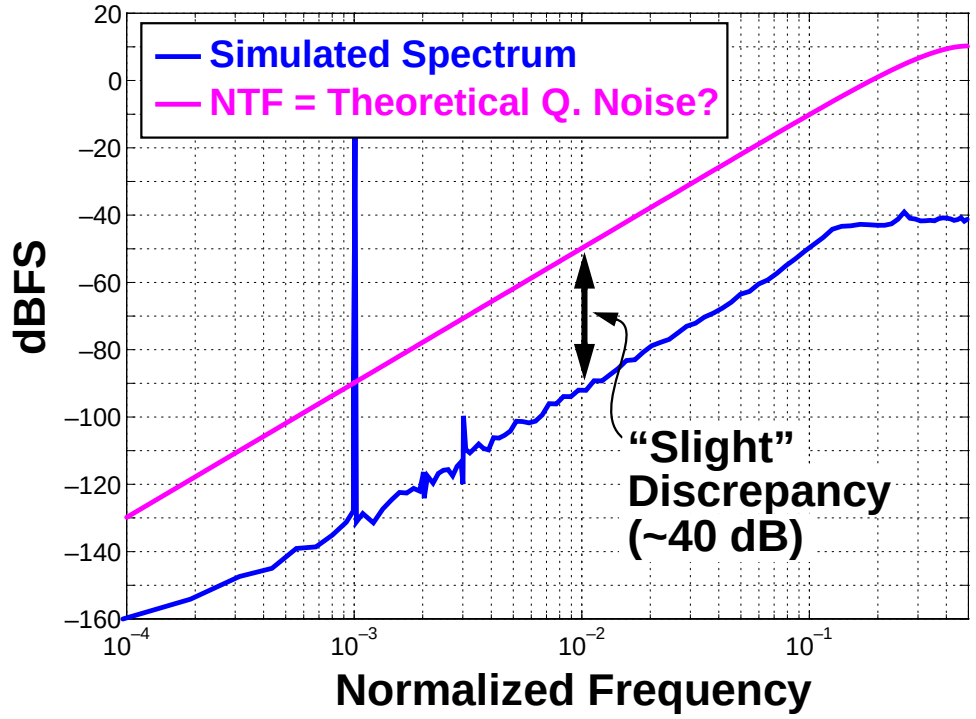
How To Do Smoothing

1. **Average multiple FFTs**
   - **Implemented by MATLAB's `psd()` function**
2. **Take one big FFT and "filter" the spectrum**
   - **Implemented by the ΔΣ Toolbox's `logsmooth()` function**
   - **`logsmooth()` averages an exponentially-increasing number of bins in order to reduce the density of points in the high-frequency regime and make a nice log-frequency plot**

Raw and Smoothed Spectra



Simulation vs. Theory (MOD2)



What Went Wrong?

1 **We normalized the spectrum so that a full-scale sine wave (which has a power of 0.5) comes out at 0 dB (whence the "dBFS" units)**

⇒ **We need to do the same for the error signal.**

i.e. use $S_{ee}(f) = 4/3$.

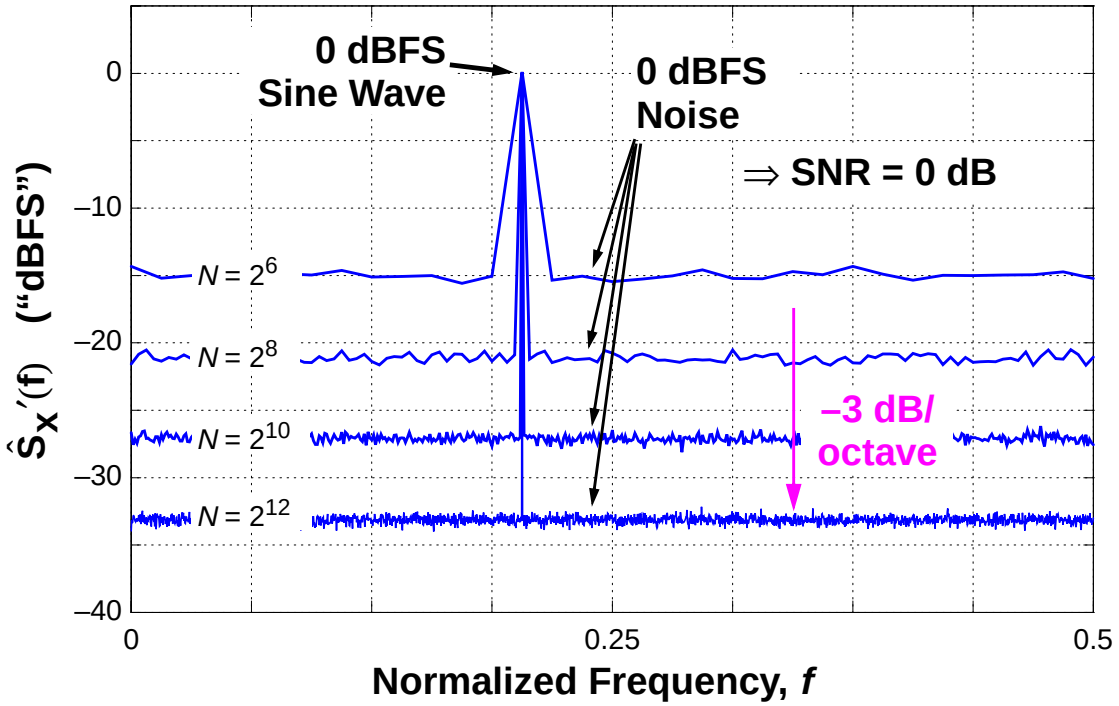
But this makes the discrepancy 3 dB worse.

