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**

  The lectures will focus on Delta-Sigma ADCs, but you may do your project on another analog system.

- **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 recognize.

Logistics

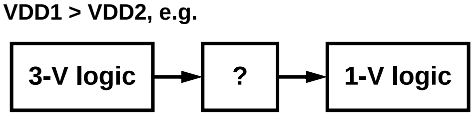
- **Format:**
  Meet Mondays 3:00-5:00 PM
  except Feb 4 and Feb 18
  12 2-hr lectures
  plus proj. presentation
- **Grading:**
  40% homework
  60% project
- **References:**
  Schreier & Temes, "Understanding ΔΣ …"
  Johns & Martin, "Analog IC Design"
  Razavi, "Design of Analog CMOS ICs"

Lecture Plan:

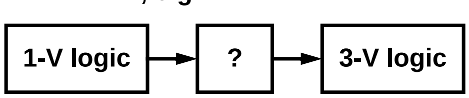
Date	Lecture			Ref	Homework
2008-01-07	RS	1	Introduction: MOD1 & MOD2	S&T 2-3, A	Matlab MOD2
2008-01-14	RS	2	Example Design: Part 1	S&T 9.1, J&M 10	Switch-level sim
2008-01-21	RS	3	Example Design: Part 2	J&M 14	Q-level sim
2008-01-28	TC	4	Pipeline and SAR ADCs		Arch. Comp.
2008-02-04	ISSCC– No Lecture				
2008-02-11	RS	5	Advanced ΔΣ	S&T 4, 6.6, 9.4, B	CTMOD2; Proj.
2008-02-18	Reading Week– No Lecture				
2008-02-25	RS	6	Comparator & Flash ADC	J&M 7	
2008-03-03	TC	7	SC Circuits	J&M 10	
2008-03-10	TC	8	Amplifier Design		
2008-03-17	TC	9	Amplifier Design		
2008-03-24	TC	10	Noise in SC Circuits	S&T C	
2008-03-31	Project Presentation				
2008-04-07	TC	11	Matching & MM-Shaping		Project Report
2008-04-14	RS	12	Switching Regulator		Q-level sim

NLCOTD: Level Translator

VDD1 > VDD2, e.g.



- VDD1 < VDD2, e.g.



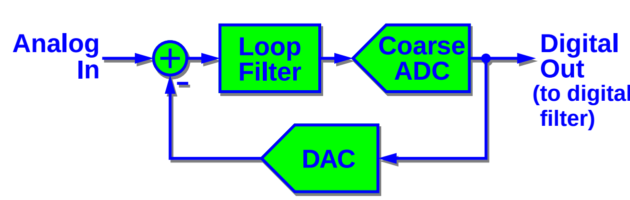
- Constraints: CMOS
  1-V and 3-V devices
  no static current

What is ΔΣ?

- **ΔΣ is NOT a fraternity**

  It is more like a way of life…

- **Simplified ΔΣ ADC structure:**



- **Key features: coarse quantization, filtering, feedback and oversampling**

  Quantization is often *quite* coarse: 1 bit!

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 $OSR \equiv f_s/(2f_B)$**

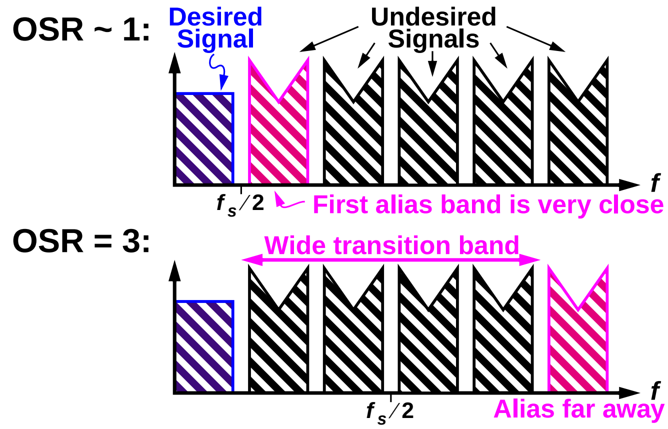
- **For a regular ADC, $OSR \sim 2 - 3$**

  To make the anti-alias filter (AAF) feasible

- **For a ΔΣ ADC, $OSR \sim 30$**

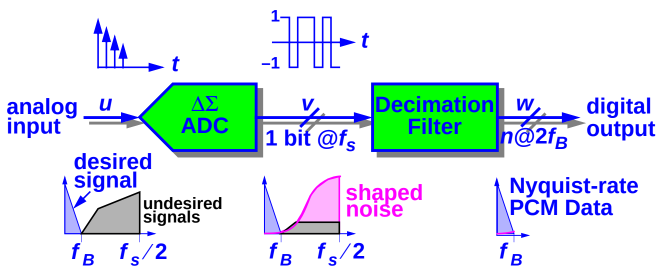
  To get adequate quantization noise suppression. All signals above f_B are removed digitally.

