ECE1371 Advanced Analog Circuits
Lecture 2

Higher-Order Modulators: MOD2 and MOD*N*

Richard Schreier
`richard.schreier@analog.com`

Trevor Caldwell
`trevor.caldwell@utoronto.ca`

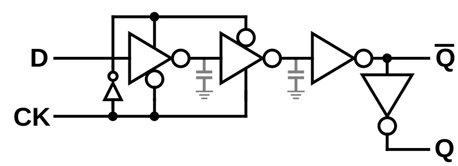
Course Goals

- **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.

Date			Lecture	Ref	Homework
2012-01-12	RS	1	Introduction: MOD1	ST 2, A	1: MOD1 in Matlab
2012-01-19	RS	2	MOD2 & MOD*N*	ST 3, 4, B	2: MOD2 in Matlab
2012-01-26	RS	3	Example Design: Part 1	ST 9.1, CCJM 14	3: Sw.-level MOD2
2012-02-02	TC	4	SC Circuits	R 12, CCJM 14	4: SC Integrator
2012-02-09	TC	5	Amplifier Design		
2012-02-16	TC	6	Amplifier Design		5: SC Int w/ Amp
2012-02-23	Reading Week + ISSCC– No Lecture				
2012-03-01	RS	7	Example Design: Part 2	CCJM 18	Start Project
2012-03-08	RS	8	Comparator & Flash ADC	CCJM 10	
2012-03-15	TC	9	Noise in SC Circuits	ST C	
2012-03-22	TC	10	Matching & MM-Shaping	ST 6.3-6.5, +	
2012-03-29	RS	11	Advanced ΔΣ	ST 6.6, 9.4	
2012-04-05	TC	12	Pipeline and SAR ADCs	CCJM 15, 17	
2012-04-12	No Lecture				
2012-04-19	Project Presentation				

NLCOTD: Dynamic Flip-Flop

- **Standard CMOS version**

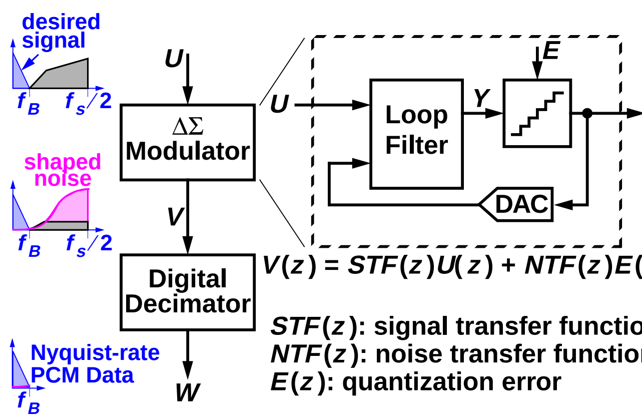


- **Can the circuit be simplified?**
  Is a complementarty clock necessary?

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

1. **Second-order modulator (MOD2)**
2. **Arbitrary-order modulator (MOD*N*) design with the ΔΣ Toolbox**

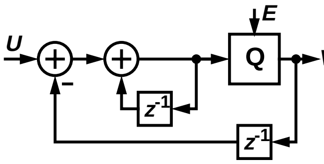
Review: A ΔΣ ADC System



$V(z) = STF(z)U(z) + NTF(z)E(z)$

***STF*(*z*)**: signal transfer function
***NTF*(*z*)**: noise transfer function
***E*(*z*)**: quantization error

Review: MOD1

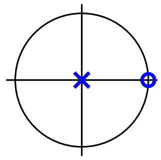


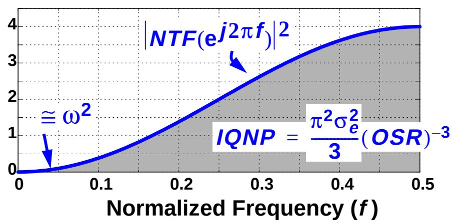
$V(z) = NTF(z)E(z) + STF(z)U(z)$

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

$STF(z) = 1$

NTF poles & zeros:





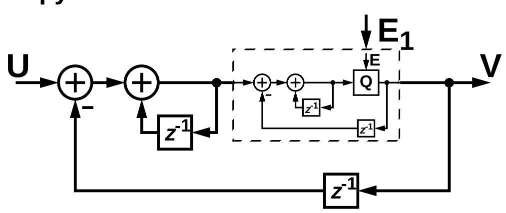
Review Summary

- $\Delta\Sigma$ 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.
  9 dB/octave SQNR-OSR trade-off.
- MOD1-CT has *inherent anti-aliasing*

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

[Ch. 3 of Schreier & Temes]

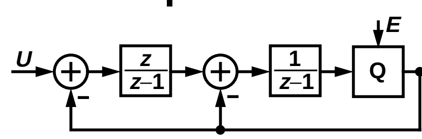
- Replace the quantizer in MOD1 with another copy of MOD1 in a recursive fashion:



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

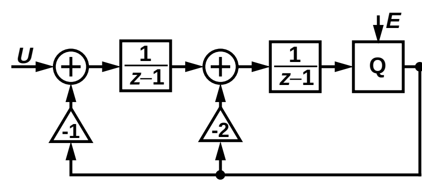
$\Rightarrow V(z) = U(z) + (1-z^{-1})^2E(z)$

Simplified Block Diagrams



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

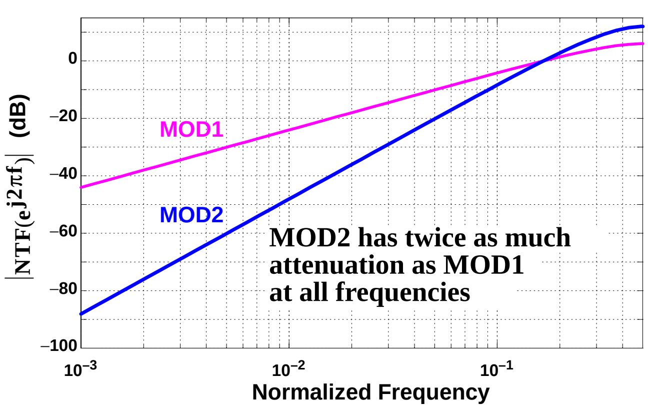
$STF(z) = z^{-1}$



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

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

NTF Comparison



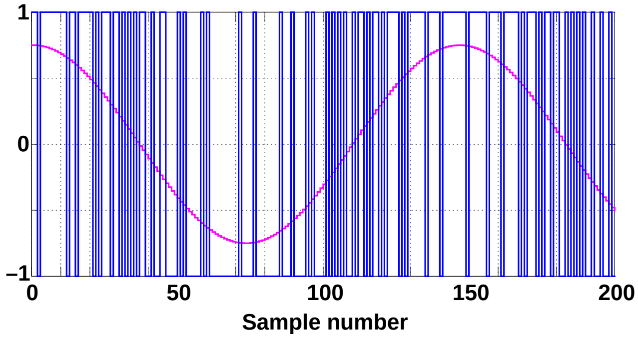
In-band Quant. Noise Power

- 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)"

