ESE 531: Digital Signal Processing

Lec 11: February 20, 2018

Data Converters, Noise Shaping

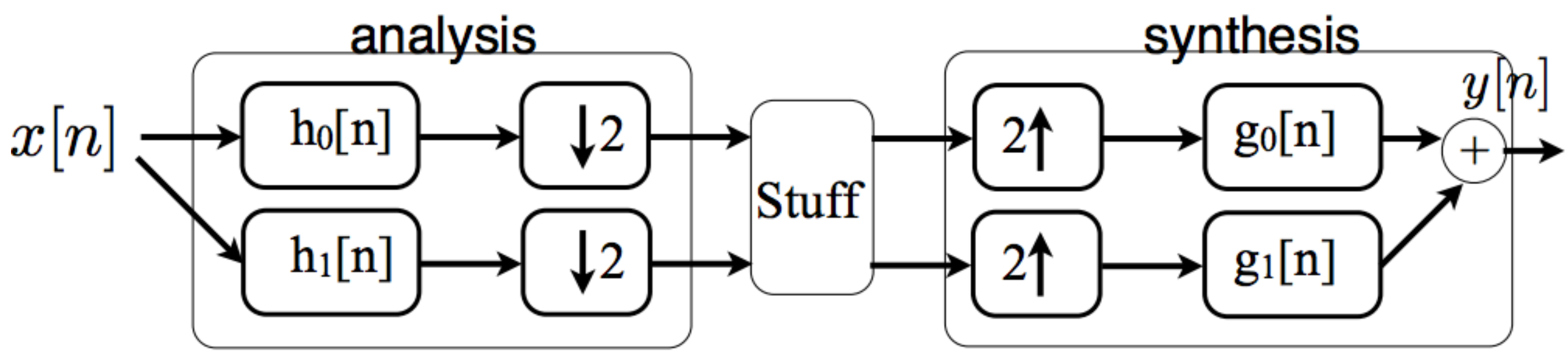
Lecture Outline

- Review: Multi-Rate Filter Banks
  - Quadrature Mirror Filters
- Data Converters
  - Anti-aliasing
  - ADC
    - Quantization
  - Practical DAC
- Noise Shaping

Multi-Rate Filter Banks

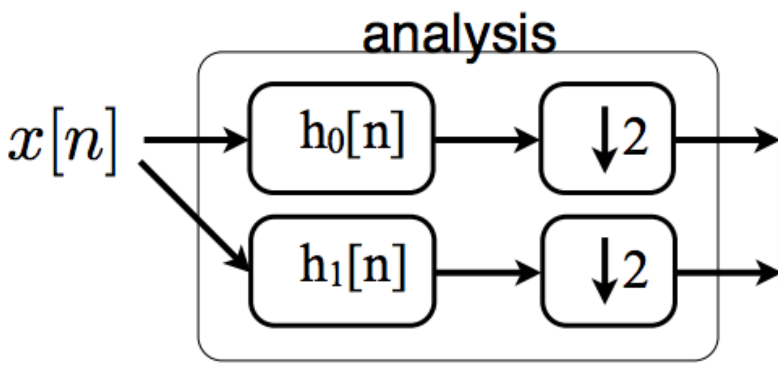
- Use filter banks to operate on a signal differently in different frequency bands
  - To save computation, reduce the rate after filtering

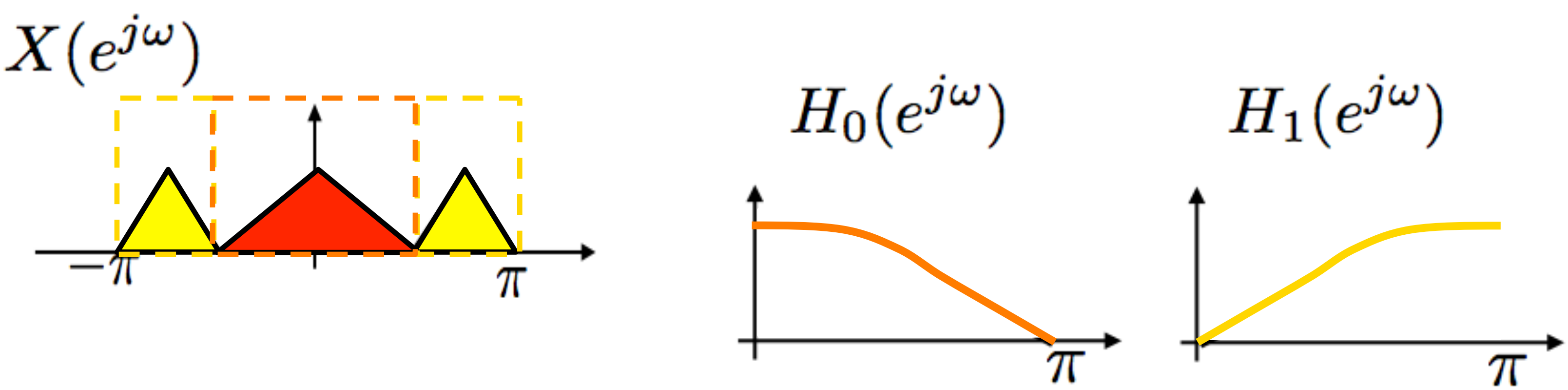
- $h_0[n]$ is low-pass, $h_1[n]$ is high-pass
  - Often $h_1[n]=e^{j\pi n}h_0[n]$ ← shift freq resp by π