2 **We tried to plot a *power spectral density* together with something that we want to interpret as a *power spectrum***

- **Sine-wave components are located in individual FFT bins, but broadband signals like noise have their power spread over all FFT bins!**

The "noise floor" depends on the length of the FFT.

Spectrum of a Sine Wave + Noise



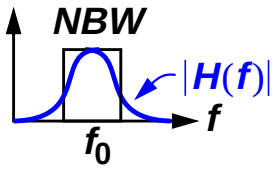
Observations

- **The power of the sine wave is given by the height of its spectral peak**
- **The power of the noise is spread over all bins**
  - **The greater the number of bins, the less power there is in any one bin.**
- **Doubling *N* reduces the power per bin by a factor of 2 (i.e. 3 dB)**
  - **But the total integrated noise power does *not* change.**

So How Do We Handle Noise?

- Recall that an FFT is like a filter bank
- The longer the FFT, the narrower the bandwidth of each filter and thus the lower the power at each output
- We need to know the *noise bandwidth* (NBW) of the filters in order to convert the power in each bin (filter output) to a power density
- For a filter with frequency response $H(f)$,

$$NBW = \frac{\int |H(f)|^2 df}{H(f_0)^2}$$



FFT Noise Bandwidth

Rectangular Window

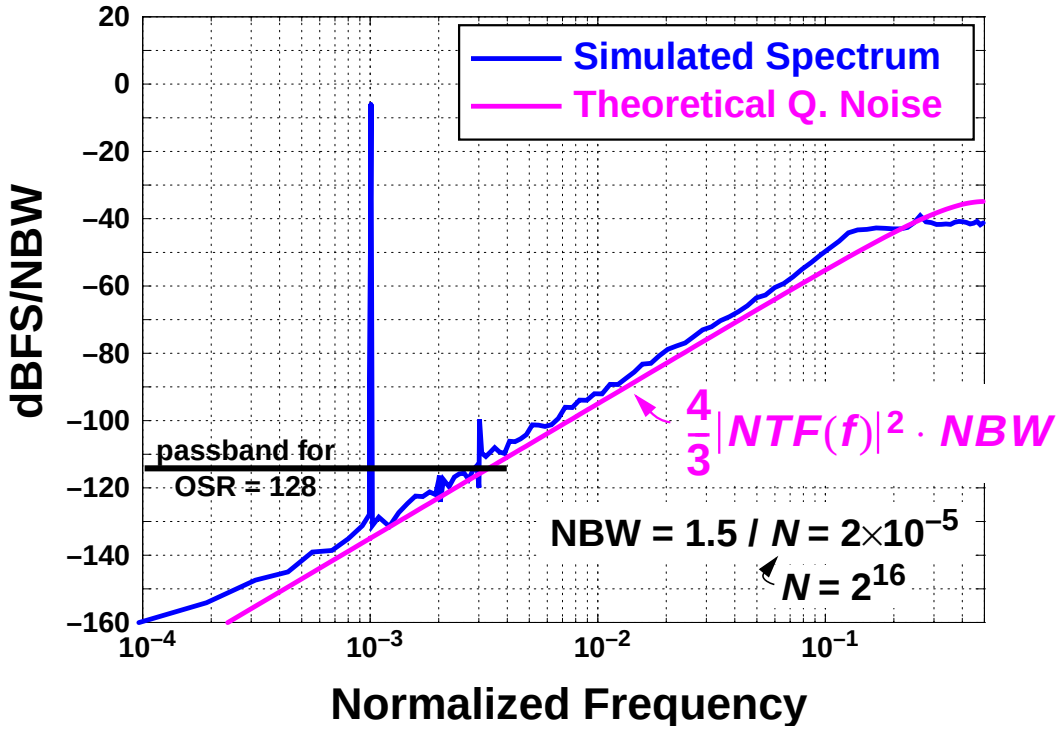
$$h_k(n) = \exp\left(j\frac{2\pi k}{N}n\right),\ H_k(f) = \sum_{n=0}^{N-1} h_k(n)\exp(-j2\pi fn)$$