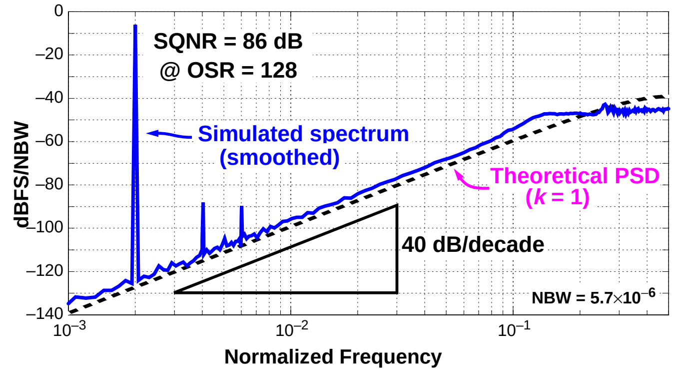
Simulation Example

Input at 75% of FullScale



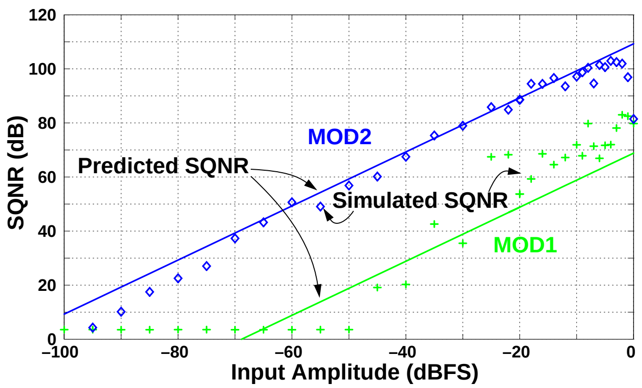
Simulated MOD2 PSD

Input at 50% of FullScale

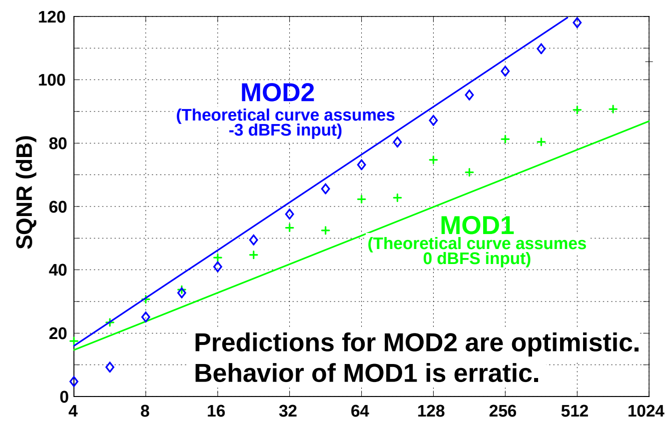


SQNR vs. Input Amplitude

MOD1 & MOD2 @ OSR = 256

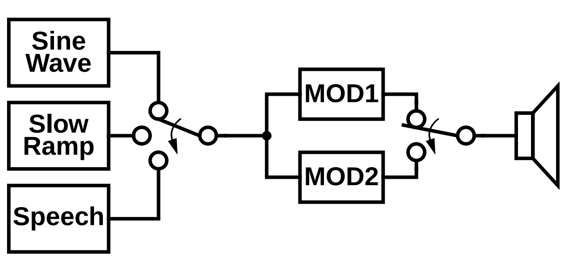


SQNR vs. OSR



Audio Demo: MOD1 vs. MOD2

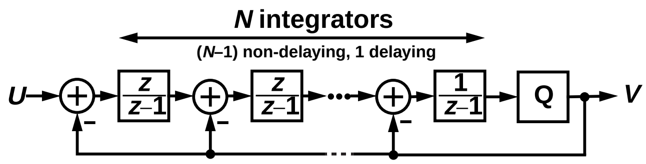
[dsdemo4]



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.

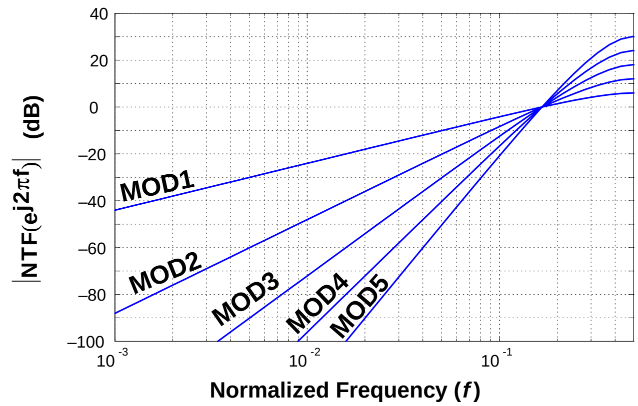
MODN
[Ch. 4 of Schreier & Temes]



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

- MODN's NTF is the N^{th} power of MOD1's NTF

NTF Comparison



Predicted Performance

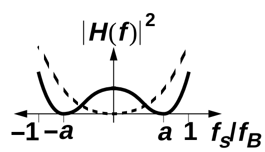
- In-band quantization noise power

$$IQNP = \int_0^{0.5/OSR} |NTF(e^{j2\pi f})|^2 \cdot S_{ee}(f)\ df$$
$$\approx \int_0^{0.5/OSR} (2\pi f)^{2N} \cdot 2\sigma_e^2 df$$
$$= \frac{\pi^{2N}}{(2N+1)(OSR)^{2N+1}}\sigma_e^2$$

- Quantization noise drops as the $(2N+1)^{th}$ power of OSR— $(6N+3)$ dB/octave SQNR-OSR trade-off

Improving NTF Performance– NTF Zero Optimization

- Minimize the integral of $|NTF|^2$ over the passband
  - Normalize passband edge to 1 for ease of calculation:



Need to find the a_i which minimize the integral

$$\int_{-1}^{1}(x^2 - a_1^2)^2 dx\ , \qquad n = 2$$
$$\int_{-1}^{1}x^2(x^2 - a_1^2)^2 dx\ , \qquad n = 3$$
$$\int_{-1}^{1}(x^2 - a_1^2)^2(x^2 - a_2^2)^2 dx\ , \qquad n = 4$$
$$\vdots$$

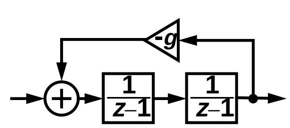
Solutions Up to Order = 8

Order	Optimal Zero Placement Relative to f_B	SQNR Improvement
1	0	0 dB
2	$\pm 1/\sqrt{3}$	3.5 dB
3	$0, \pm\sqrt{3/5}$	8 dB
4	$\pm\sqrt{3/7 \pm \sqrt{(3/7)^2 - 3/35}}$	13 dB
5	$0, \pm\sqrt{5/9 \pm \sqrt{(5/9)^2 - 5/21}}$ [Y. Yang]	18 dB
6	±0.23862, ±0.66121, ±0.93247	23 dB
7	0, ±0.40585, ±0.74153, ±0.94911	28 dB
8	±0.18343, ±0.52553, ±0.79667, ±0.96029	34 dB

Topological Implication

- Feedback around pairs integrators:

2 Delaying Integrators



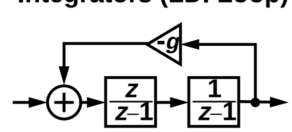
Poles are the roots of

$$1 + g\left(\frac{1}{z-1}\right)^2 = 0$$

i.e. $z = 1 \pm j\sqrt{g}$

Not quite on the unit circle, but fairly close if g<<1.

Non-delaying + Delaying Integrators (LDI Loop)



Poles are the roots of

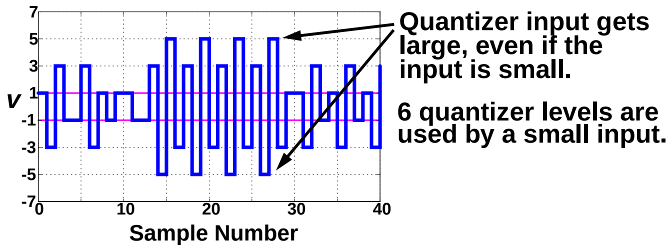
$$1 + \frac{gz}{(z-1)^2} = 0$$

i.e. $z = e^{\pm j\theta}$, $\cos\theta = 1 - g/2$

Precisely on the unit circle, regardless of the value of g.

