Oversampling Converters

David Johns and Ken Martin
University of Toronto
(johns@eecg.toronto.edu)
(martin@eecg.toronto.edu)

Motivation

- Popular approach for medium-to-low speed A/D and D/A applications requiring high resolution

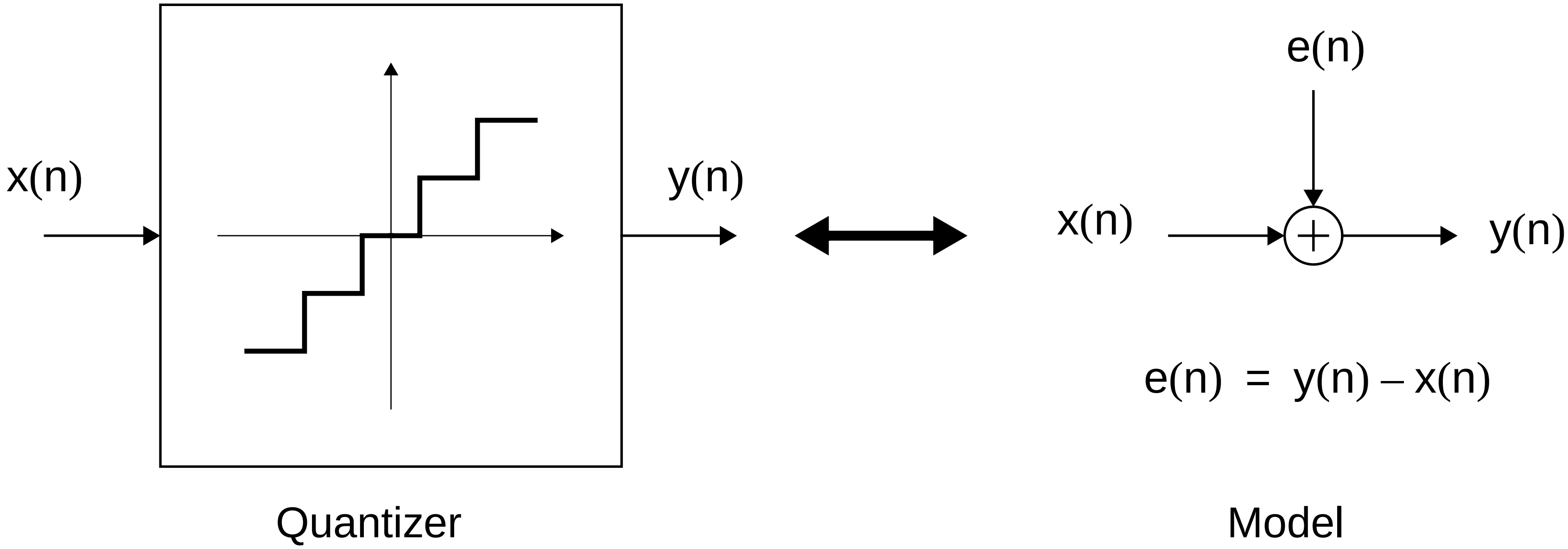
Easier Analog

- reduced matching tolerances
- relaxed anti-aliasing specs
- relaxed smoothing filters

More Digital Signal Processing

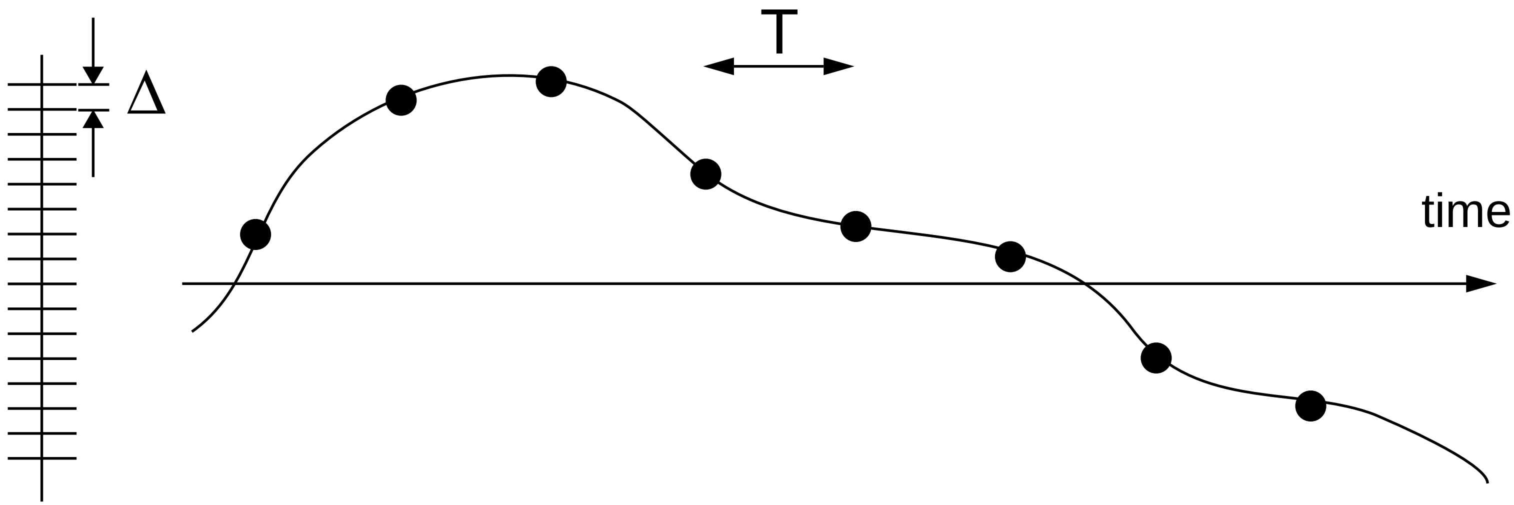
- Needs to perform strict anti-aliasing or smoothing filtering
- Also removes shaped quantization noise and decimation (or interpolation)

Quantization Noise



- Above model is exact
  — approx made when assumptions made about $e(n)$
- Often assume $e(n)$ is white, uniformily distributed number between $\pm\Delta/2$
- Δ is difference between two quantization levels

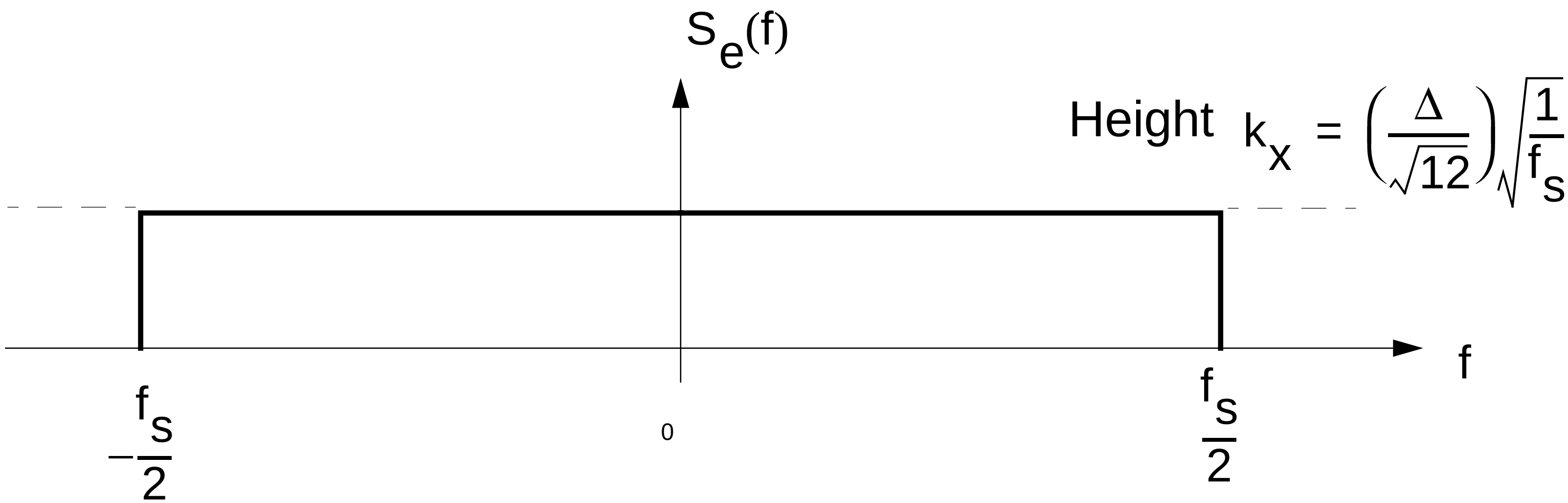
Quantization Noise



- White noise assumption reasonable when:
  — fine quantization levels
  — signal crosses through many levels between samples
  — sampling rate not synchronized to signal frequency
- Sample lands somewhere in quantization interval leading to random error of $\pm\Delta/2$

<u>Quantization Noise</u>

- Quantization noise power shown to be $\Delta^2/12$ and is ***independent of sampling frequency***
- If white, then spectral density of noise, $S_e(f)$, is constant.

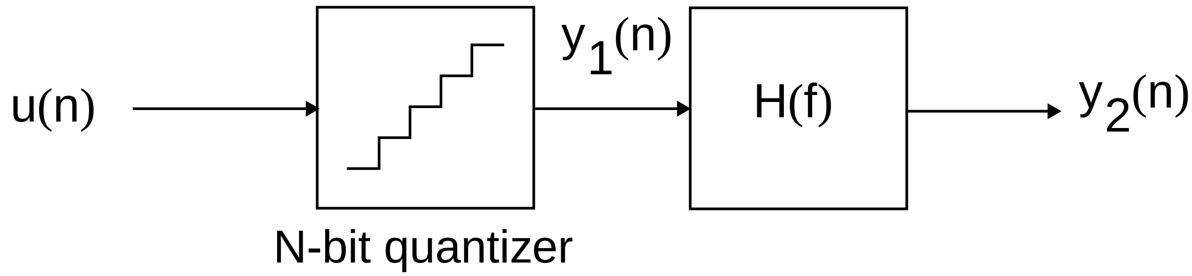


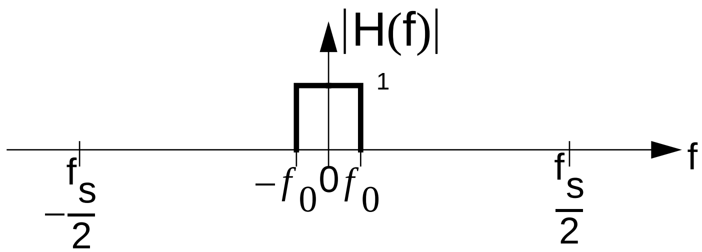
Oversampling Advantage

- Oversampling occurs when signal of interest is bandlimited to f_0 but we sample higher than $2f_0$
- Define oversampling-rate

$$\mathrm{OSR} = f_s/(2f_0) \qquad (1)$$

- After quantizing input signal, pass it through a brickwall digital filter with passband up to f_0





Oversampling Advantage

- Output quantization noise after filtering is:

$$P_e = \int_{-f_s/2}^{f_s/2} S_e^2(f)|H(f)|^2\, df = \int_{-f_0}^{f_0} k_x^2\, df = \frac{\Delta^2}{12}\left(\frac{1}{OSR}\right) \quad (2)$$

- Doubling OSR reduces quantation noise power by 3dB (i.e. 0.5 bits/octave)
- Assuming peak input is a sinusoidal wave with a peak value of $2^N(\Delta/2)$ leading to $P_s = ((\Delta 2^N)/(2\sqrt{2}))^2$
- Can also find peak SNR as:

$$SNR_{max} = 10\,\log\left(\frac{P_s}{P_e}\right) = 10\,\log\left(\frac{3}{2}2^{2N}\right) + 10\log(OSR) \quad (3)$$

Oversampling Advantage

Example

- A dc signal with 1V is combined with a noise signal uniformily distributed between $\pm\sqrt{3}$ giving 0 dB SNR.
— {0.94, –0.52, –0.73, 2.15, 1.91, 1.33, –0.31, 2.33}.
- Average of 8 samples results in 0.8875
- Signal adds linearly while noise values add in a square-root fashion — noise filtered out.

Example

- 1-bit A/D gives 6dB SNR.
- To obtain 96dB SNR requires 30 octaves of oversampling ((96-6)/3 dB/octave)
- If $f_0 = 25\text{ kHz}$, $f_s = 2^{30} \times f_0 = 54,000\text{ GHz}$!