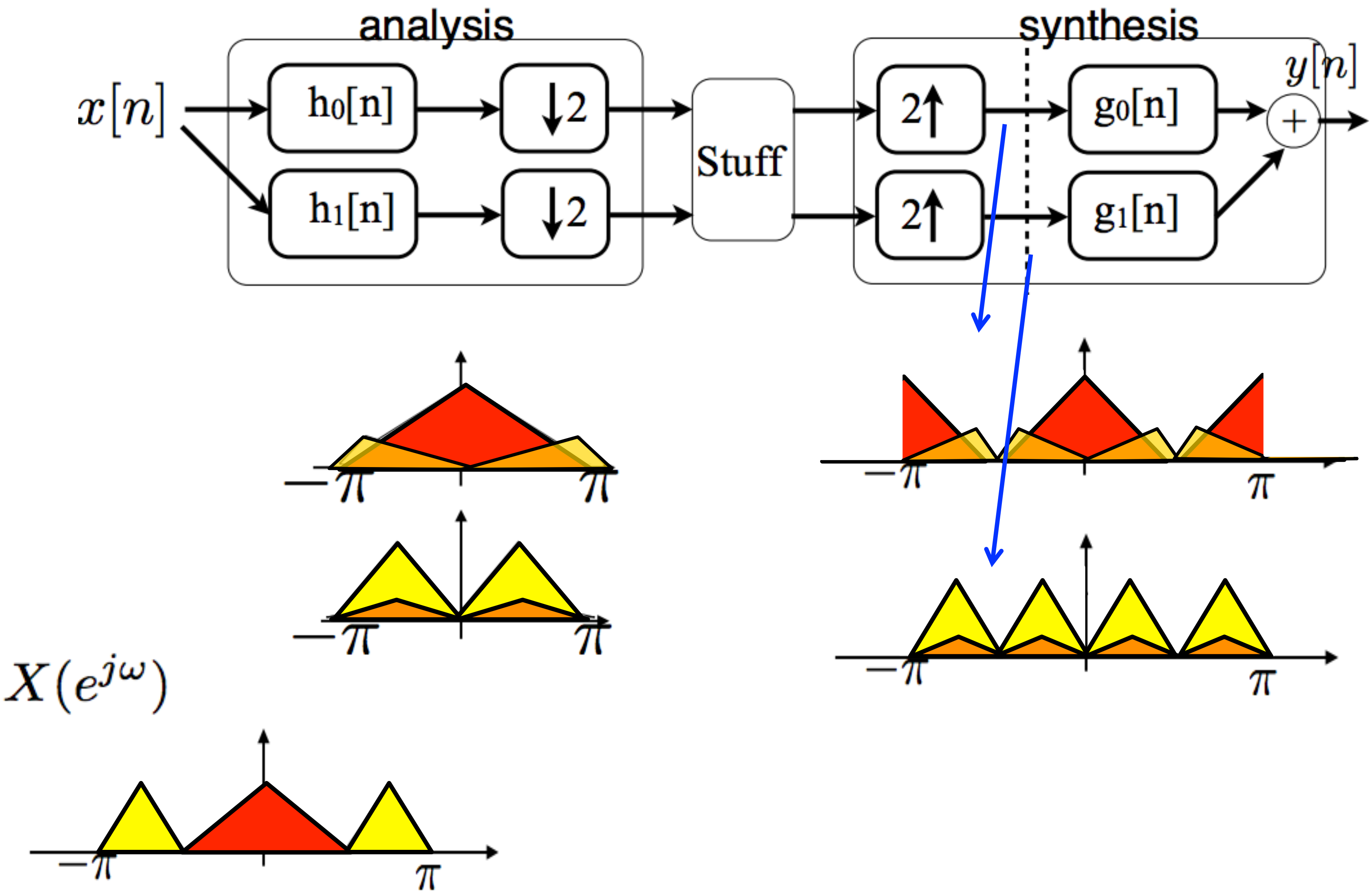
Multi-Rate Filter Banks

- h_0, h_1 are NOT ideal low/high pass

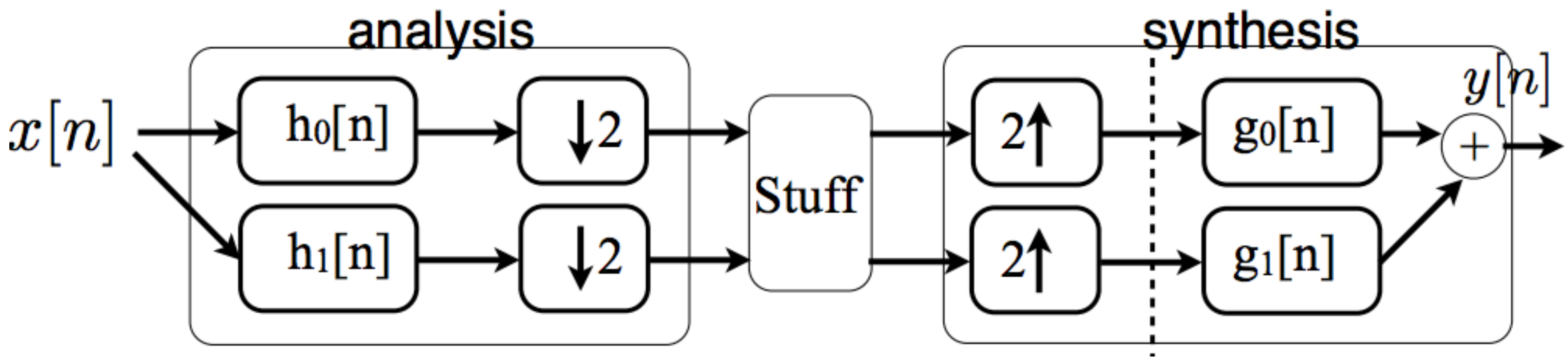




Non Ideal Filters



Perfect Reconstruction non-Ideal Filters