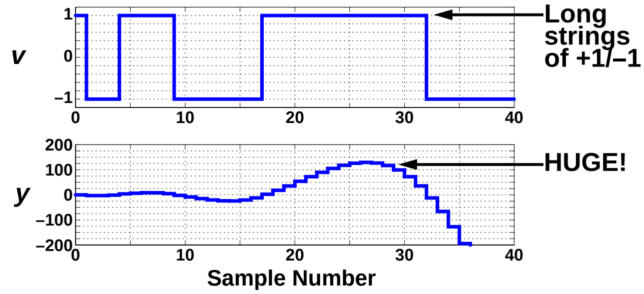
Problem: A High-Order Modulator Wants a Multi-bit Quantizer

E.g. MOD3 with an Infinite Quantizer and Zero Input



Simulation of MOD3-1b

(MOD3 with a Binary Quantizer)



- MOD3-1b is unstable, even with zero input!

Solutions to the Stability Problem

Historical Order

1. Multi-bit quantization

   Initially considered undesirable because we lose the inherent linearity of a 1-bit DAC.

2. More general NTF (not pure differentiation)

   Lower the NTF gain so that quantization error is amplified less.

   Unfortunately, reducing the NTF gain reduces the amount by which quantization noise is attenuated.

3. Multi-stage (MASH) architecture

- Combinations of the above are possible

Multi-bit Quantization

A modulator with $NTF = H$ and $STF = 1$ is guaranteed to be stable if $|u| < u_{max}$ at all times, where $u_{max} = nlev + 1 - \|h\|_1$ and $\|h\|_1 = \sum_{i=0}^{\infty} |h(i)|$

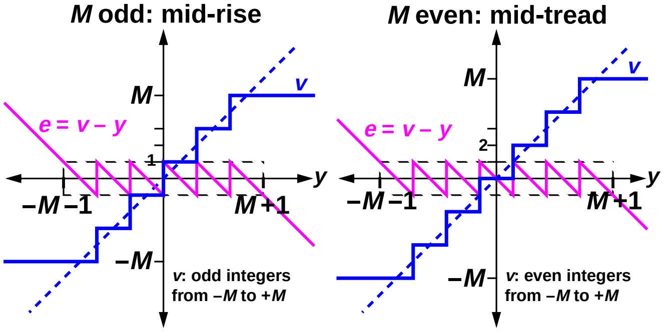
- In MODN $H(z) = (1 - z^{-1})^N$, so $h(n) = \{1, -a_1, a_2, -a_3, \ldots (-1)^N a_N, 0 \ldots\}$, $a_i > 0$ and thus $\|h\|_1 = H(-1) = 2^N$
- $nlev = 2^N$ implies $u_{max} = nlev + 1 - \|h\|_1 = 1$

  MODN is guaranteed to be stable with an N-bit quantizer if the input magnitude is less than $\Delta/2 = 1$.

  This result is quite conservative.
- Similarly, $nlev = 2^{N+1}$ guarantees that MODN is stable for inputs up to 50% of full-scale

M-Step Symmetric Quantizer

$\Delta = 2$, ($nlev = M + 1$)



- No-overload range: $|y| \le nlev \Rightarrow |e| \le \Delta/2 = 1$

Inductive Proof of $\|h\|_1$ Criterion

- Assume STF = 1 and $(\forall n)(|u(n)| \le u_{max})$
- Assume $|e(i)| \le 1$ for $i < n$.[Induction Hypothesis]

$$
\begin{aligned}
|y(n)| &= \left| u(n) + \sum_{i=1}^{\infty} h(i) e(n-i) \right| \\
&\le u_{max} + \sum_{i=1}^{\infty} |h(i)| |e(n-i)| \\
&\le u_{max} + \sum_{i=1}^{\infty} |h(i)| = u_{max} + \|h\|_1 - 1
\end{aligned}
$$

Then $u_{max} = nlev + 1 - \|h\|_1$

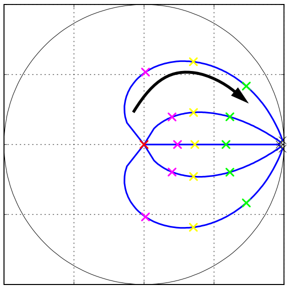
$\Rightarrow \quad |y(n)| \le nlev$

$\Rightarrow \quad |e(n)| \le 1$

- So by induction $|e(i)| \le 1$ for all $i > 0$

More General NTF

- Instead of $NTF(z) = A(z)/B(z)$ with $B(z) = z^n$, use a more general $B(z)$
  - Roots of B are the poles of the NTF and must be inside the unit circle.



Moving the poles away from $z = 1$ toward $z = 0$ makes the gain of the NTF approach unity.