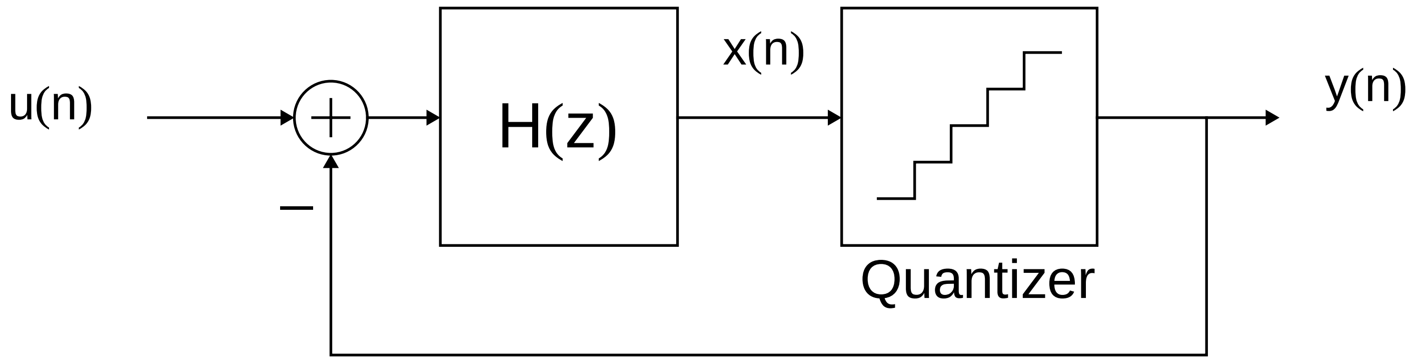
Advantage of 1-bit D/A Converters

- Oversampling improves SNR but not linearity
- To acheive 16-bit linear converter using a 12-bit converter, 12-bit converter must be linear to 16 bits

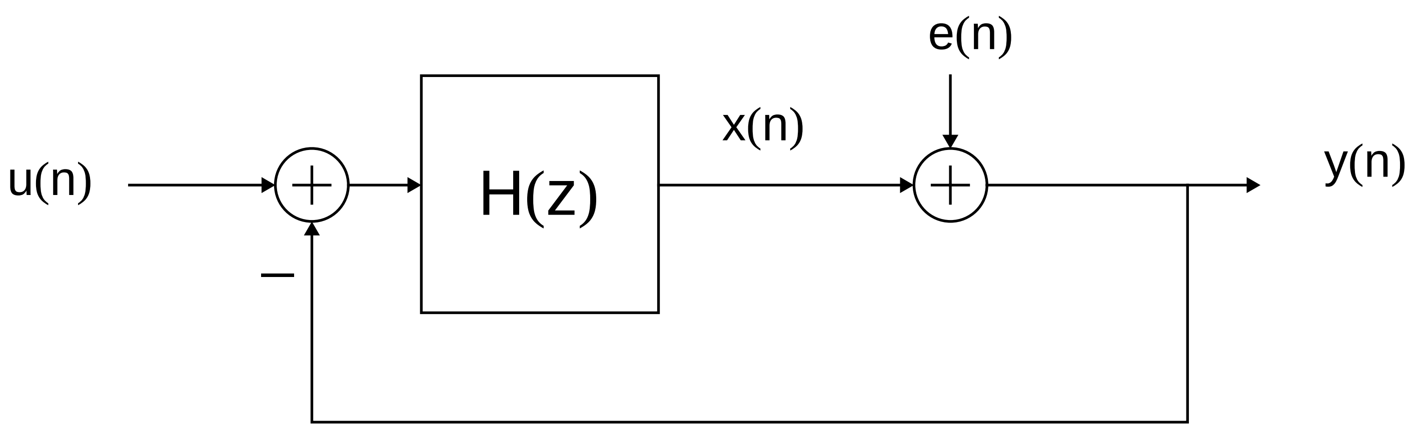
  — i.e. integral nonlinearity better than $1/2^4$ LSB
- A 1-bit D/A is ***inherently linear***
  — 1-bit D/A has only 2 output points
  — 2 points always lie on a straight line
- Can acheive better than 20 bits linearity without trimming (will likely have gain and offset error)
- Second-order effects (such as D/A memory or signal-dependent reference voltages) will limit linearity.

Oversampling with Noise Shaping

- Place the quantizer in a feedback loop



Delta-Sigma Modulator



Linear model

Oversampling with Noise Shaping

- Shapes quantization noise away from signal band of interest

Signal and Noise Transfer-Functions

$$S_{TF}(z) \equiv \frac{Y(z)}{U(z)} = \frac{H(z)}{1 + H(z)} \tag{4}$$

$$N_{TF}(z) \equiv \frac{Y(z)}{E(z)} = \frac{1}{1 + H(z)} \tag{5}$$

$$Y(z) = S_{TF}(z)U(z) + N_{TF}(z)E(z) \tag{6}$$

- Choose $H(z)$ to be large over 0 to f_0
- Resulting quantization noise near 0 where $H(z)$ large
- Signal transfer-function near 1 where $H(z)$ large

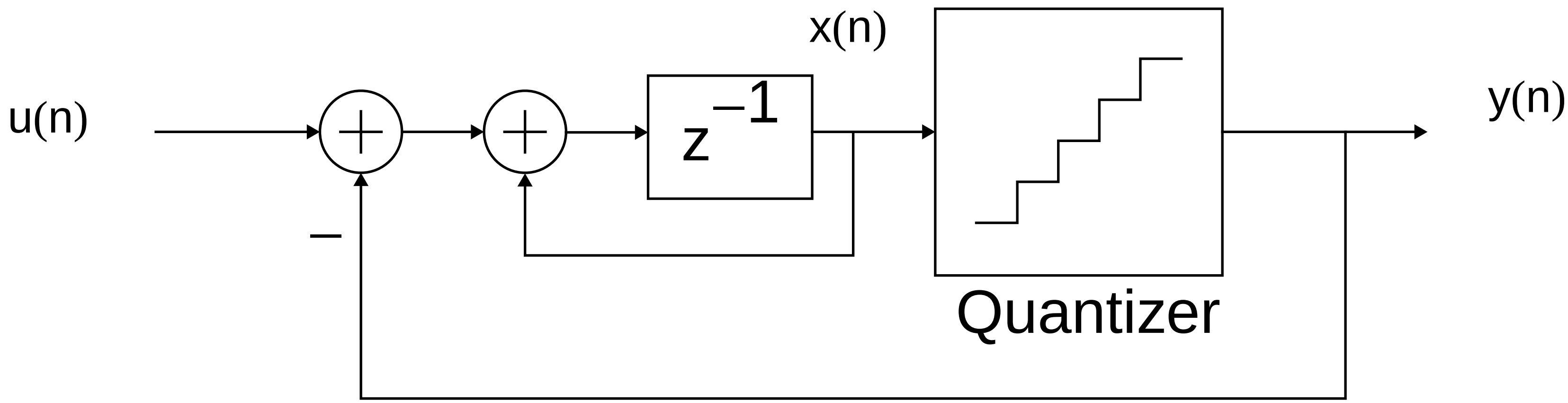
Oversampling with Noise Shaping

- Input signal is limited to range of quantizer output when $H(z)$ large
- For 1-bit quantizers, input often limited to 1/4 quantizer outputs
- Out-of-band signals can be larger when $H(z)$ small
- Stability of modulator can be an issue (particularily for higher-orders of $H(z)$)
- Stability defined as when input to quantizer becomes so large that quantization error greater than $\pm\Delta/2$ — said to “overload the quantizer”

First-Order Noise Shaping

- Choose $H(z)$ to be a discrete-time integrator

$$H(z) = \frac{1}{z-1} \tag{7}$$



- If stable, average input of integrator must be zero
- Average value of $u(n)$ must equal average of $y(n)$

Example

- The output sequence and state values when a dc input, $u(n)$, of $1/3$ is applied to a 1'st order modulator with a two-level quantizer of ±1.0. Initial state for $x(n)$ is 0.1.

n	x(n)	x(n + 1)	y(n)	e(n)
0	0.1	–0.5667	1.0	0.9
1	–0.5667	0.7667	–1.0	–0.4333
2	0.7667	0.1	1.0	0.2333
3	0.1	–0.5667	1.0	0.9
4	–0.5667	0.7667	–1.0	–0.4333
5	…	…	…	…

- Average of $y(n)$ is $1/3$ as expected
- Periodic quantization noise in this case

Transfer-Functions

Signal and Noise Transfer-Functions

$$S_{TF}(z) = \frac{Y(z)}{U(z)} = \frac{1/(z-1)}{1 + 1/(z-1)} = z^{-1} \tag{8}$$

$$N_{TF}(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + 1/(z-1)} = (1 - z^{-1}) \tag{9}$$

- Noise transfer-function is a discrete-time differentiator (i.e. a highpass filter)

$$\begin{aligned} N_{TF}(f) &= 1 - e^{-j2\pi f/f_s} = \frac{e^{j\pi f/f_s} - e^{-j\pi f/f_s}}{2j} \times 2j \times e^{-j\pi f/f_s} \\ &= \sin\left(\frac{\pi f}{f_s}\right) \times 2j \times e^{-j\pi f/f_s} \end{aligned} \tag{10}$$

Signal to Noise Ratio

Magnitude of noise transfer-function

$$|N_{TF}(f)| = 2\sin\left(\frac{\pi f}{f_s}\right) \qquad (11)$$

Quantization noise power

$$P_e = \int_{-f_0}^{f_0} S_e^2(f)|N_{TF}(f)|^2 df = \int_{-f_0}^{f_0} \left(\frac{\Delta^2}{12}\right)\frac{1}{f_s}\left[2\sin\left(\frac{\pi f}{f_s}\right)\right]^2 df \qquad (12)$$

- Assuming $f_0 << f_s$ (i.e., $OSR >> 1$)

$$P_e \cong \left(\frac{\Delta^2}{12}\right)\left(\frac{\pi^2}{3}\right)\left(\frac{2f_0}{f_s}\right)^3 = \frac{\Delta^2\pi^2}{36}\left(\frac{1}{OSR}\right)^3 \qquad (13)$$

Max SNR

- Assuming peak input is a sinusoidal wave with a peak value of $2^N(\Delta/2)$ leading to $P_s = ((\Delta 2^N)/(2\sqrt{2}))^2$
- Can find peak SNR as:

$$\mathrm{SNR}_{max} = 10\log\left(\frac{P_s}{P_e}\right)$$

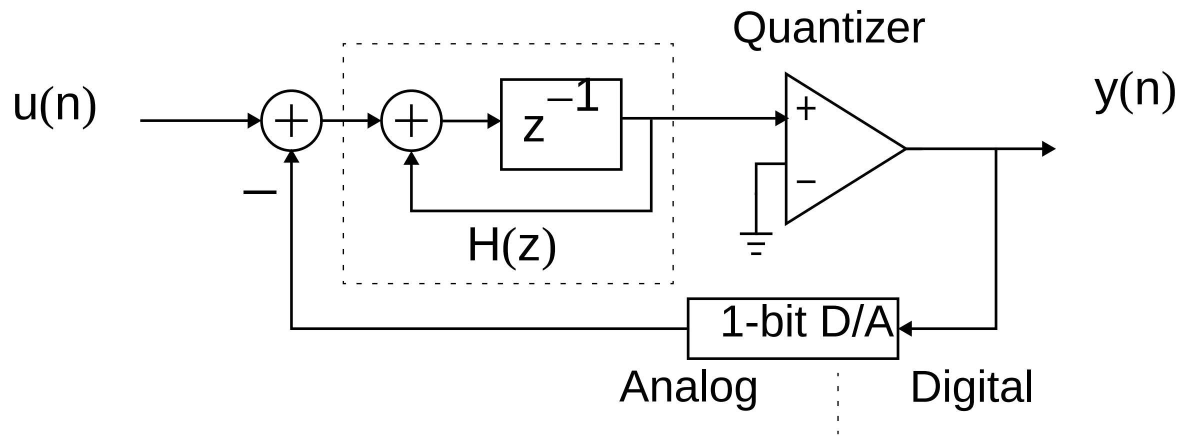
$$= 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log\left[\frac{3}{\pi^2}(OSR)^3\right] \tag{14}$$