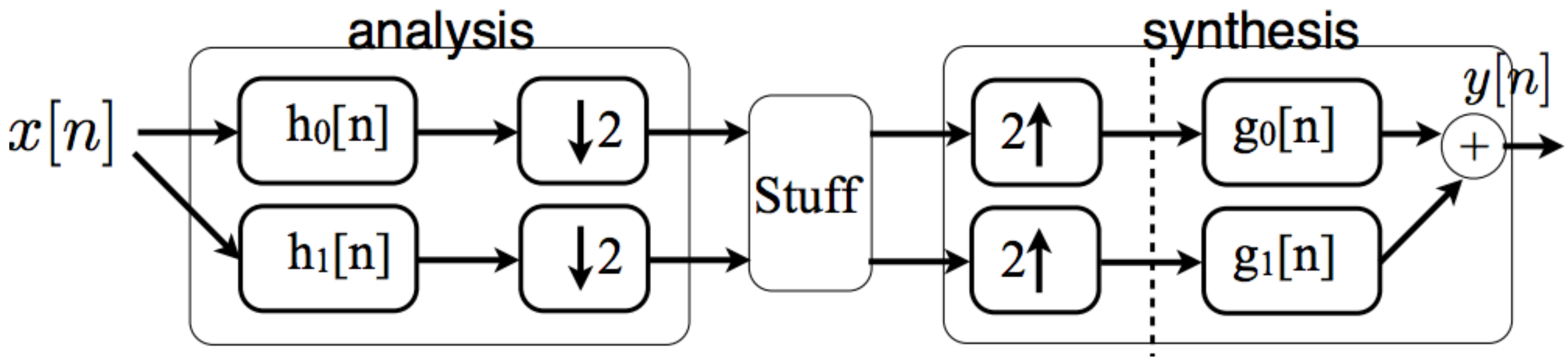


$$Y(e^{j\omega}) = \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right] X(e^{j\omega})$$
$$+\frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right] X(e^{j(\omega-\pi)})$$

need to cancel!