The Lee Criterion for Stability in a 1-bit Modulator: $\|H\|_\infty \le 2$

[Wai Lee, 1987]

- The measure of the "gain" of H is the maximum magnitude of H over frequency, aka the *infinity-norm* of H: $\|H\|_\infty \equiv \max_{\omega \in [0, 2\pi]} (H(e^{j\omega}))$

Q: Is the Lee criterion <u>necessary</u> for stability?

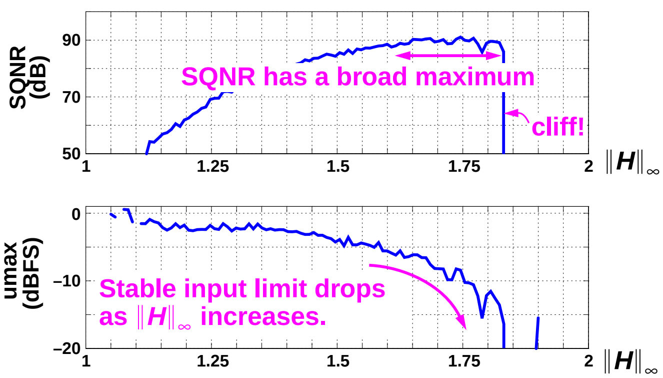
No. MOD2 is stable (for DC inputs less than FS) but $\|H\|_\infty = 4$.

Q: Is the Lee criterion <u>sufficient</u> to ensure stability?

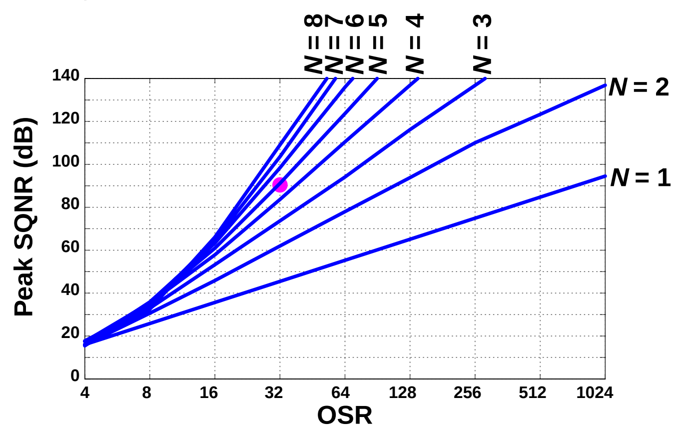
No. There are lots of counter-examples, but $\|H\|_\infty \le 1.5$ often works.