Oversampling Simplifies AAF

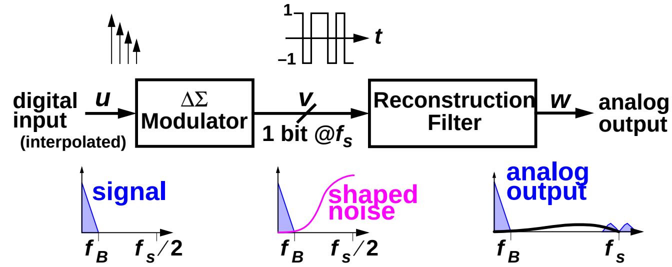


How Does A ΔΣ ADC Work?

- **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. These can often be removed with a simple RC section.

  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.

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

1. **1st- and 2nd-order modulator structures and theory of operation**
2. **Inherent linearity of binary modulators**
3. **Inherent anti-aliasing of continuous-time modulators**
4. **Spectrum estimation with FFTs**

Background
(Stuff you already know)

- The SQNR* of an ideal n-bit ADC with a full-scale sine-wave input is $(6.02n + 1.76)$ 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. $X \rightarrow \boxed{H(z)} \rightarrow Y \quad S_{yy}(f) = |H(e^{j2\pi f})|^2 \cdot S_{xx}(f)$

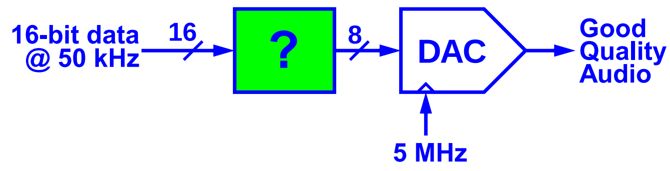
- The power in any frequency band is the integral of the PSD over that band

*. Signal-to-Quantization-Noise Ratio

Poor Man's ΔΣ DAC

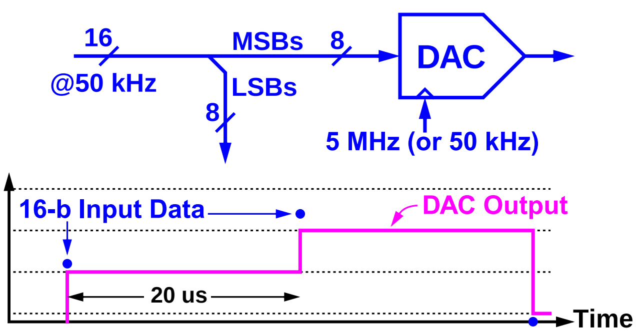
Suppose you have low-speed 16-bit data and a high-speed 8-bit DAC

- How can you get good analog performance?

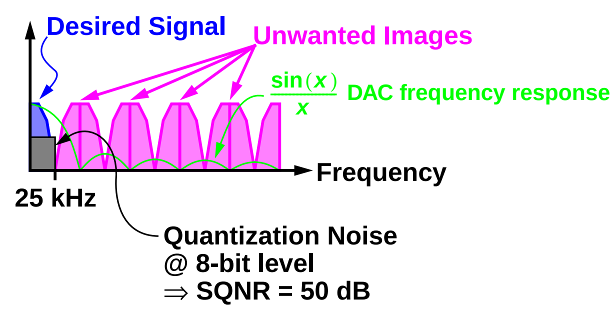


Simple (-Minded) Solution

- Only connect the MSBs; leave the LSBs hanging

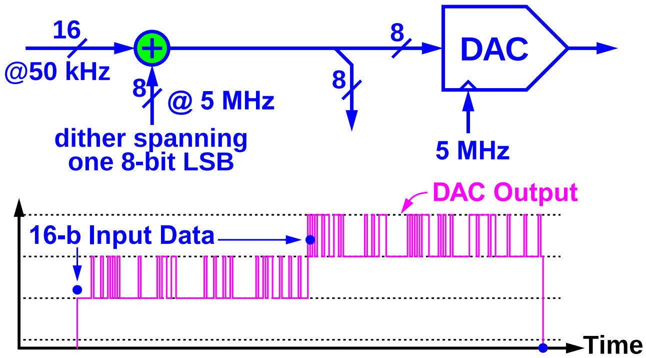


Spectral Implications



Better Solution

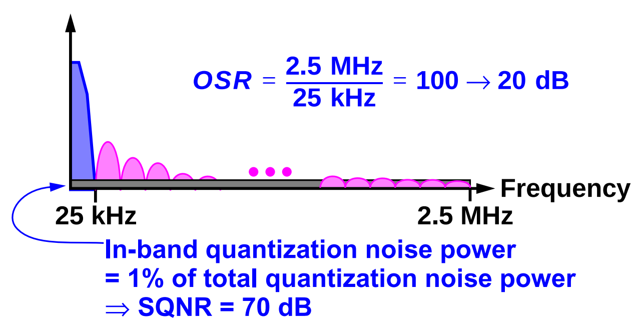
- Exploit oversampling: Clock fast and add dither



Spectral Implications

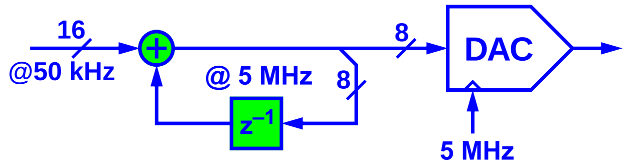
- Quantization noise is now spread over a broad frequency range