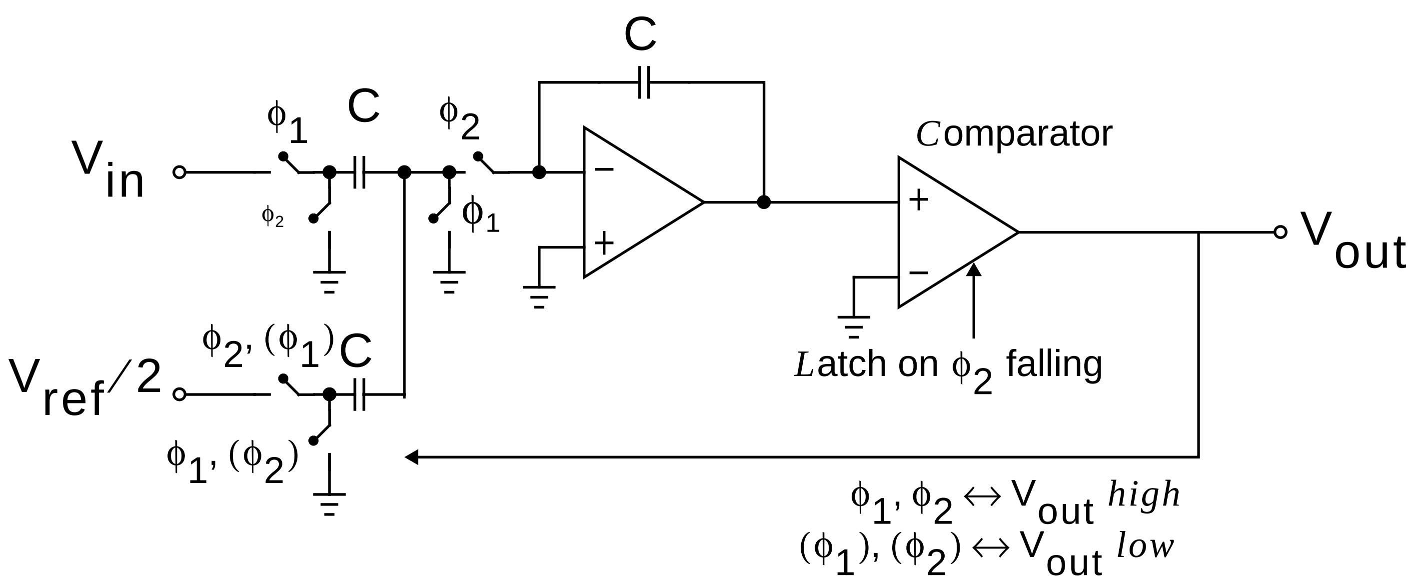
or, equivalently,

$$\mathrm{SNR}_{max} = 6.02N + 1.76 - 5.17 + 30\log(OSR) \tag{15}$$

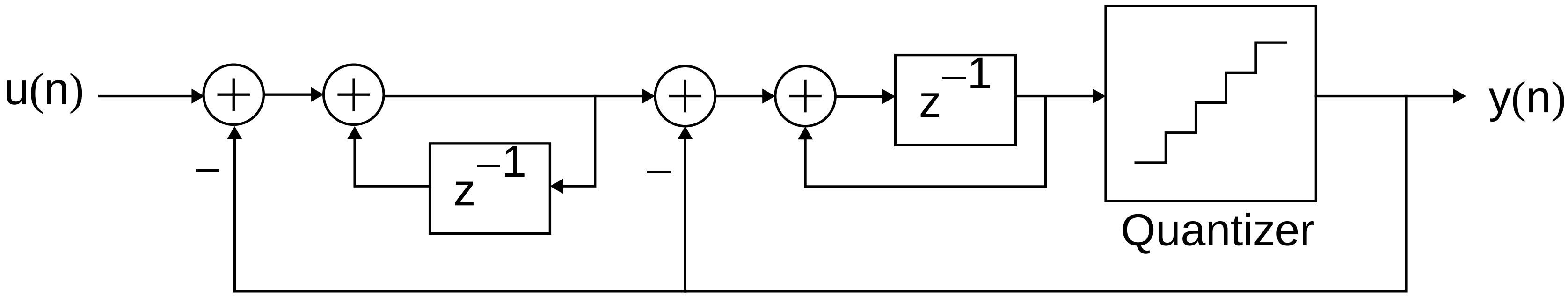
- Doubling OSR gives an SNR improvement 9 dB or, equivalently, a benefit of 1.5 bits/octave

SC Implementation





Second-Order Noise Shaping



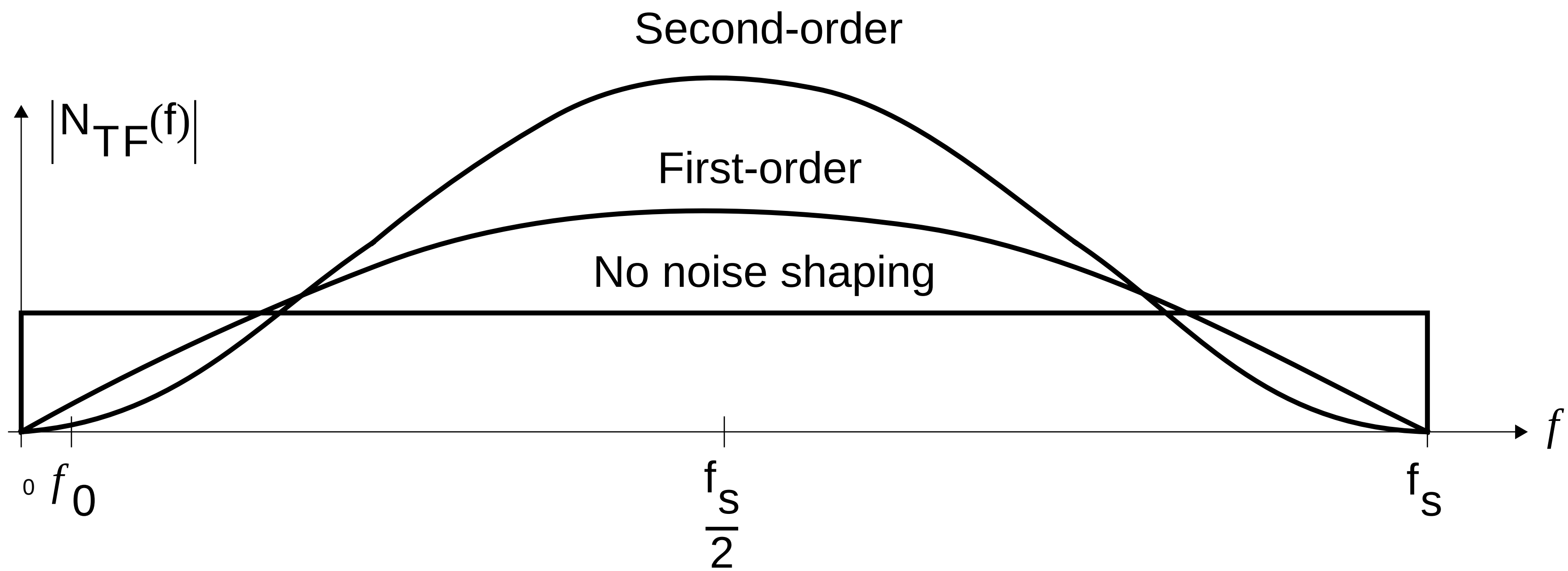
$$S_{TF}(f) = z^{-1} \tag{16}$$

$$N_{TF}(f) = (1 - z^{-1})^2 \tag{17}$$

$$SNR_{max} = 6.02N + 1.76 - 12.9 + 50\log(OSR) \tag{18}$$

- Doubling OSR improves SNR by 15 dB (i.e., a benefit of 2.5 bits/octave)

Noise Transfer-Function Curves



- Out-of-band noise increases for high-order modulators
- Out-of-band noise peak controlled by poles of noise transfer-function
- Can also spread zeros over band-of-interest

Example

- 90 dB SNR improvement from A/D with $f_0 = 25\ \mathrm{kHz}$

Oversampling with no noise shaping

- From before, straight oversampling requires a sampling rate of $54{,}000$ GHz.

First-Order Noise Shaping

- Lose 5 dB (see (15)), require 95 dB divided by 9 dB/octave, or 10.56 octaves — $f_s = 2^{10.56} \times 2 f_0 \cong 75$ MHz

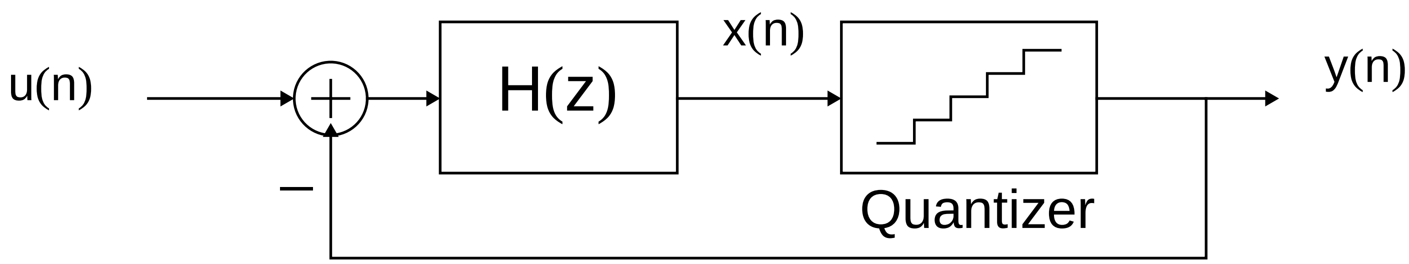
Second-Order Noise Shaping

- Lose 13 dB, required 103 dB divided by 15 dB/octave, $f_s = 5.8\ \mathrm{MHz}$ (does not account for reduced input range needed for stability).

Quantization Noise Power of 1-bit Modulators

- If output of 1-bit mod is ± 1, total power of output signal, $y(n)$, is normalized power of 1 watt.
- Signal level often limited to well below ± 1 level in higher-order modulators to maintain stability
- For example, if maximum peak level is ± 0.25, max signal power is 62.5 mW.
- Max signal is approx 12 dB below quantization noise (but most noise in different frequency region)
- Quantization filter must have dynamic range capable of handling full power of $y(n)$ at input.
- Easy for A/D — digital filter
- More difficult for D/A — analog filter

Zeros of NTF are poles of H(z)



- Write $H(z)$ as

$$H(z) = \frac{N(z)}{D(z)} \tag{19}$$

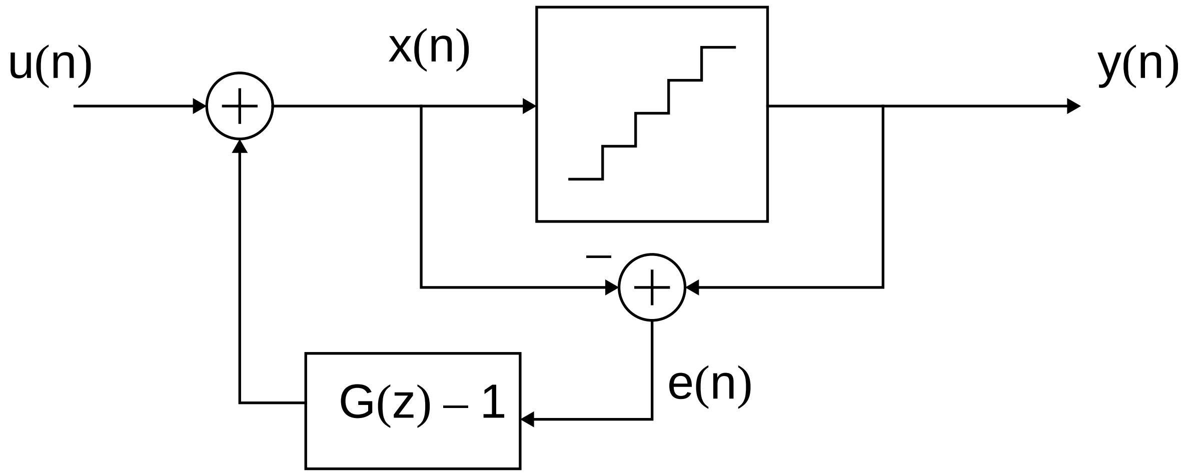
- NTF is given by:

$$\mathrm{NTF}(z) = \frac{1}{1 + H(z)} = \frac{D(z)}{D(z) + N(z)} \tag{20}$$

- If poles of $H(z)$ are well-defined then so are zeros of NTF

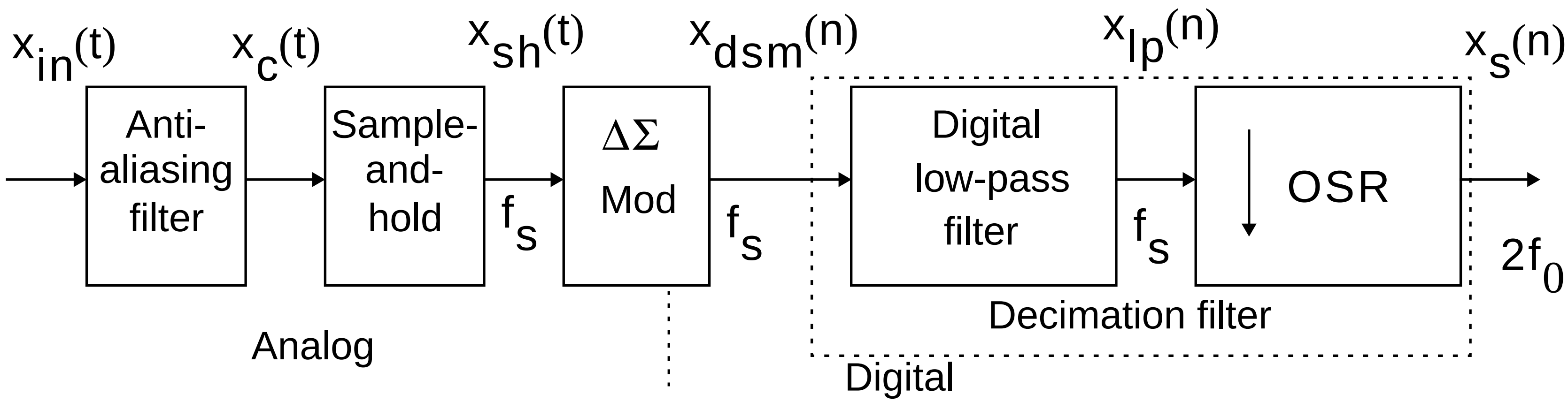
Error-Feedback Structure

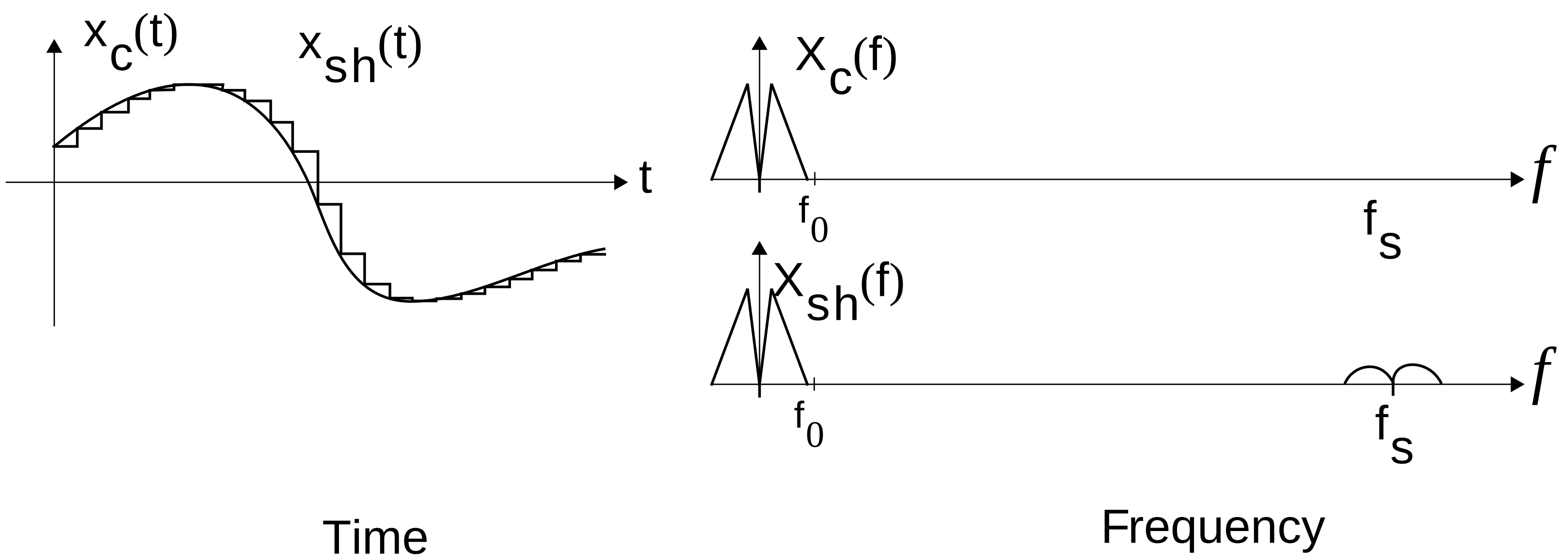
- Alternate structure to interpolative



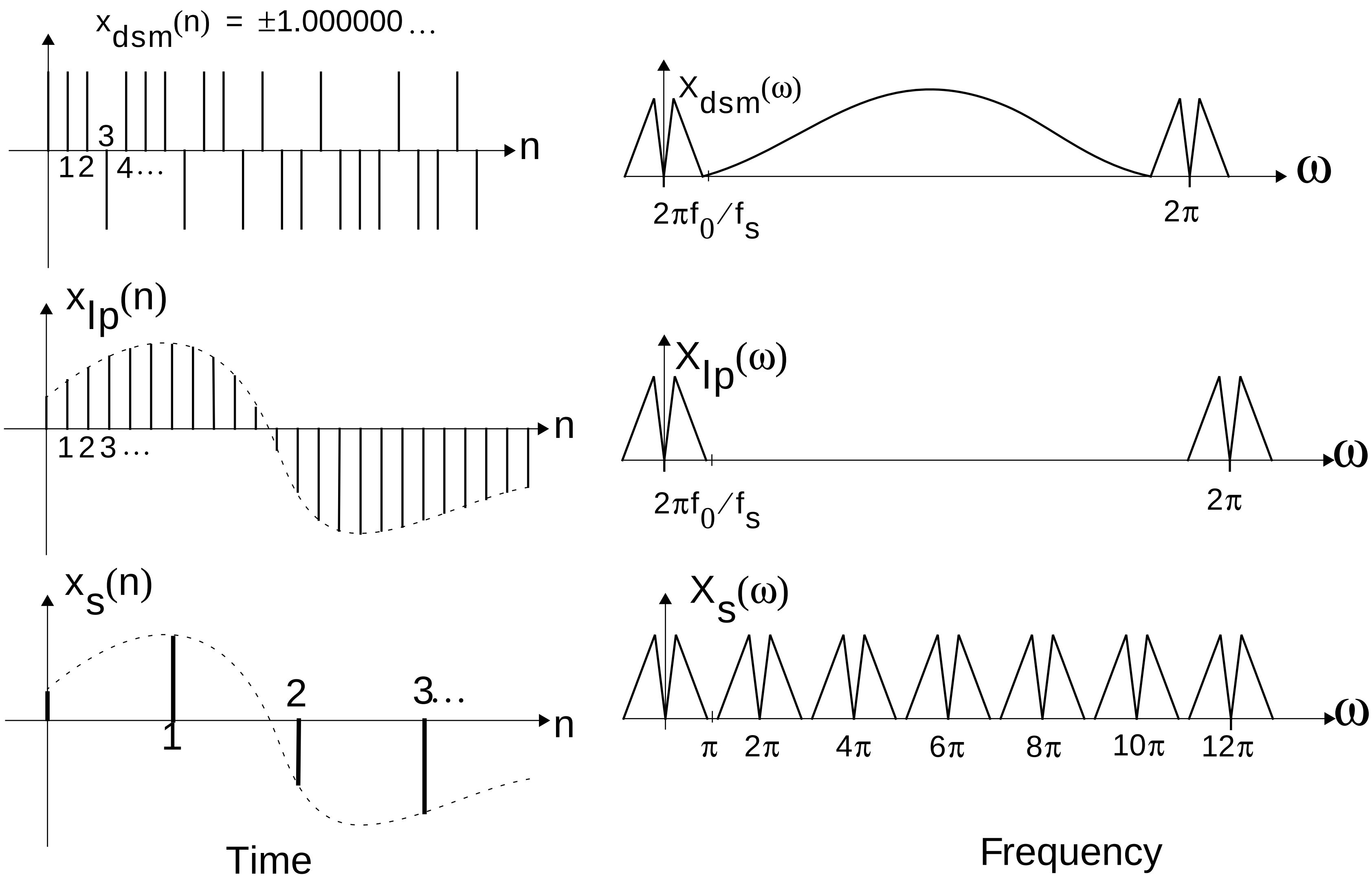
- Signal transfer-function equals unity while noise transfer-function equals $G(z)$
- First element of $G(z)$ equals 1 for no delay free loops
- First-order system — $G(z) - 1 = -z^{-1}$
- More sensitive to coefficient mismatches

Architecture of Delta-Sigma A/D Converters





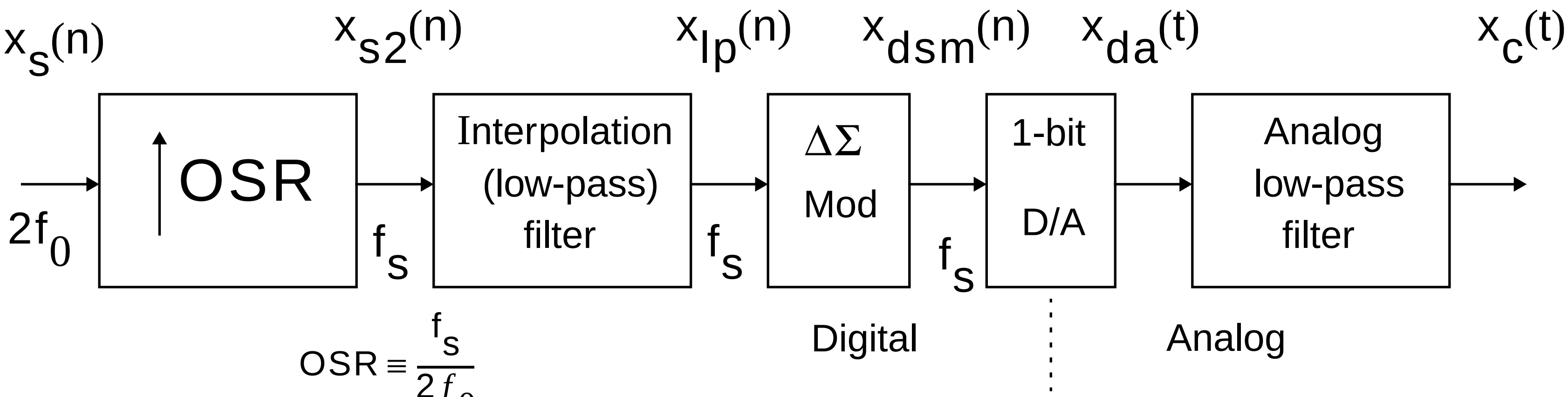
Architecture of Delta-Sigma A/D Converters



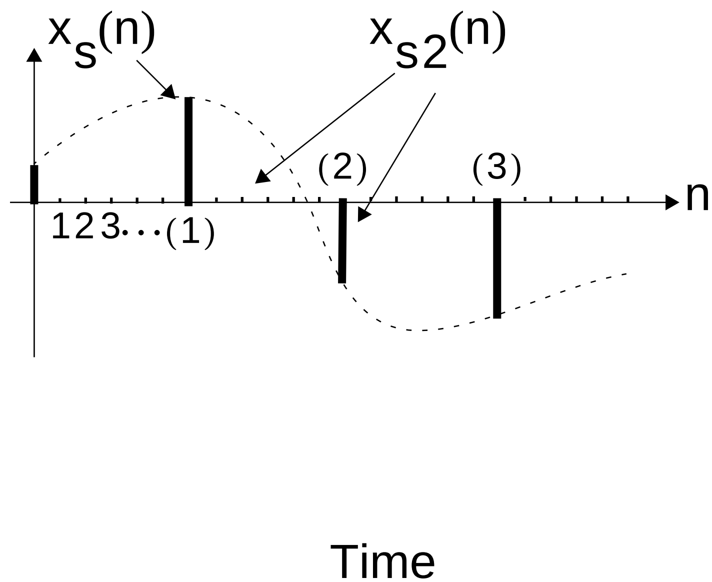
Architecture of Delta-Sigma A/D Converters

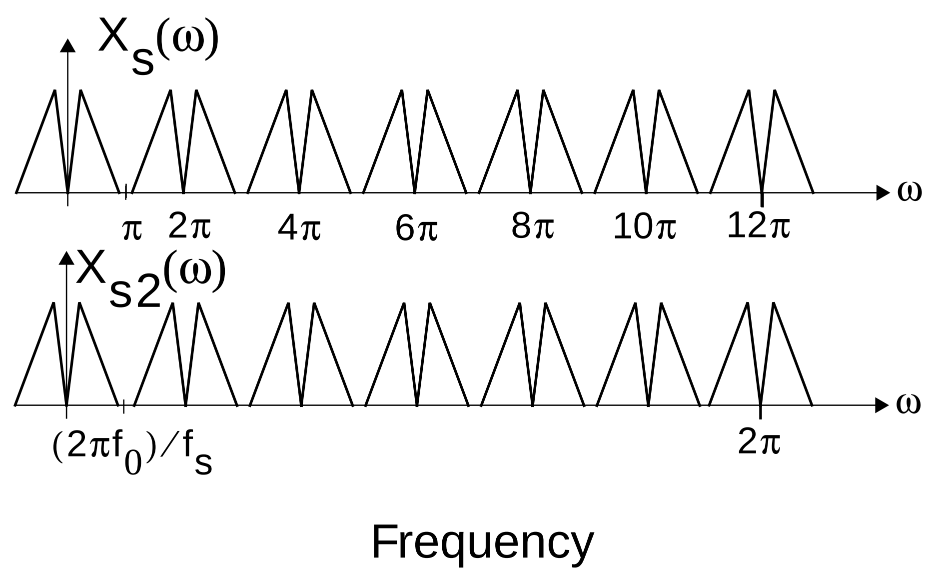
- Relaxes analog anti-aliasing filter
- Strict anti-aliasing done in digital domain
- Must also remove quantization noise before downsampling (or aliasing occurs)
- Commonly done with a multi-stage system
- Linearity of D/A in modulator important — results in overall nonlinearity
- Linearity of A/D in modulator unimportant (effects reduced by high gain in feedback of modulator)