$$f_0 = \frac{k}{N},\ H_k(f_0) = \sum_{n=0}^{N-1} 1 = N$$

$$\int |H_k(f)|^2 = \sum |h_k(n)|^2 = N \text{ [Parseval]}$$

$$\therefore NBW = \frac{\int |H_k(f)|^2 df}{H_k(f_0)^2} = \frac{N}{N^2} = \frac{1}{N}$$

Better Spectral Plot



Homework #1 (Due 2015-01-12)

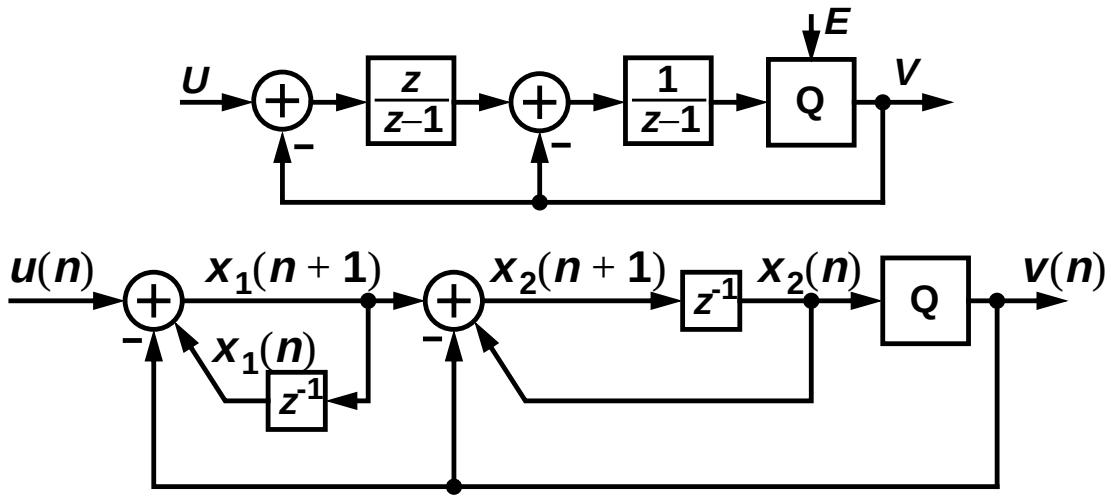
A. Create a Matlab function that computes MOD1's output sequence given a vector of input samples and exercise your function in the following ways.

1. **Verify that the average of the output equals the input for a few random DC inputs in [–1,1].**
2. **Plot the output spectrum with a half-scale sine-wave input. Use good FFT practice. Include the theoretical quantization noise curve and list the theoretical and simulated SQNR for OSR = 128.**

B. Repeat with MOD2.

C. Compose your own question and answer it.

MOD2 Expanded



Difference Equations:

$$v(n) = Q(x_2(n))$$
$$x_1(n+1) = x_1(n) - v(n) + u(n)$$
$$x_2(n+1) = x_2(n) - v(n) + x_1(n+1)$$

Example Matlab Code

```
function [v] = simulateMOD2(u)
    x1 = 0;
    x2 = 0;
    for i = 1:length(u)
        v(i) = quantize( x2 );
        x1 = x1 + u(i) - v(i);
        x2 = x2 + x1 - v(i);
    end
return

function v = quantize( y )
    if y>=0
        v = 1;
    else
        v = -1;
    end
return
```

What You Learned Today

And what the homework should solidify

1. **MOD1: 1st-order $\Delta\Sigma$ modulator**
   - **Structure and theory of operation**
2. **Inherent linearity of binary modulators**
3. **Inherent anti-aliasing of continuous-time modulators**
4. **MOD2: 2nd-order $\Delta\Sigma$ modulator**
5. **Good FFT practice**