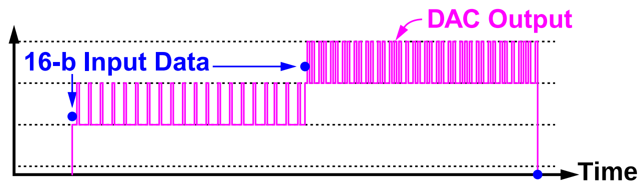
  Oversampling reduces quantization noise density



Even More Clever Method

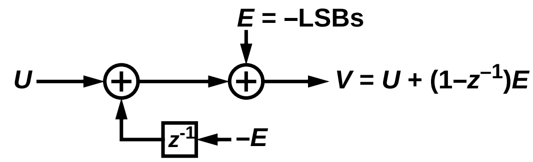
- Add LSBs back into the input data





Mathematical Model

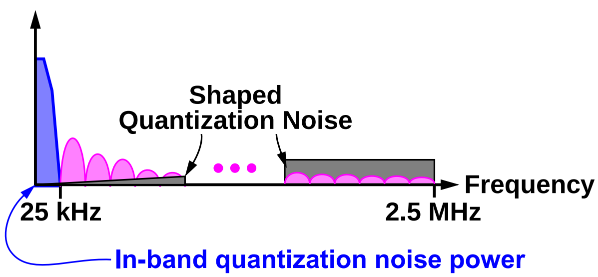
- Assume the DAC is ideal, model truncation as the addition of error:



- Hmm… Oversampling, coarse quantization and feedback. Noise-Shaping!
- Truncation noise is shaped by a $1-z^{-1}$ transfer function, which provides ~35 dB of attenuation in the 0-25 kHz frequency range

Spectral Implications

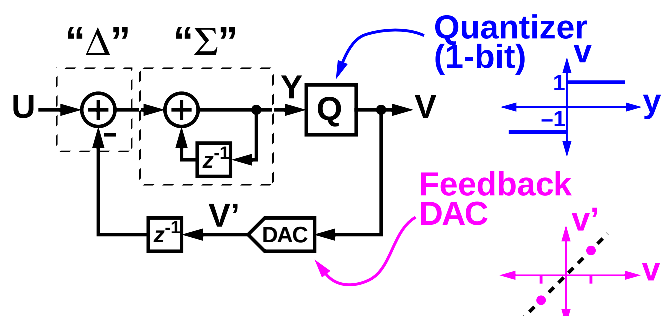
- Quantization noise is heavily attenuated at low frequencies



In-band quantization noise power
is very small, 55 dB below total power
⇒ SQNR = 105 dB!

MOD1: 1st-Order ΔΣ Modulator

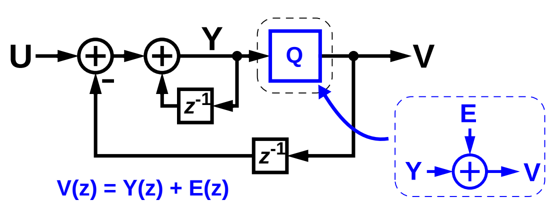
Standard Block Diagram



Since two points define a line,
a binary DAC is inherently linear.

MOD1 Analysis

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



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

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

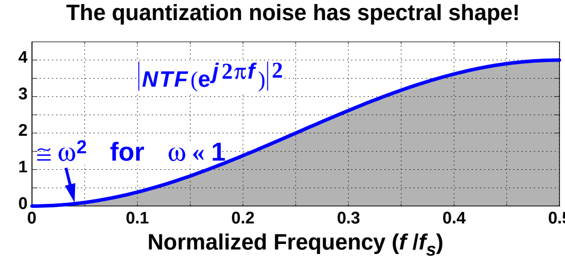
$\Rightarrow (1-z^{-1})\, V(z) = U(z) - z^{-1}V(z) + (1-z^{-1})E(z)$

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

The Noise Transfer Function

- 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!



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

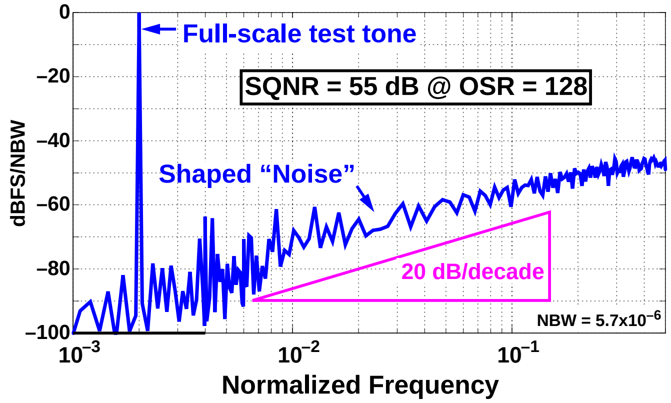
In-band Noise Power

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

$$N_0^2 = \int_0^{\omega_B} |H(e^{j\omega})|^2 S_{ee}(\omega)d\omega \cong \frac{\sigma_e^2}{\pi}\int_0^{\omega_B}\omega^2 d\omega$$