Simulated SQNR vs. $\|H\|_\infty$

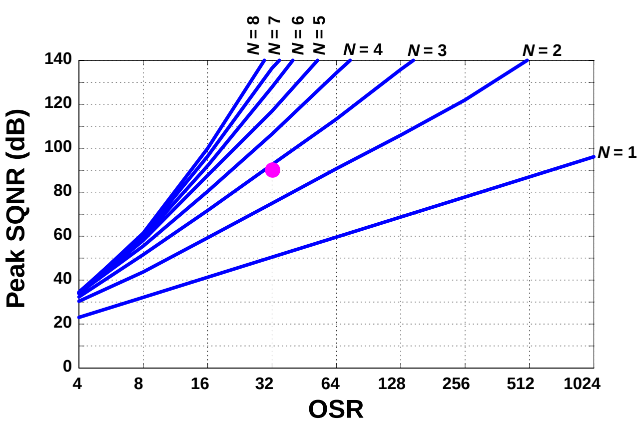
5th-order NTFs; 1-b Quant.; OSR = 32



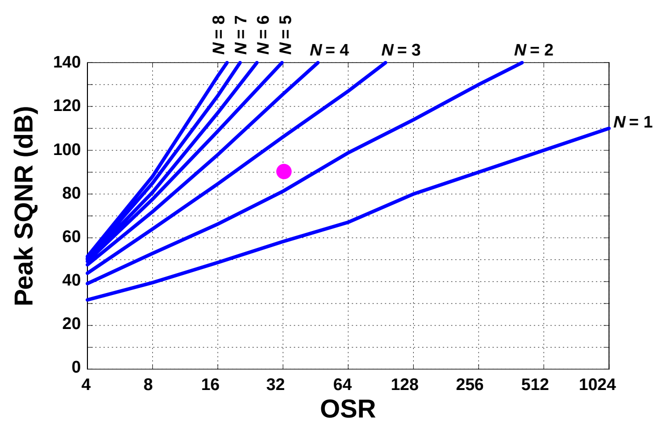
SQNR Limits— 1-bit Modulation



SQNR Limits for 2-bit Modulators

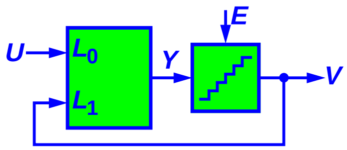


SQNR Limits for 3-bit Modulators



Generic Single-Loop ΔΣ ADC

- **Linear Loop Filter + Nonlinear Quantizer:**



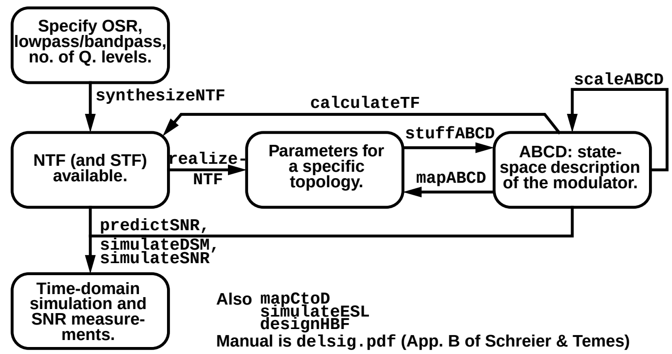
$$Y = L_0 U + L_1 V \qquad V = Y + E \;\Rightarrow\; \boxed{V = STF \cdot U + NTF \cdot E}\text{, where}$$

$$NTF = \frac{1}{1 - L_1} \;\;\&\;\; STF = L_0 \cdot NTF$$

Inverse Relations:
$L_1 = 1 - 1/NTF$, $L_0 = STF\,/\,NTF$

ΔΣ Toolbox

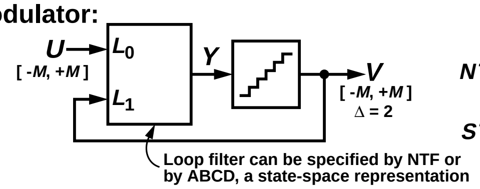
`http://www.mathworks.com/matlabcentral/fileexchange`
Search for "Delta Sigma Toolbox"



Also `mapCtoD`, `simulateESL`, `designHBF`
Manual is `delsig.pdf` (App. B of Schreier & Temes)

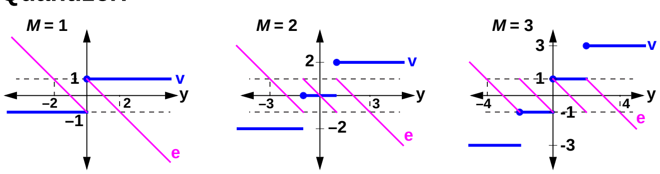
ΔΣ Toolbox Modulator Model

Modulator:



$$NTF = \frac{1}{1 - L_1} \qquad STF = \frac{L_0}{1 - L_1}$$

Quantizer:



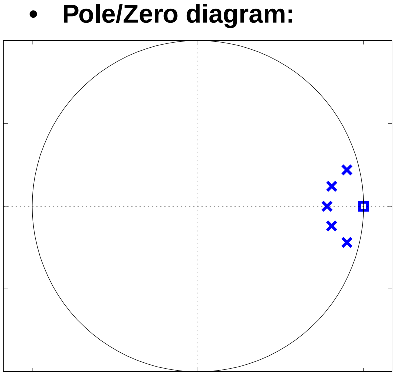
Mid-tread quantizer; *v*: even integers [$-M$,$+M$]
Mid-rise quantizer; *v*: odd integers [$-M$,$+M$]

NTF Synthesis
`synthesizeNTF`

- **Not all NTFs are realizable**
  Causality requires $h(0) = 1$, or, in the frequency domain, $H(\infty) = 1$. Recall $H(z) = h(0)z^0 + h(1)z^{-1} + \ldots$
- **Not all NTFs yield stable modulators**
  Rule of thumb for single-bit modulators: $\|H\|_\infty < 1.5$ [Lee].
- **Can optimize NTF zeros to minimize the mean-square value of *H* in the passband**
- **The NTF and STF share poles, and in some modulator topologies the STF zeros are not arbitrary**
  Restrict the NTF such that an all-pole STF is maximally flat. (Almost the same as Butterworth poles.)

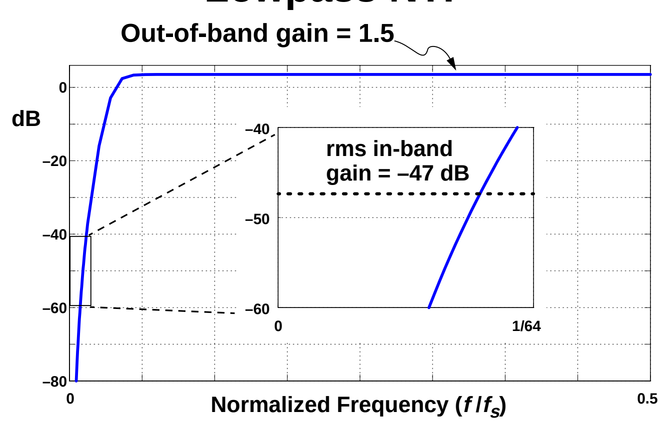
Lowpass Example [dsdemo1]
5^{th}-order NTF, all zeros at DC