Architecture of Delta-Sigma D/A Converters

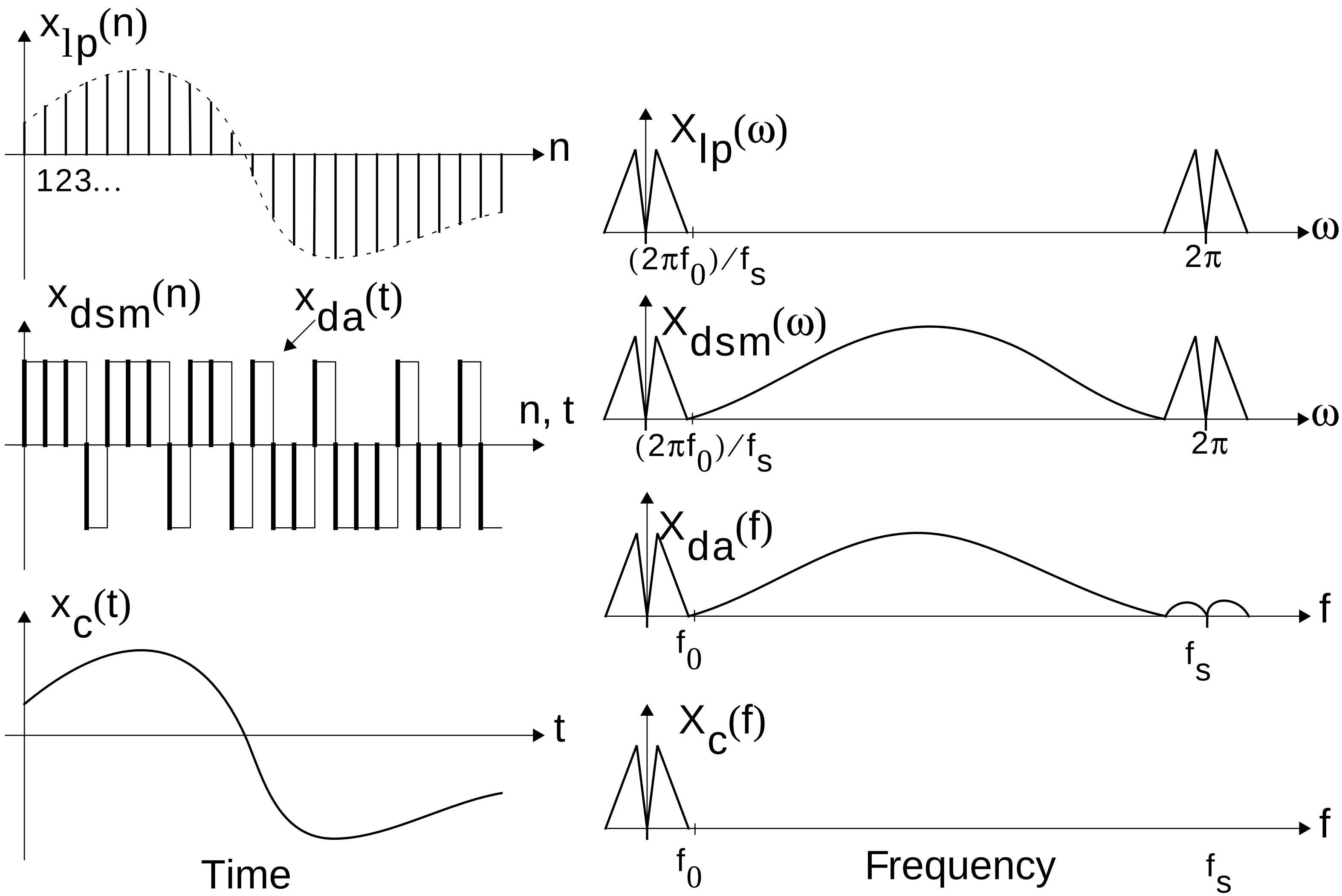


$$\mathrm{OSR} \equiv \frac{f_s}{2f_0}$$





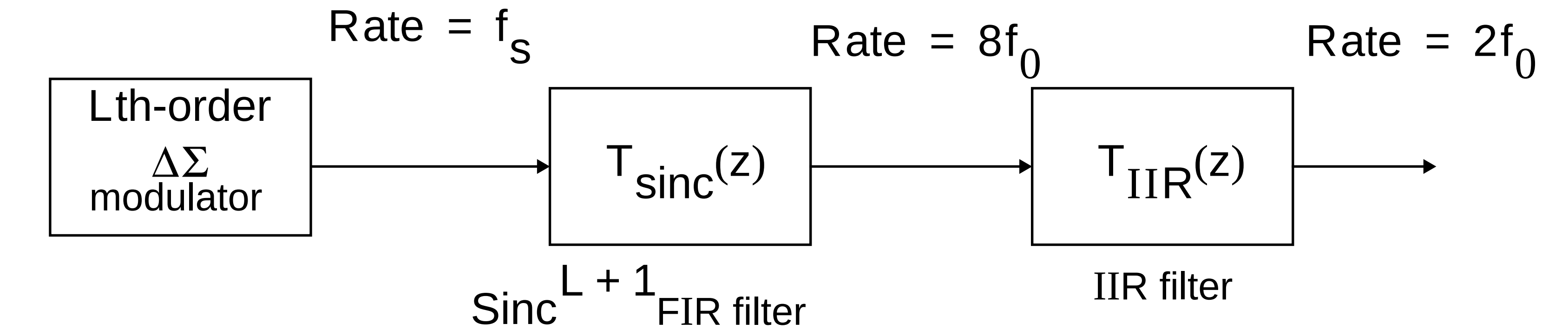
Architecture of Delta-Sigma D/A Converters

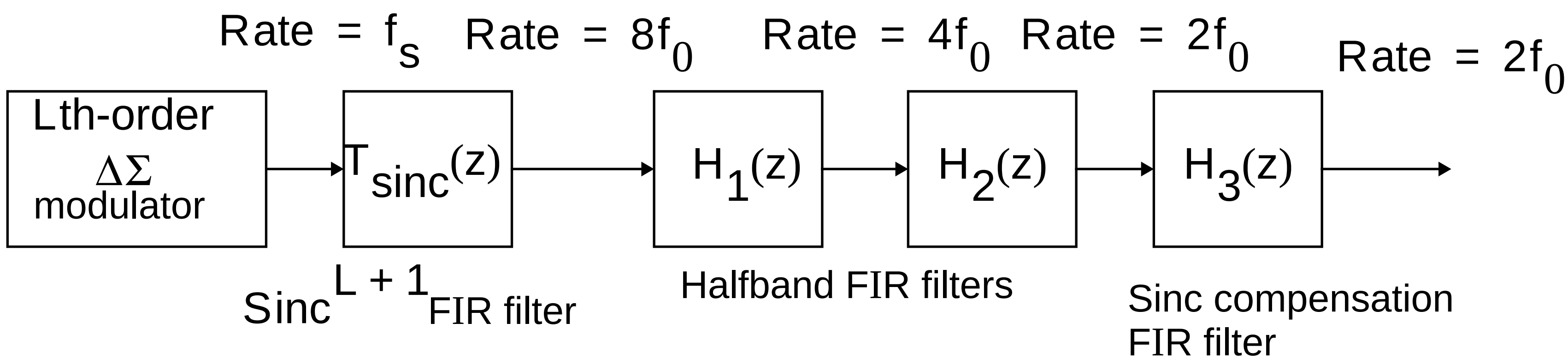


Architecture of Delta-Sigma D/A Converters

- Relaxes analog smoothing filter (many multibit D/A converters are oversampled without noise shaping)
- Smoothing filter of first few images done in digital (then often below quantization noise)
- Order of lowpass filter should be at least one order higher than that of modulator
- Results in noise dropping off (rather than flat)
- Analog filter must attenuate quantization noise and should not modulate noise back to low freq — strong motivation to use multibit quantizers

Multi-Stage Digital Decimation





- Sinc filter removes much of quantization noise
- Following filter(s) — anti-aliasing filter and noise

Sinc Filter

- sinc^{L+1} is a cascade of $L+1$ averaging filters

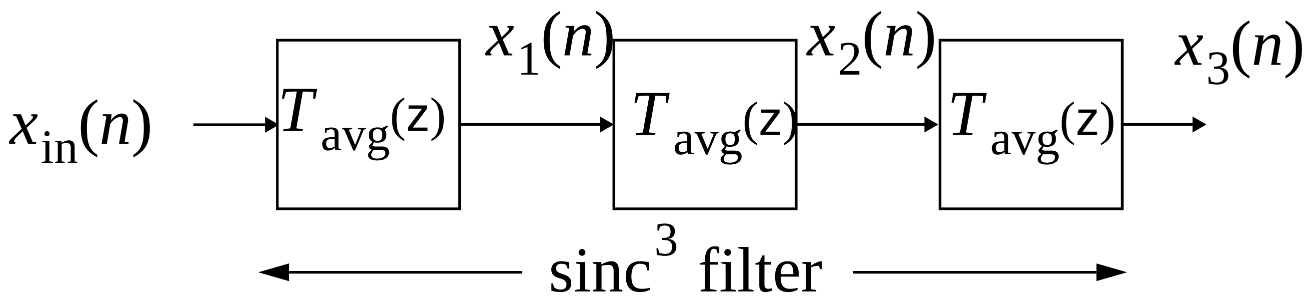
Averaging filter

$$T_{avg}(z) = \frac{Y(z)}{U(z)} = \frac{1}{M}\sum_{i=0}^{M-1} z^{-i} \tag{21}$$

- M is integer ratio of $f_s/(8f_0)$
- It is a linear-phase filter (symmetric coefficients)
- If M is power of 2, easy division (shift left)
- Can not do all decimation filtering here since not sharp enough cutoff

Sinc Filter

- Consider $x_{in}(n) = \{1, 1, -1, 1, 1, -1, \ldots\}$ applied to $M = 4$ averaging filters in cascade



- $x_1(n) = \{0.5, 0.5, 0.0, 0.5, 0.5, 0.0, \ldots\}$
- $x_2(n) = \{0.38, 0.38, 0.25, 0.38, 0.38, 0.25, \ldots\}$
- $x_3(n) = \{0.34, 0.34, 0.31, 0.34, 0.34, 0.31, \ldots\}$
- Converging to sequence of all 1/3 as expected

Sinc Filter Response

- Can rewrite averaging filter in recursive form as

$$T_{avg}(z) = \frac{Y(z)}{U(z)} = \frac{1}{M}\left(\frac{1-z^{-M}}{1-z^{-1}}\right) \tag{22}$$

and a cascade of $L+1$ averaging filters results in

$$T_{\text{sinc}}(z) = \frac{1}{M^{L+1}}\left(\frac{1-z^{-M}}{1-z^{-1}}\right)^{L+1} \tag{23}$$

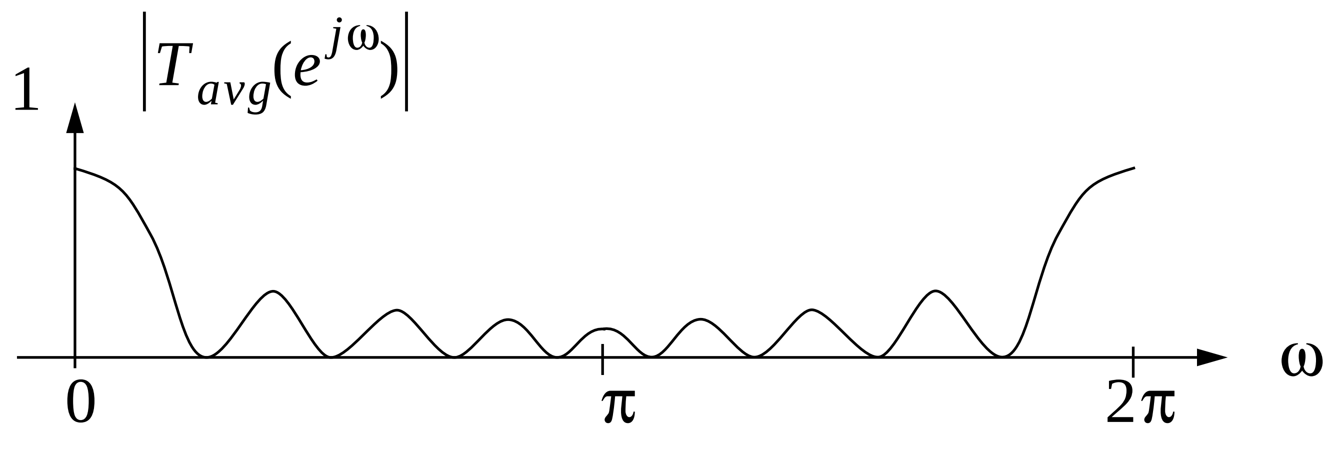
- Use $L+1$ cascade to roll off quantization noise faster than it rises in L'th order modulator