aliasing

Quadrature Mirror Filters



Quadrature mirror filters

$$
\begin{aligned}
H_1(e^{j\omega}) &= H_0(e^{j(\omega-\pi)}) \\
G_0(e^{j\omega}) &= 2H_0(e^{j\omega}) \\
G_1(e^{j\omega}) &= -2H_1(e^{j\omega})
\end{aligned}
$$

Perfect Reconstruction non-Ideal Filters

$$Y(e^{j\omega}) = \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right]X(e^{j\omega})$$
$$+\frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right]X(e^{j(\omega-\pi)})$$

need to cancel!

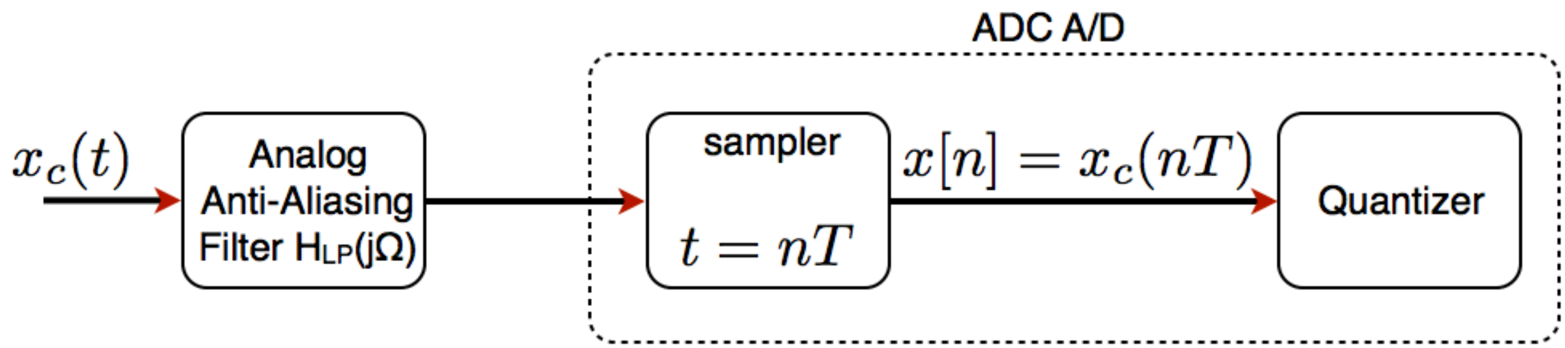
aliasing

$$\begin{aligned} H_1(e^{j\omega}) &= H_0(e^{j(\omega-\pi)}) \\ G_0(e^{j\omega}) &= 2H_0(e^{j\omega}) \\ G_1(e^{j\omega}) &= -2H_1(e^{j\omega}) \end{aligned}$$