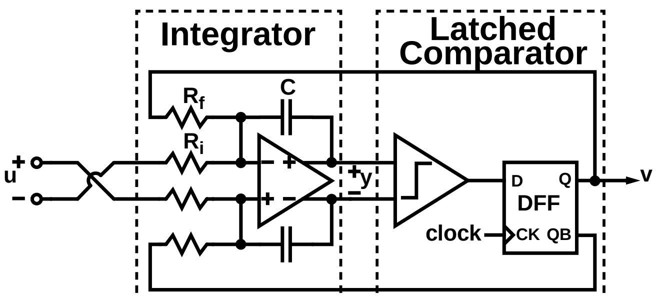
- **Since $OSR \equiv \frac{\pi}{\omega_B}$, $N_0^2 = \frac{\pi^2\sigma_e^2}{3}(OSR)^{-3}$**
- **For MOD1, an octave increase in *OSR* increases SQNR by 9 dB**
  1.5-bit/octave SQNR-OSR trade-off.

A Simulation of MOD1

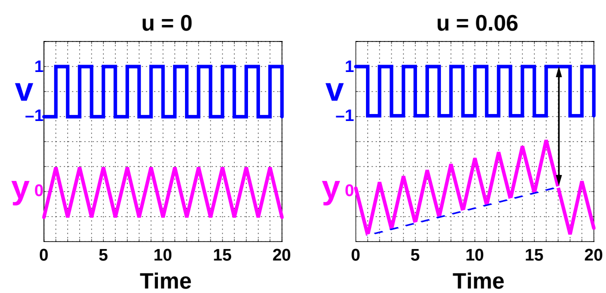


CT Implementation of MOD1

- **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



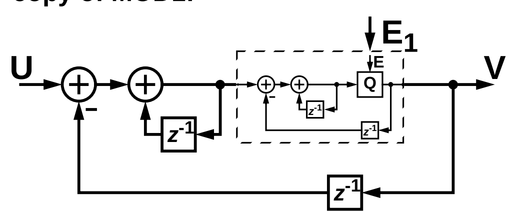
- **With *u*=0, *v* alternates between +1 and –1**
- **With *u*>0, *y* drifts upwards; *v* contains consecutive +1s to counteract this drift**

Summary So Far

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

MOD2: 2nd-Order $\Delta\Sigma$ Modulator

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

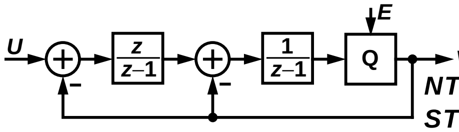


$V(z) = U(z) + (1-z^{-1})E_1(z)$,

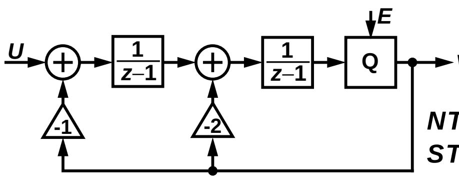
$E_1(z) = (1-z^{-1})E(z)$

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

Simplified Block Diagrams

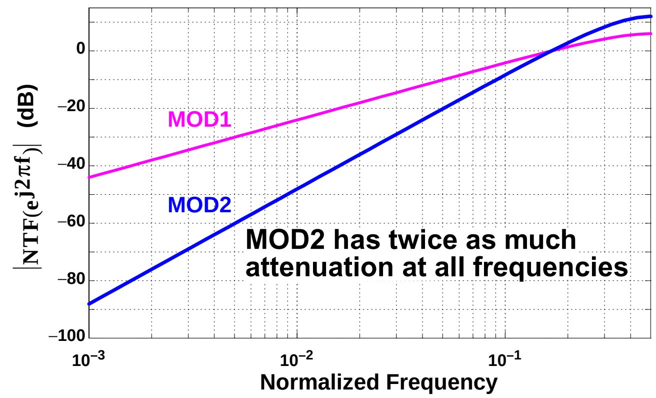


$$NTF(z) = (1 - z^{-1})^2$$
$$STF(z) = z^{-1}$$



$$NTF(z) = (1 - z^{-1})^2$$
$$STF(z) = z^{-2}$$

NTF Comparison



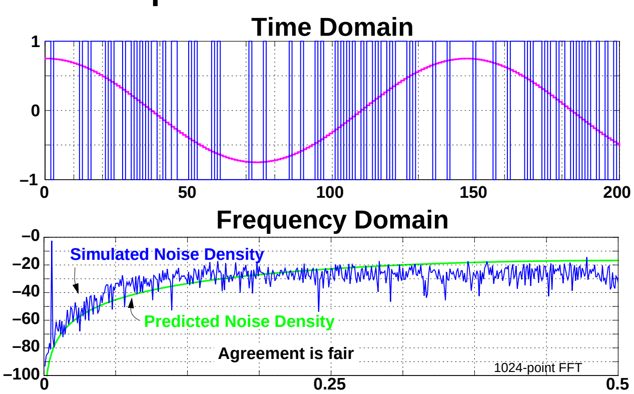
In-band Noise Power

- For MOD2, $|H(e^{j\omega})|^2 \approx \omega^4$
- As before, $N_0^2 = \int_0^{\omega_B} |H(e^{j\omega})|^2 S_{ee}(\omega) d\omega$ and $S_{ee}(\omega) = \sigma_e^2 / \pi$
- So now $N_0^2 = \dfrac{\pi^4 \sigma_e^2}{5} (OSR)^{-5}$ ✱