Sinc Filter Frequency Response

- Let $z = e^{j\omega}$

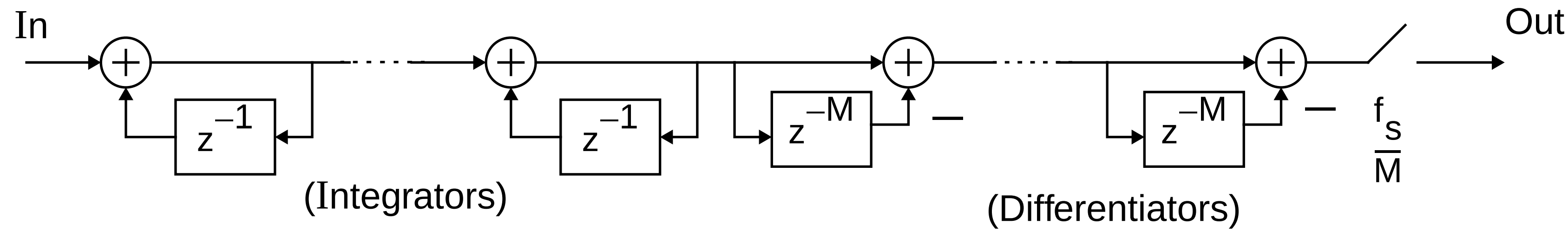
$$T_{avg}(e^{j\omega}) = \frac{\operatorname{sinc}\left(\frac{\omega M}{2}\right)}{\operatorname{sinc}\left(\frac{\omega}{2}\right)} \tag{24}$$

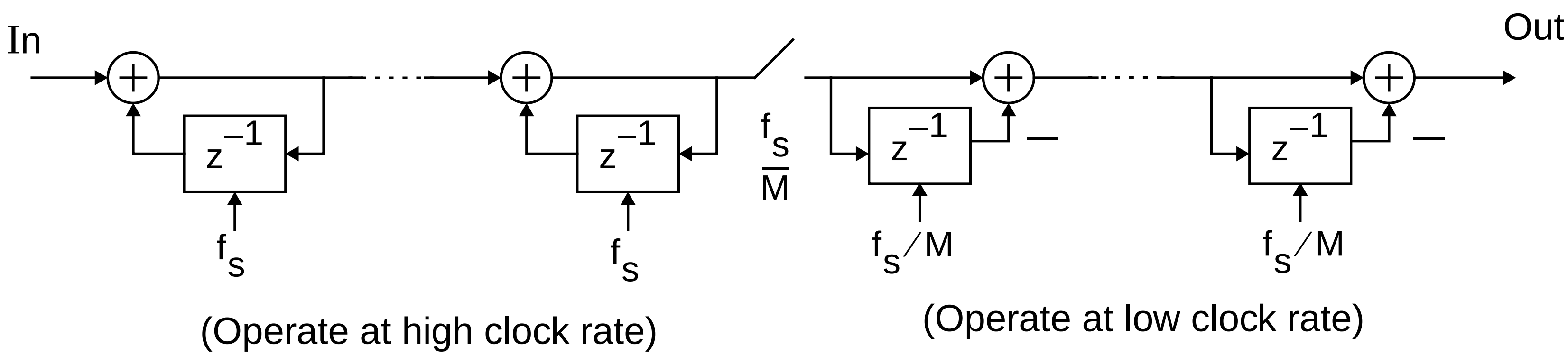
where $\operatorname{sinc}(x) \equiv \sin(x)/x$



Sinc Implementation

$$T_{\text{sinc}}(z) = \left(\frac{1}{1 - z^{-1}}\right)^{L+1} (1 - z^{M})^{L+1} \frac{1}{M^{L+1}} \qquad (25)$$



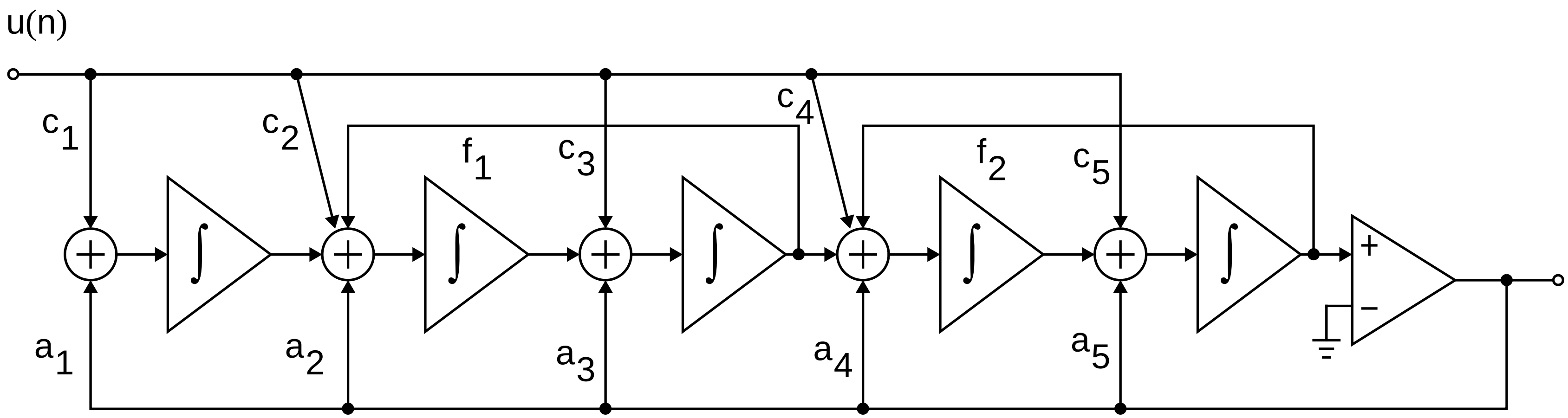


- If 2’s complement arithmetic used, wrap-around okay since followed by differentiators

Higher-Order Modulators

- An L'th order modulator improves SNR by 6L+3 dB/octave

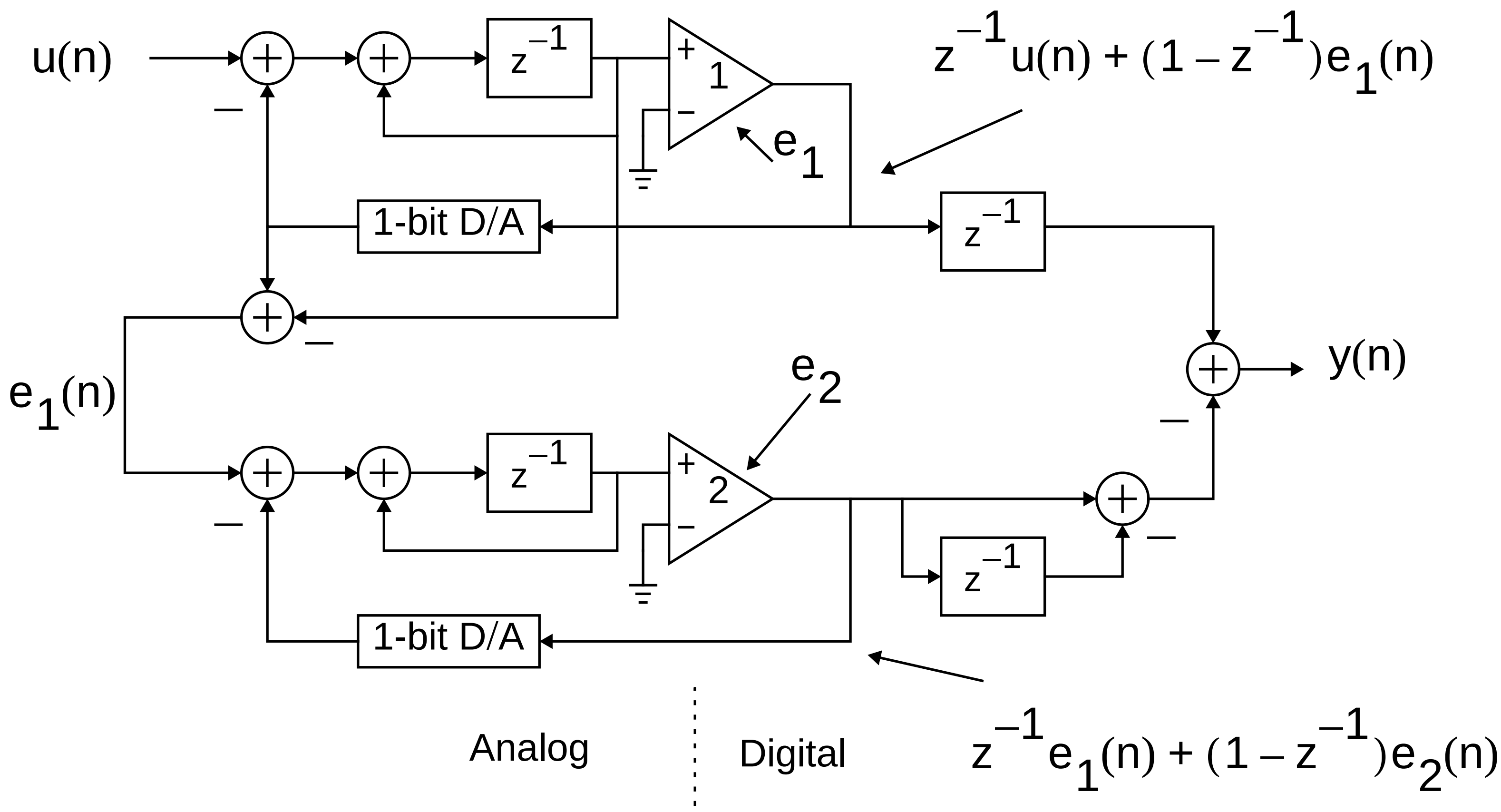
Interpolative Architecture



- Can spread zeros over freq of interest using resonators with f_1 and f_2
- Need to worry about stability (more later)

MASH Architecture

- Multi-stAge noise SHaping - MASH
- Use multiple lower order modulators and combine outputs to cancel noise of first stages



MASH Architecture

- Output found to be:

$$Y(z) = z^{-2}U(z) - (1 - z^{-1})^2 E_2(z) \tag{26}$$

Multibit Output

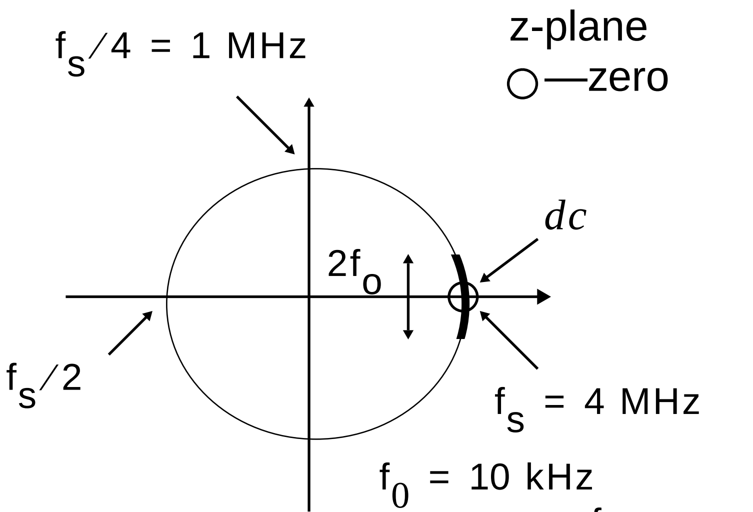
- Output is a 4-level signal though only single-bit D/A's
  — if D/A application, then linear 4-level D/A needed
  — if A/D, slightly more complex decimation

A/D Application

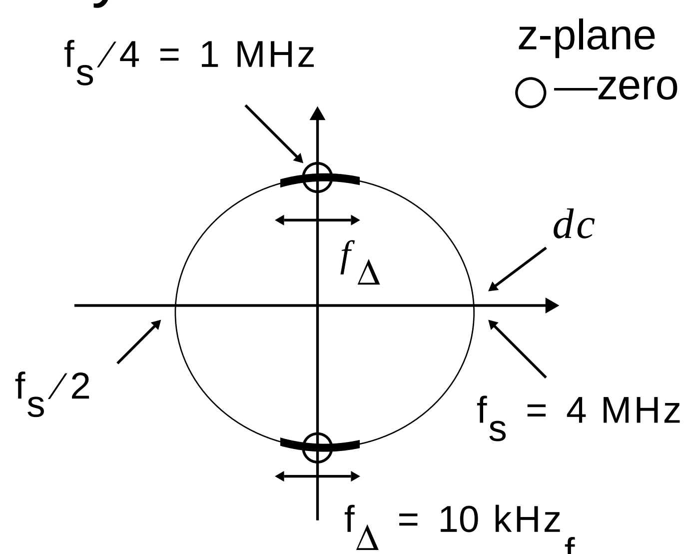
- Mismatch between analog and digital can cause first-order noise, e_1, to leak through to output
- Choose first stage as higher-order (say 2'nd order)

Bandpass Oversampling Converters

- Choose $H(z)$ to have high gain near freq f_c
- NTF shapes quantization noise to be small near f_c
- OSR is ratio of sampling-rate to twice bandwidth — not related to center frequency