- **Pole/Zero diagram:**



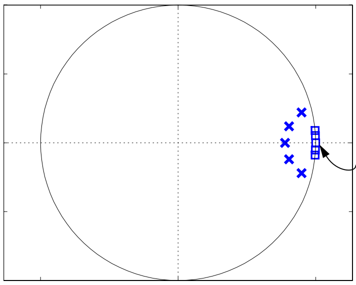
```
OSR = 32;
H = synthesizeNTF(5);
plotPZ(H);

f = linspace(0,0.5);
z = exp(2i*pi*f);
H_z = evalTF(H,z);
plot(f,dbv(H_z));
g = rmsGain(H,0,0.5/OSR)
```


Lowpass NTF



Improved 5th-Order Lowpass NTF

Zeros optimized for OSR=32

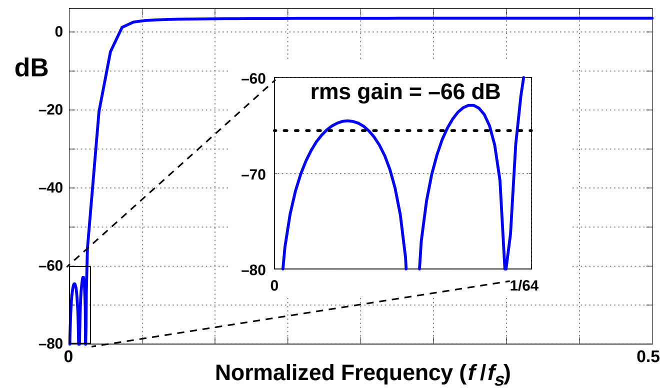


optimization flag

```
OSR = 32;
H = synthesizeNTF(5,OSR,1);
...
```

Zeros spread across the band-of-interest to minimize the rms value of the NTF.

Improved NTF

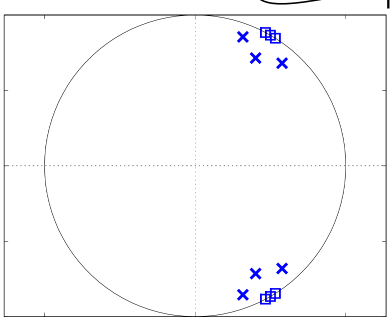


Bandpass Example

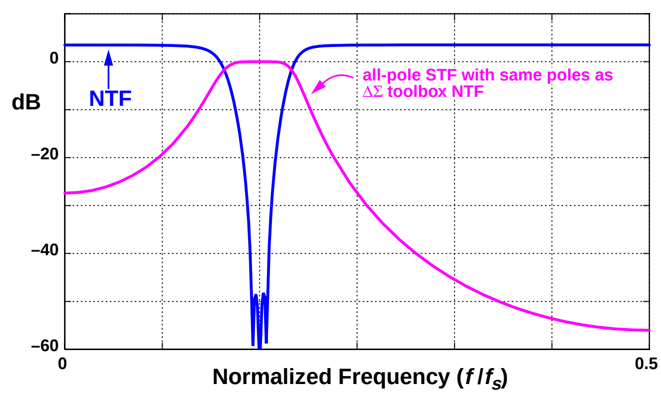
```
OSR = 64;
f0 = 1/6;
H=synthesizeNTF(6,OSR,1,[],f0);...
```

center frequency

[] or NaN means use default value, i.e. Hinf = 1.5



Bandpass NTF and STF



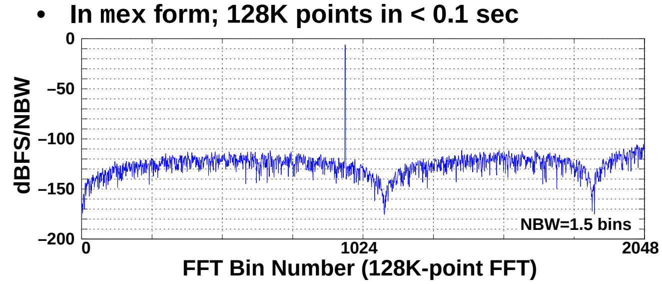
Summary: NTF Selection

- If OSR is high, a single-bit modulator may work
- To improve SQNR,
  - Optimize zeros,
  - Increase $\|H\|_\infty$, or
  - Increase order.
- If SQNR is insufficient, must use a multi-bit design
  - Can turn all the above knobs to enhance performance.
- Feedback DAC assumed to be ideal

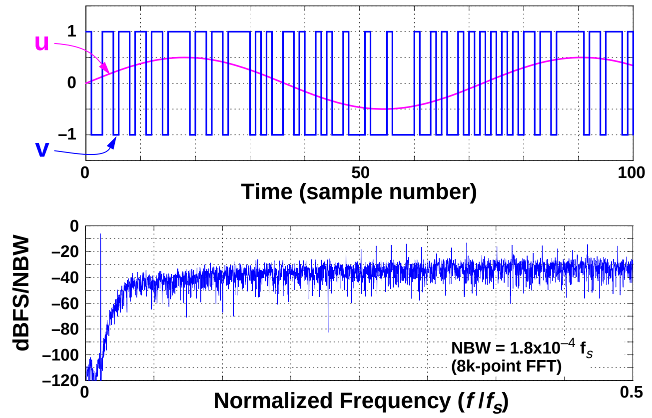
NTF-Based Simulation [dsdemo2]

```
order=5; OSR=32;
ntf = synthesizeNTF(order,OSR,1);
N=2^17; fbin=959; A=0.5; % 128K points
input = A*sin(2*pi*fbin/N*[0:N-1]);
output = simulateDSM(input,ntf);
spec = fft(output.*ds_hann(N)/(N/4));
plot(dbv(spec(1:N/(2*OSR))));
```