  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 SNR by 15 dB (2.5 bits)"

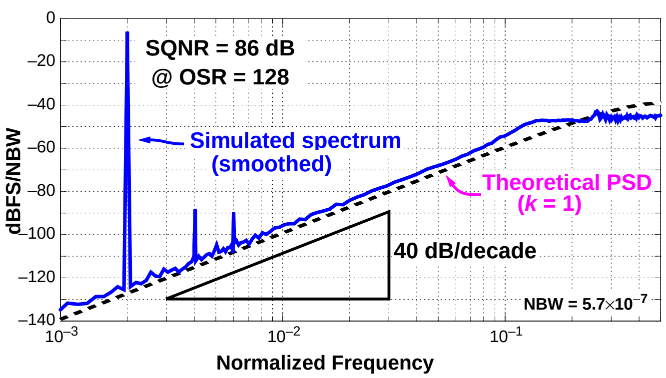
Simulation Example

Input at 75% of FullScale



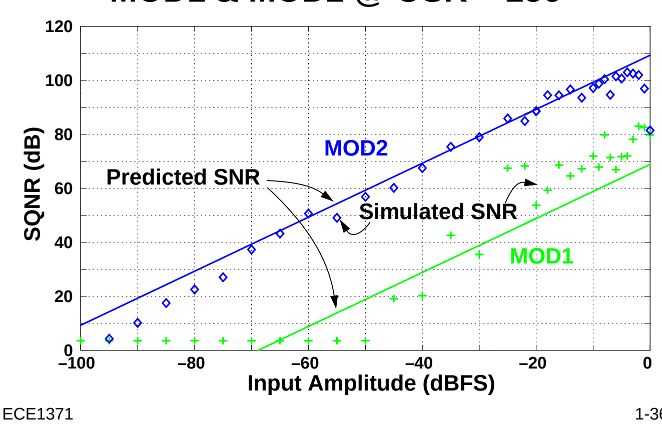
Simulated MOD2 PSD

Input at 50% of FullScale

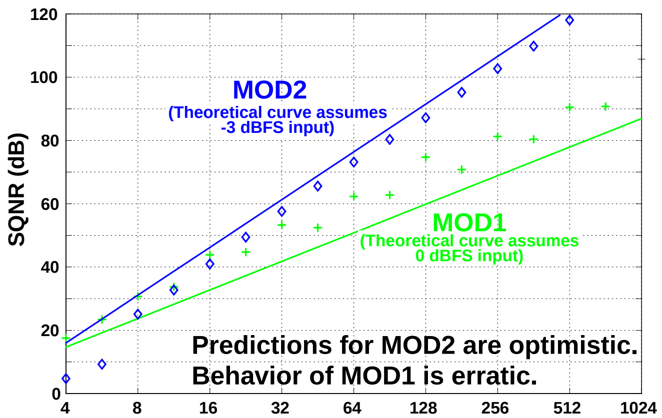


SQNR vs. Input Amplitude

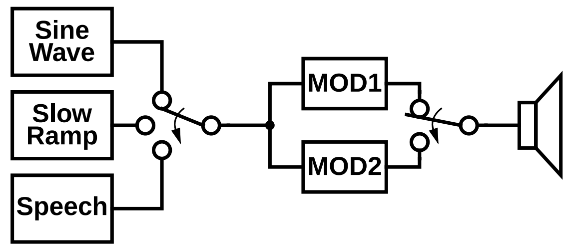
MOD1 & MOD2 @ OSR = 256



SQNR vs. OSR



Audio Demo: MOD1 vs. MOD2



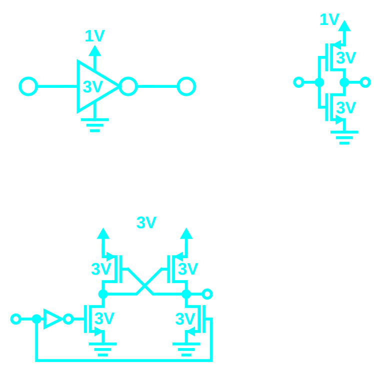
MOD1 + MOD2 Summary

- $\Delta\Sigma$ 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 and ADC; image filter in a DAC
- Binary quantization yields inherent linearity
- CT loop filter provides inherent anti-aliasing
- MOD2 is better than MOD1
  15 dB/octave vs. 9 dB/octave SNR-OSR trade-off.
  Quantization noise more white.
  Higher-order modulators are even better.