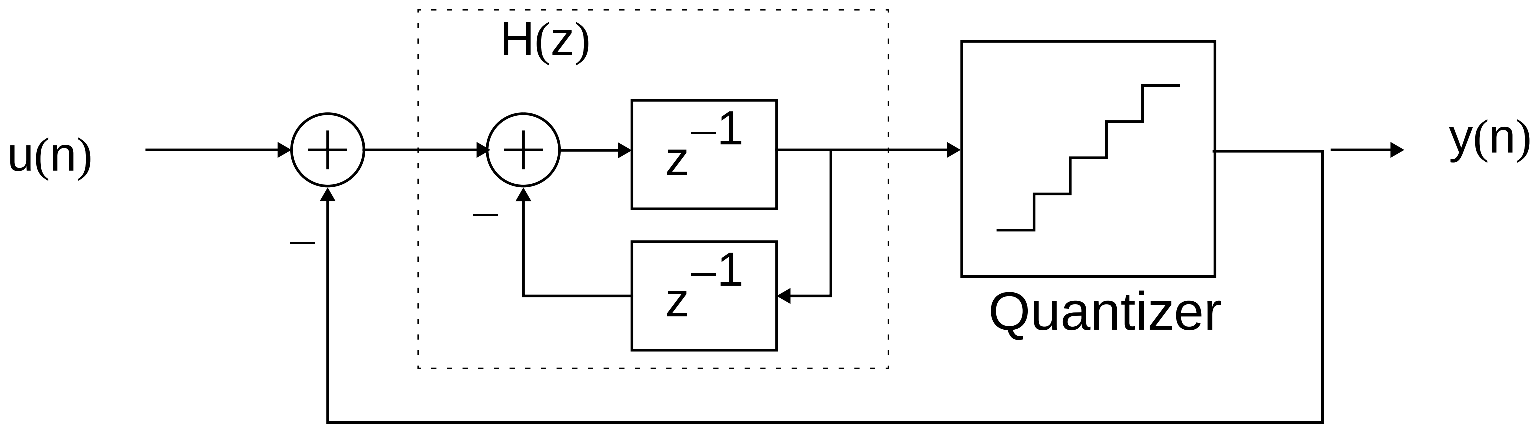


Lowpass $\mathrm{OSR} = \dfrac{f_s}{2f_0} = 200$



Bandpass $\mathrm{OSR} = \dfrac{f_s}{2f_\Delta} = 200$

Bandpass Oversampling Converters



- Above $H(z)$ has poles at $\pm j$ (which are zeros of NTF)
  — $H(z)$ is a resonator with infinite gain at $f_s/4$
  — $H(z) = z/(z^2+1)$
- Note one zero at +j and one zero at -j
  — similar to lowpass first-order modulator
  — only 9 dB/octave
- For 15 dB/octave, need 4'th order BP modulator

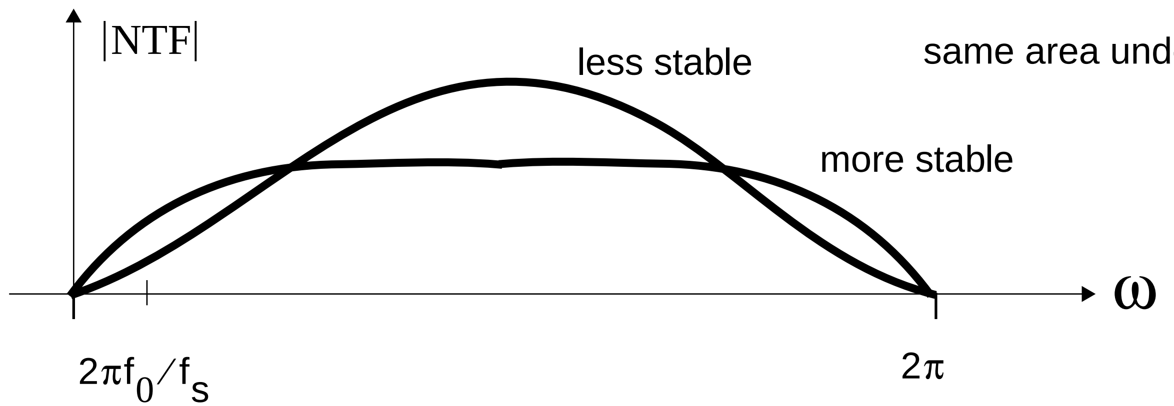
Modulator Stability

- Since feedback involved, stability is an issue
- Considered stable if quantizer input does not ***overload*** quantizer
- Non-trivial to analyze due to quantizer
- There are rigorous tests to guarantee stability but they are too conservative
- For a 1-bit quantizer, heuristic test is:

$$\left|N_{TF}(e^{j\omega})\right| \le 1.5 \qquad \text{for } 0 \le \omega \le \pi \tag{27}$$

- Peak of NTF should be less than 1.5
- Can be made more stable by placing poles of NTF closer to its zeros
- Dynamic range suffers since less noise power pushed out-of-band

Modulator Stability



Stability Detection

- Might look at input to quantizer
- Might look for long strings of 1s or 0s at comp output

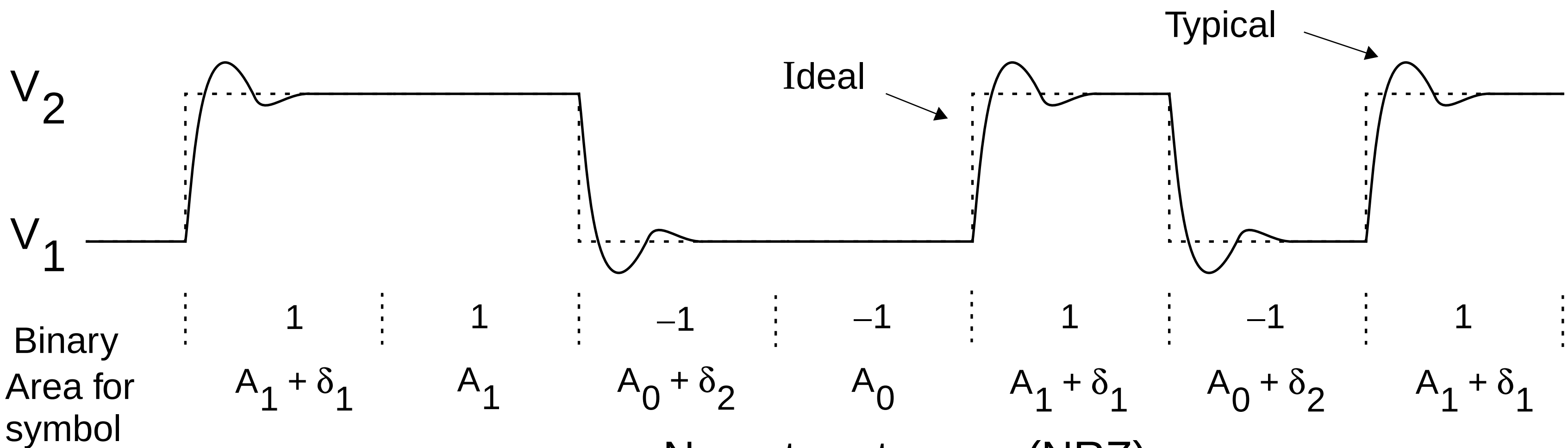
When instability detected ...

- reset integrators
- Damp some integrators to force more stable

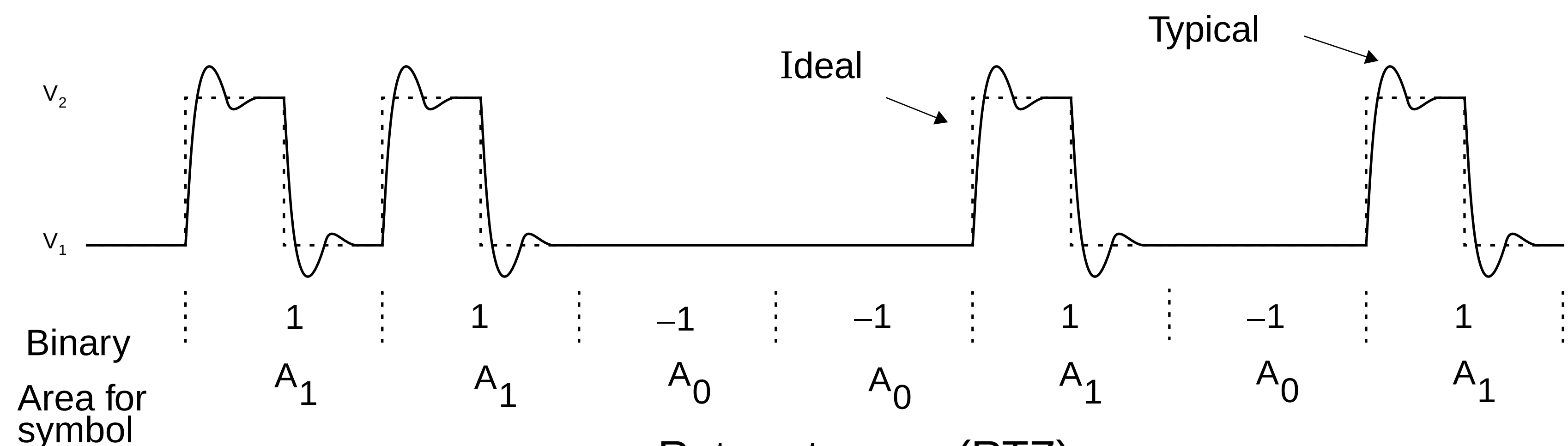
Linearity of Two-Level Converters

- For high-linearity, levels should NOT be a function of input signal
  — power supply variation might cause symptom
- Also need to be memoryless
  — switched-capacitor circuits are inherently memoryless if enough settling-time allowed
- Above linearity issues also applicable to multi-level
- A nonreturn-to-zero is NOT memoryless
- Return-to-zero is memoryless if enough settling time
- Important for continuous-time D/A

Linearity of Two-Level Converters



Nonreturn-to-zero (NRZ)



Return-to-zero (RTZ)

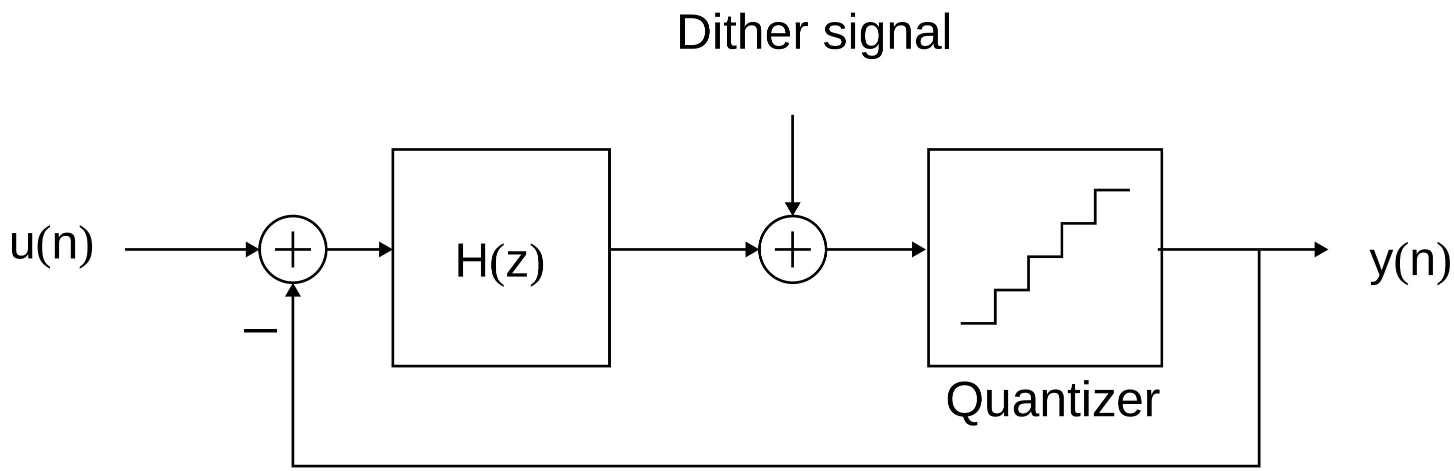
Idle Tones

- 1/3 into 1'st order modulator results in output

$$y(n) = \{1, 1, -1, 1, 1, -1, 1, 1, \ldots\} \qquad (28)$$

- Fortunately, tone is out-of-band at $f_s/3$
- $(1/3 + 1/24) = 3/8$ into modulator has tone at $f_s/16$
- Similar examples can cause tones in band-of-interest and are not filtered out — say $f_s/256$
- Also true for higher-order modulators
- Human hearing can detect tones below noise floor
- Tones might not lie at single frequency but be short term periodic patterns.
  — could be a tone varying between 900 and 1100 Hz varying in a random-like pattern