- In mex form; 128K points in < 0.1 sec

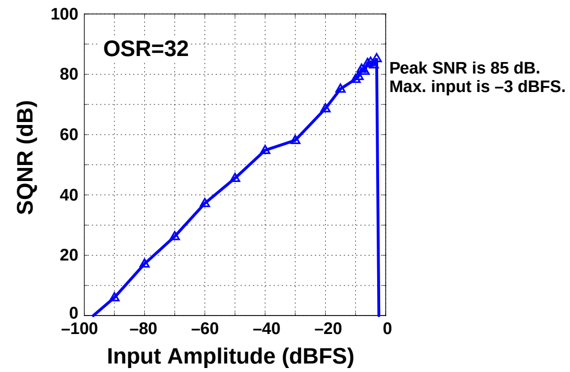


Simulation Example Cont'd



SNR vs. Amplitude: `simulateSNR`

```
[snr amp] = simulateSNR(ntf,OSR);
plot(amp,snr,'b-^');
```



Homework #2– Part A

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. Plot the error $\overline{v} - u$ as a function of u using 100-point rectangular and triangular* averages for v.
2. Produce a spectral plot like that on Slide 14.
3. Compare the in-band quantization noise of your system with a half-scale sine-wave input against the relation given on Slide 12 for OSR in $[2^3, 2^{10}]$.

*. $\overline{v_{tri}} = \sum_{n=0}^{N-1} tri(n)v(n)$, $tri = rect_{(N/2)} * rect_{(N/2)}$

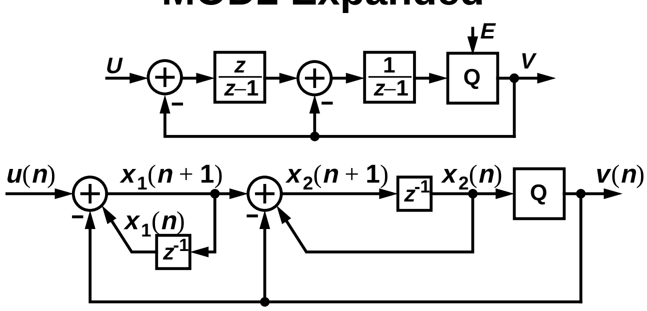
Homework #2– Part B

Extract code from `dsdemo1` & `dsdemo2` and modify it to:

1. Create a 3rd-order NTF with zeros optimized for OSR = 32. Plot the poles/zeros and frequency response of your NTF.
2. Simulate a 3-level ΔΣ modulator with this NTF. Plot example input and output waveforms. Plot a spectrum and the predicted noise curve.† Plot the SQNR vs. input amplitude curve and note the maximum stable input.

†. Beware that with an M-step modulator, the full-scale is M and this must be accounted for in normalizing the FFT and in computing the expected noise curve. You ought to check your code using a large value for M, say $M = 20$.

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

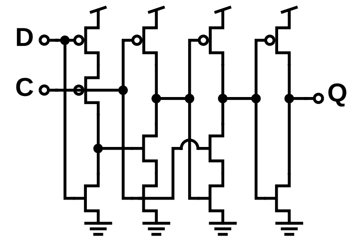
```
function [v] = simulateMOD2(u)
    x1 = 0;
    x2 = 0;
    for i = 1:length(u)
        v(i) = quantize( x2 );
        x1 = x1 + u(i) - v(i);
        x2 = x2 + x1 - v(i);
    end
return

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

What You Learned Today

And what the homework should solidify

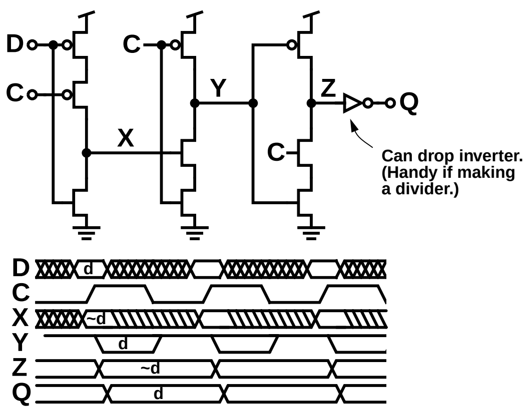
1. Second-order modulator (MOD2)
2. Arbitrary-order modulator (MOD*N*) design with the $\Delta\Sigma$ Toolbox

NLCOTD: True Single-Phase Dynamic FF



- \+ Clock not inverted anywhere
- \+ Small
- \+ Fast

TSPFF Operation



TSPFF Gotchas

- Leakage:
  Won't work if clock is too slow.
  Possible high current if clock is stopped.
  - Need to add devices that hold the dynamic nodes at a safe value.
- No positive feedback
  - Vulnerable to metastability.