NLCOTD

3V $\rightarrow$ 1V:

1V $\rightarrow$ 3V:

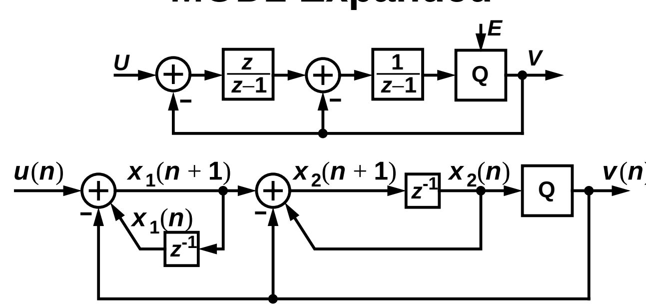


Homework #1

Create a Matlab function that computes MOD2'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 DC inputs in [–1,1].
2. Produce a spectral plot like that on Slide 35.
3. a) Construct a SQNR vs. input amplitude curve for OSR = 128 for amplitudes from –100 to 0 dBFS.
   b) Determine approximately how much the interstage gain and feedback coefficients need to shift in order to have a significant (~3-dB) impact.
4. Compare the in-band quantization noise of your system with a half-scale sine-wave input against the relation given on Slide 33 for OSR in $[2^3,2^{10}]$.

MOD2 Expanded



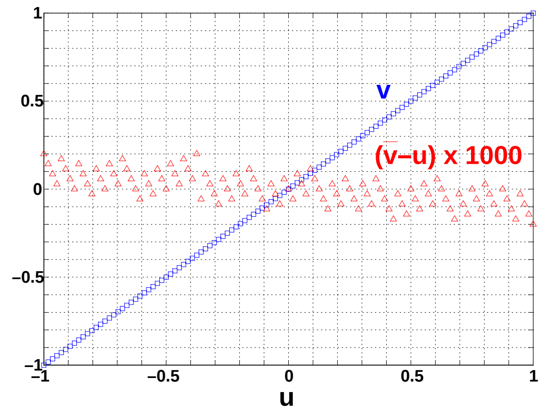
Difference Equations:

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

Example Matlab Code

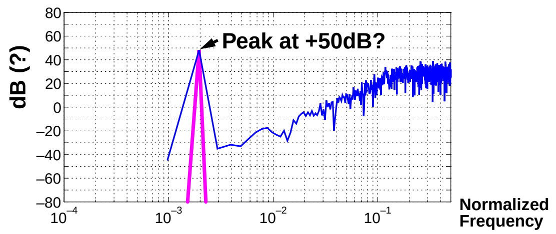
```
function [v,x] = simulateMOD2(u)
    x1 = 0;
    x2 = 0;
    for i = 1:length(u)
        v(i) = quantize( x2 );
        x1 = x1 + u(i) - v(i);
        x2 = x2 + x1 - v(i);
    end
return

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


~100 10^4-Point Simulations



Example Spectrum