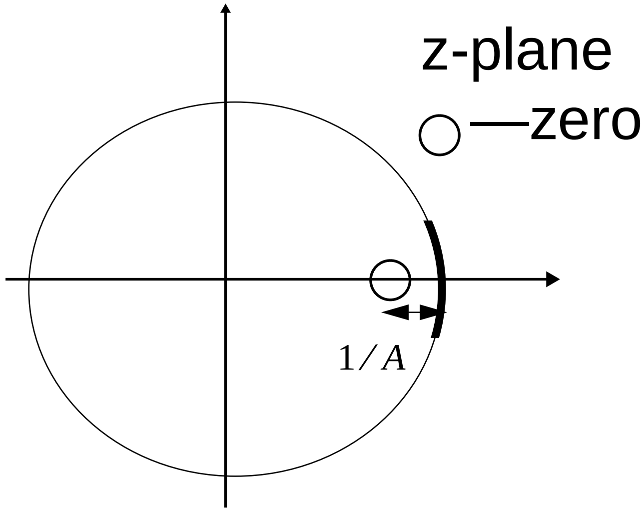
Dithering

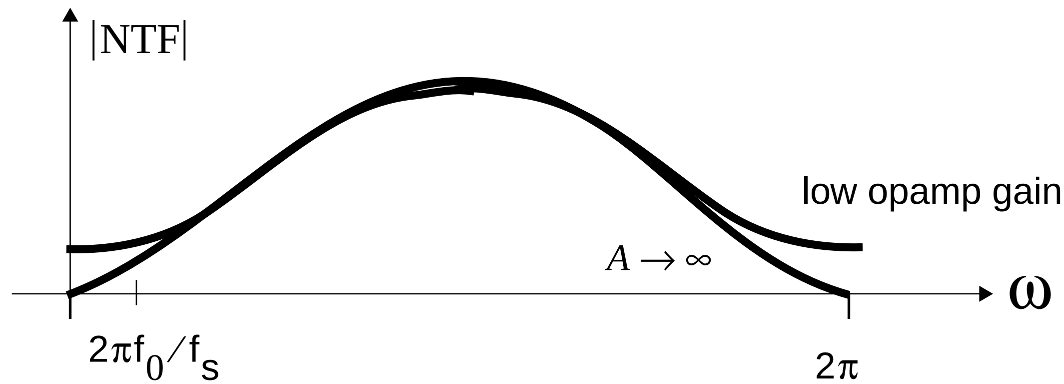


- Add pseudo-random signal into modulator to break up idle tones (not just mask them)
- If added before quantizer, it is noise shaped and large dither can be added.
  — A/D: few bit D/A converter needed
  — D/A: a few bit adder needed
- Might affect modulator stability

Opamp Gain

- Finite opamp gain, A, moves pole at $z = 1$ left by $1/A$





- Flattens out noise at low frequency
  — only 3 dB/octave for high OSR
- Typically, require

$$A > OSR/\pi \qquad (29)$$

Multi-bit Oversampled Converters

- A multi-bit DAC has many advantages
  — more stable - higher peak $|NTF|$
  — higher input range
  — less quantization noise introduced
  — less idle tones (perhaps no dithering needed)
- Need highly linear multi-bit D/A converters

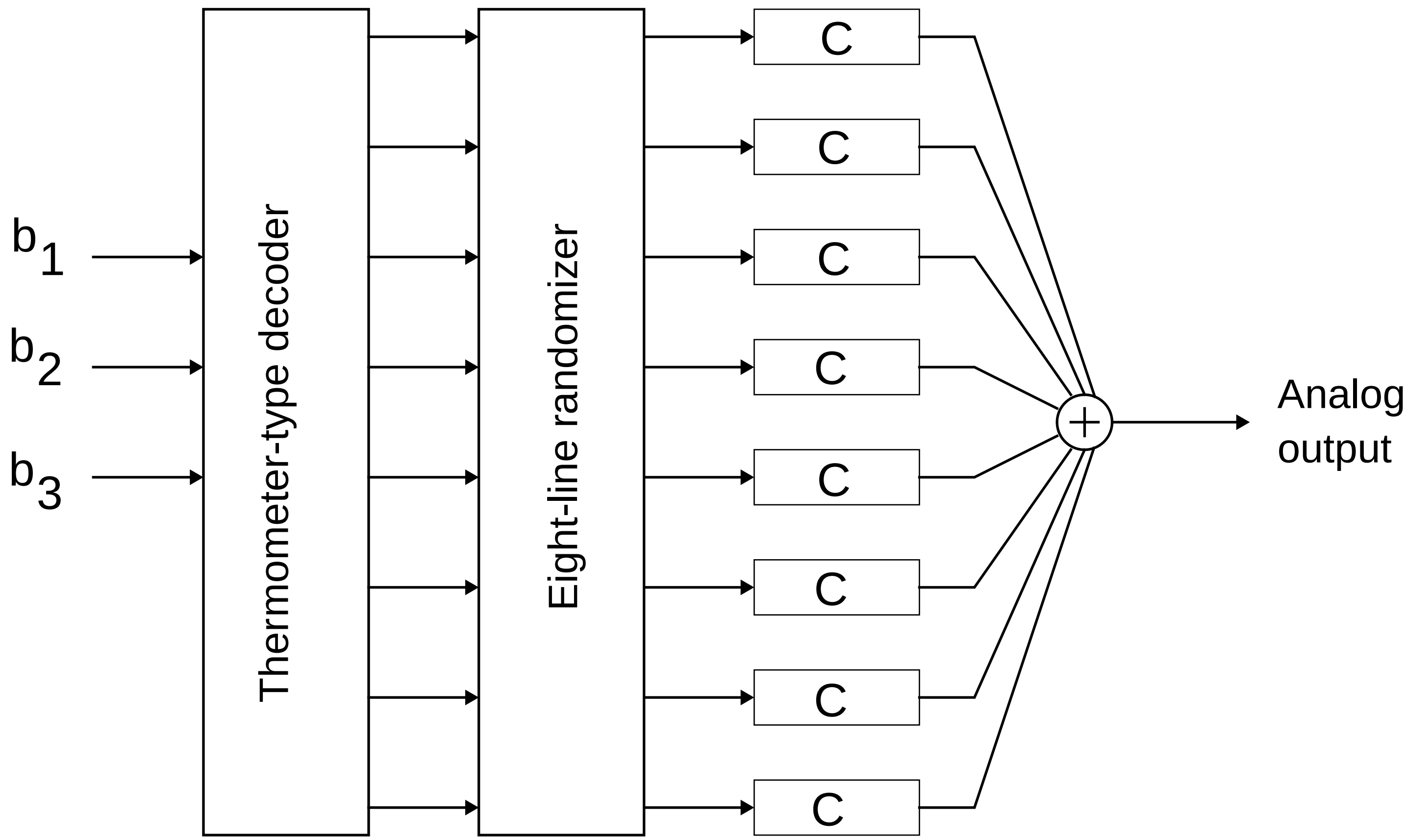
Example

- A 4-bit DAC has 18 dB less quantization noise, up to 12 dB higher input range — perhaps 30 dB improved SNR over 1-bit

Large Advantage in DAC Application

- Less quantization noise — easier analog lowpass filter

Multi-bit Oversampled Converters



- Randomize thermometer code
- Can also “shape” nonlinearities

Third-Order A/D Design Example

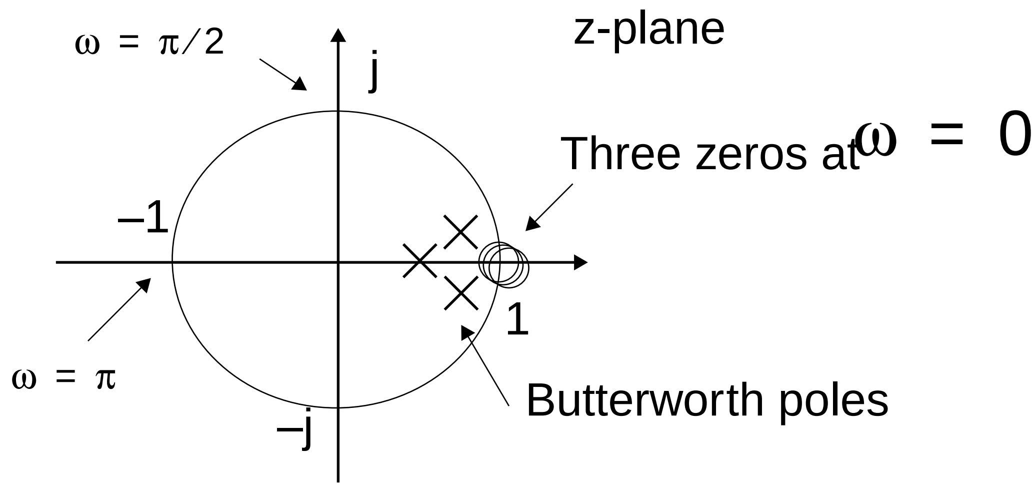
- All NTF zeros at $z = 1$

$$NTF(z) = \frac{(z-1)^3}{D(z)} \tag{30}$$

- Find $D(z)$ such that $\left|NTF(e^{j\omega})\right| < 1.4$
- Use Matlab to find a Butterworth highpass filter with peak gain near 1.4
- If passband edge at $f_s/20$ then peak gain = 1.37

$$NTF(z) = \frac{(z-1)^3}{z^3 - 2.3741z^2 + 1.9294z - 0.5321} \tag{31}$$

Third-Order A/D Design Example



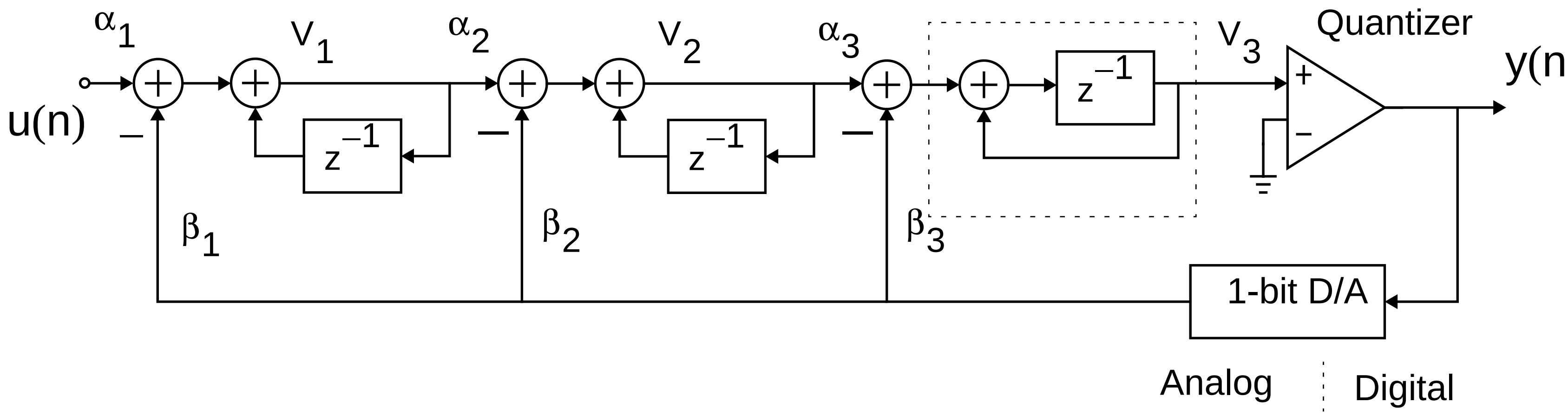
- Find $H(z)$ as

$$H(z) = \frac{1 - NTF(z)}{NTF(z)} \tag{32}$$

$$H(z) = \frac{0.6259z^2 - 1.0706z + 0.4679}{(z-1)^3} \tag{33}$$

Third-Order A/D Design Example

- Choosing a cascade of integrator structure



- α_i coefficients included for dynamic-range scaling
  — initially $\alpha_2 = \alpha_3 = 1$
  — last term, α_1, initially set to β_1 so input is stable for a reasonable input range
- Initial β_i found by deriving transfer function from 1-bit D/A output to V_3 and equating to $-H(z)$

Third-Order A/D Design Example

$$H(z) = \frac{z^2(\beta_1 + \beta_2 + \beta_3) - z(\beta_2 + 2\beta_3) + \beta_3}{(z-1)^3} \qquad (34)$$

- Equating (33) and (34) results in

$$\begin{aligned} \alpha_1 &= 0.0232, \quad \alpha_2 = 1.0, \quad \alpha_3 = 1.0 \\ \beta_1 &= 0.0232, \quad \beta_2 = 0.1348, \quad \beta_3 = 0.4679 \end{aligned} \qquad (35)$$

Third-Order A/D Design Example

Dynamic Range Scaling

- Apply sinusoidal input signal with peak value of 0.7 and frequency $\pi/256$ rad/sample
- Simulation shows max values at nodes V_1, V_2, V_3 of 0.1256, 0.5108, and 1.004
- Can scale node V_1 by k_1 by multiplying α_1 and β_1 by k_1 and dividing α_2 by k_1
- Can scale node V_2 by k_2 by multiplying α_2/k_1 and β_2 by k_2 and dividing α_3 by k_2

$$\begin{aligned} &\alpha'_1 = 0.1847, \quad \alpha'_2 = 0.2459, \quad \alpha'_3 = 0.5108 \\ &\beta'_1 = 0.1847, \quad \beta'_2 = 0.2639, \quad \beta'_3 = 0.4679 \end{aligned} \tag{36}$$

Third-Order A/D Design Example