```
Nfft = 2^10;
ftest = 2;
t = 0:Nfft-1;
u = 0.5*sin(2*pi*ftest/Nfft*t); % Has ftest cycles in Nfft points
v = simulateMOD2(u);
U = fft(u);
V = fft(v);
f = linspace(0,1,Nfft+1); f=f(1:Nfft);
semilogx(f,dbv(U),'m', f,dbv(V),'b');
figureMagic([1e-4 0.5],[],[],[-80 80],10,2);
```



FFT Considerations (Partial)

- **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** $x(n) = A\sin(2\pi fn/N)^{\dagger}$**, then**

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

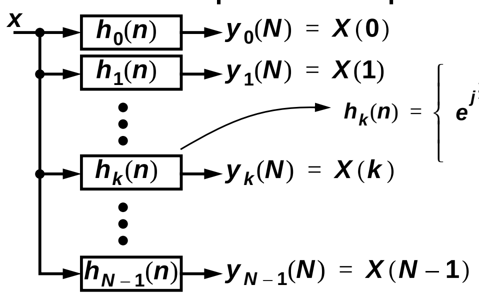
$\Rightarrow$ **Need to divide FFT by** $(N/2)$ **to get A.**

---

†. 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:**



$$h_k(n) = \begin{cases} e^{j\frac{2\pi k}{N}n} & , 0 \le n < N \\ 0 & , \text{otherwise} \end{cases}$$

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

How To Do Smoothing

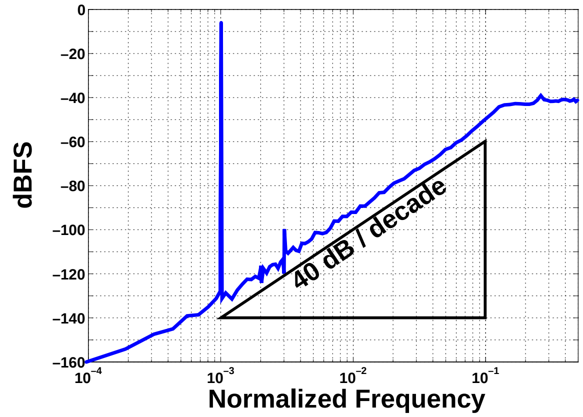
1. **Average multiple FFTs**

   Implemented by MATLAB's `psd()` function
2. **Take one big FFT and "filter" the spectrum**

   Implemented by the $\Delta\Sigma$ 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**

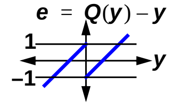
Smoothed Spectrum

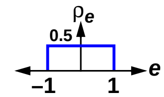


Quantization Noise Spectrum?

- Assume that the quantization error e is uniformly distributed in [–1,+1]

$e = Q(y) - y$

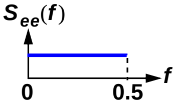




$$\sigma_e^2 = \int \rho_e(e) e^2 de = \left(0.5 \cdot \frac{e^3}{3}\right)\Bigg|_{-1}^{1} = \frac{1}{3}$$

- Assume e is white

1-sided PSD: $S_{ee}(f)$

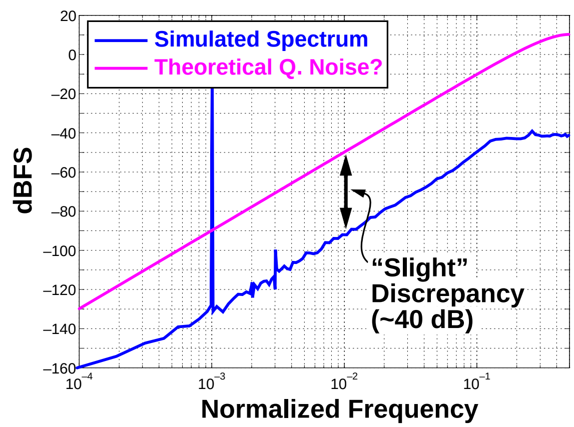


$$\sigma_e^2 = \int_0^{0.5} S_{ee}(f) df$$

$$\Rightarrow S_{ee}(f) = 2\sigma_e^2$$

- Multiply $S_{ee}(f)$ by $|NTF(e^{j2\pi f})|^2$ to get the PSD of the shaped error

Simulation vs. Theory



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)

$\Rightarrow$ We need to do the same for the error signal.

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

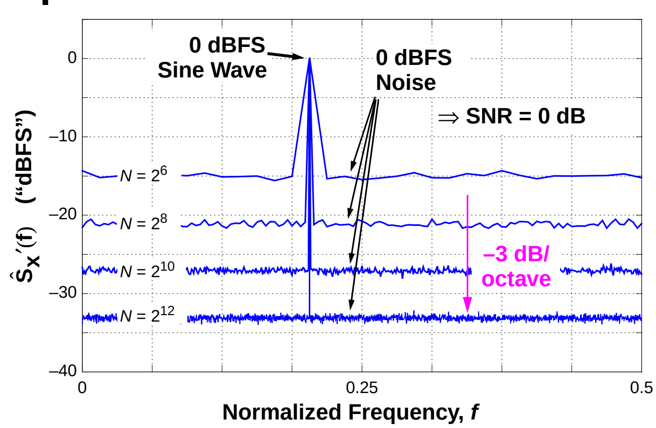
But this makes the discrepancy 3 dB worse.

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

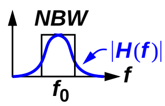
  The greater the number of bins, the less power there is in any one bin.
- Doubling N reduces the power per bin by a factor of 2 (i.e. 3 dB)

  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

$$h(n) = \exp\left(j\frac{2\pi k}{N}n\right)$$

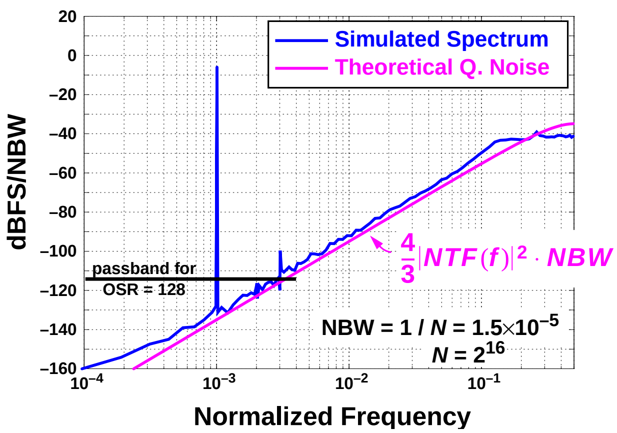
$$H(f) = \sum_{n=0}^{N-1} h(n)\exp(-j2\pi fn)$$

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

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

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

Better Spectral Plot



SQNR Calculation

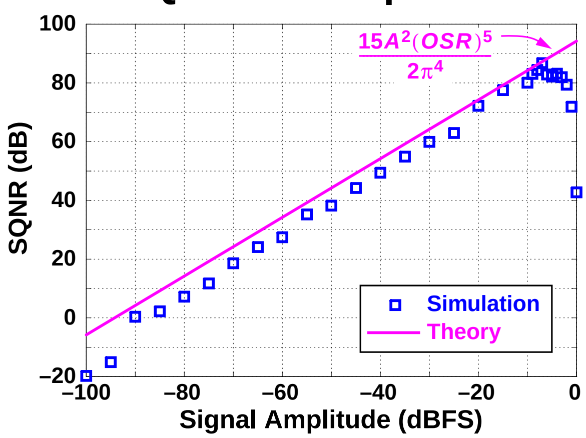
- S = power in the signal bin
- QN = sum of the powers in the non-signal in-band noise bins

$\Rightarrow$ Using MATLAB to perform these calculations for the preceding simulation yields SQNR = 84.2 dB at OSR = 128

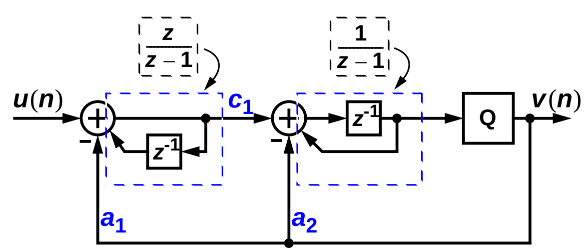
- Can also eyeball SQNR from the plot:
  S = –6 dB
  QN = –113 + dbp‡(BW/NBW) = –89 dB
  $\Rightarrow$ SQNR = –83 dB

‡. dbp(x) = 10log$_{10}$(x).

SQNR vs. Amplitude

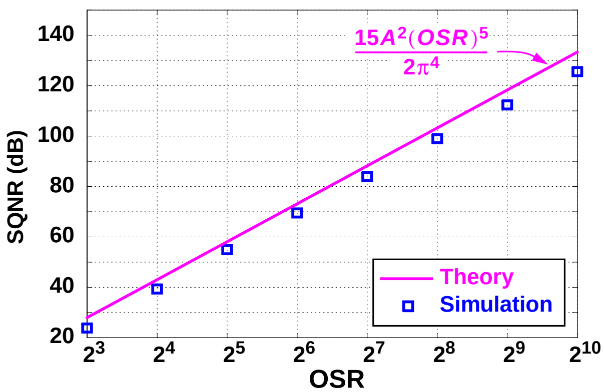


Tolerable Coefficient Errors?



- a_1 & a_2 are the feedback coefficients; nominally 1
- c_1 is the interstage coefficient; nominally 1
- You should find that the SQNR stays high even if these coefficients individually vary over a 2:1 range

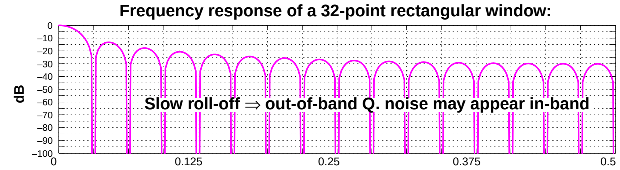
SQNR vs. OSR for MOD2
Half-Scale Input (A = 0.5)



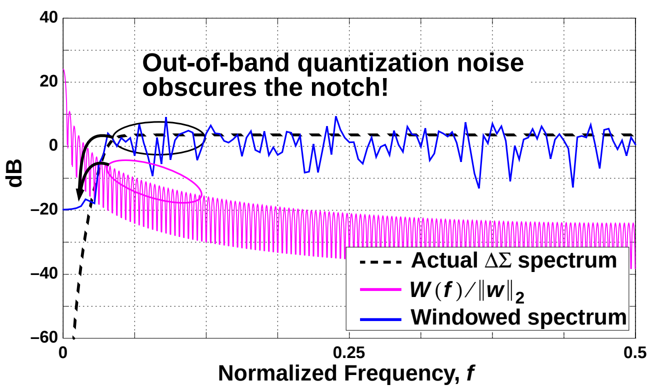
Windowing

- $\Delta\Sigma$ 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"

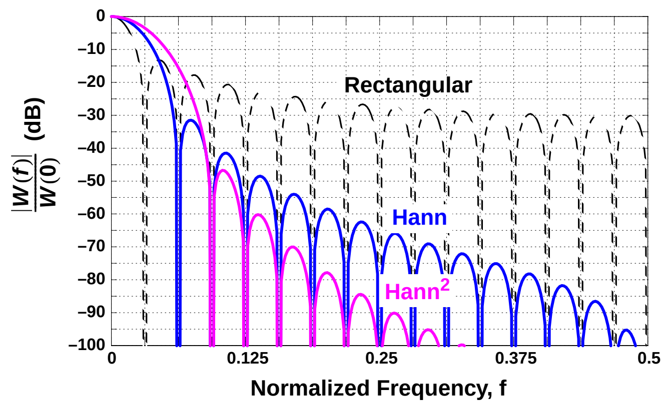
Frequency response of a 32-point rectangular window:



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	$\dfrac{1-\cos\frac{2\pi n}{N}}{2}$	$\left(\dfrac{1-\cos\frac{2\pi n}{N}}{2}\right)^2$
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	$N/2$	$3N/8$
$NBW = \dfrac{\|w\|_2^2}{W(0)^2}$	$1/N$	$1.5/N$	$35/18N$

†. 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 ⇒ >15 in-band bins ⇒ $> 30 \cdot OS$

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

Good FFT Practice

[Appendix A of Schreier & Temes]

- **Use coherent sampling**
  Need an integer number of cycles in the record.
- **Use windowing**
  A Hann window works well.
- **Use enough points**
  $N = 64 \cdot OSR$
- **Scale the spectrum**
  A full-scale sine wave should yield a 0-dBFS peak.
- **State the noise bandwidth**
  For a Hann window, $NBW = 1.5 / N$.
- **Smooth the spectrum if you want a pretty plot**

What You Learned Today

And what the homework should solidify

1. **MOD1 and MOD2 structure and linear theory**
   SQNR-OSR trade-offs:
   9 dB/octave for MOD1
   15 dB/octave for MOD2
2. **Inherent linearity of binary modulators**
3. **Inherent anti-aliasing of continuous-time modulators**
4. **Proper use of FFTs for spectral analysis**
5. **(Hwk) MOD1 and MOD2 are tolerant of large coefficient errors**