ADC

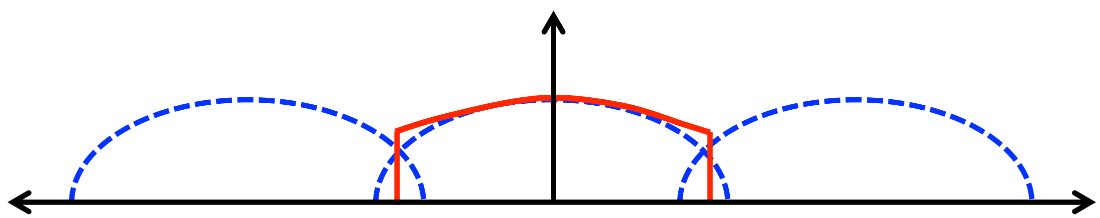
Analog to Digital Converter

Anti-Aliasing Filter with ADC



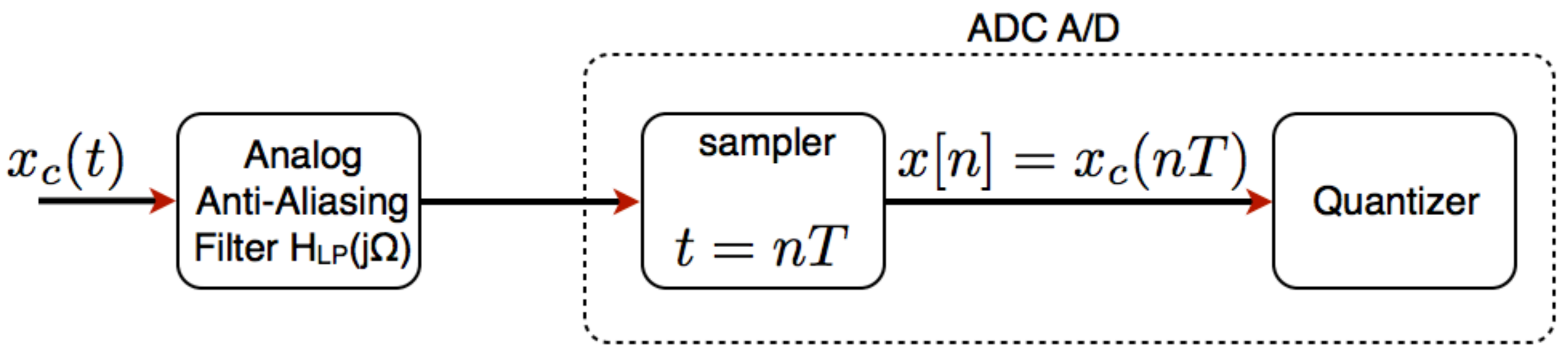
Aliasing

- If $\Omega_N > \Omega_s/2$, $x_r(t)$ an aliased version of $x_c(t)$

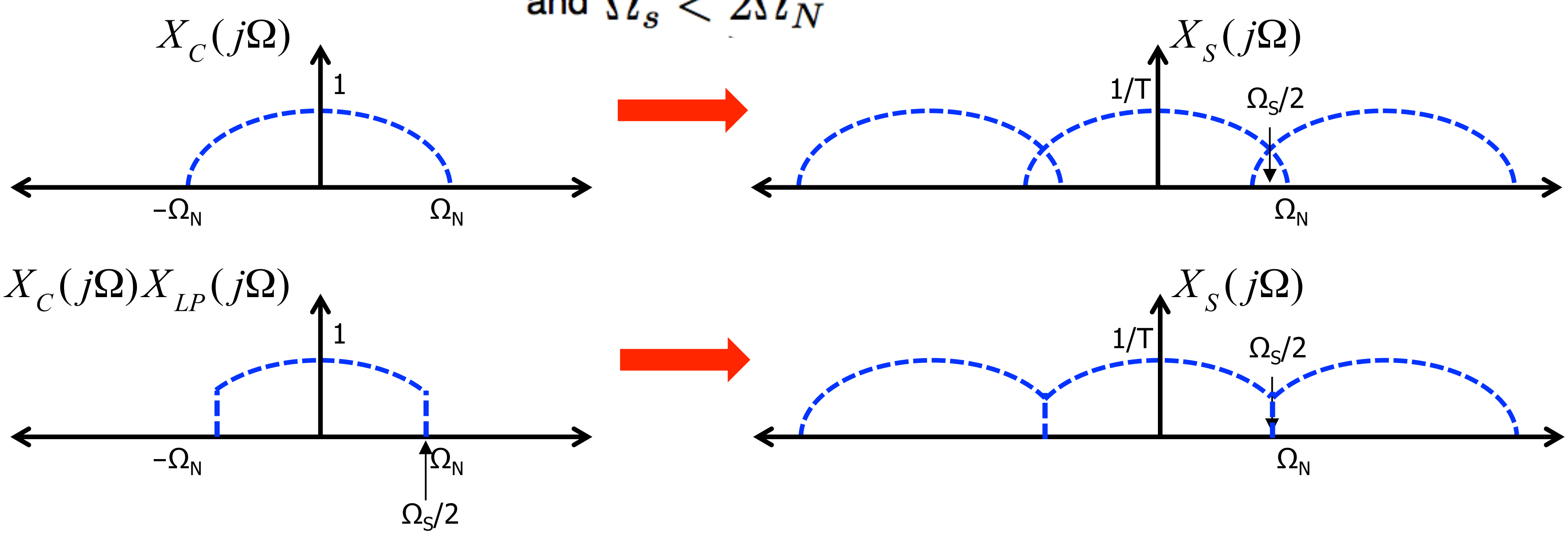


$$X_r(j\Omega) = \begin{cases} TX_s(j\Omega) & \text{if} \quad |\Omega| \le \Omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

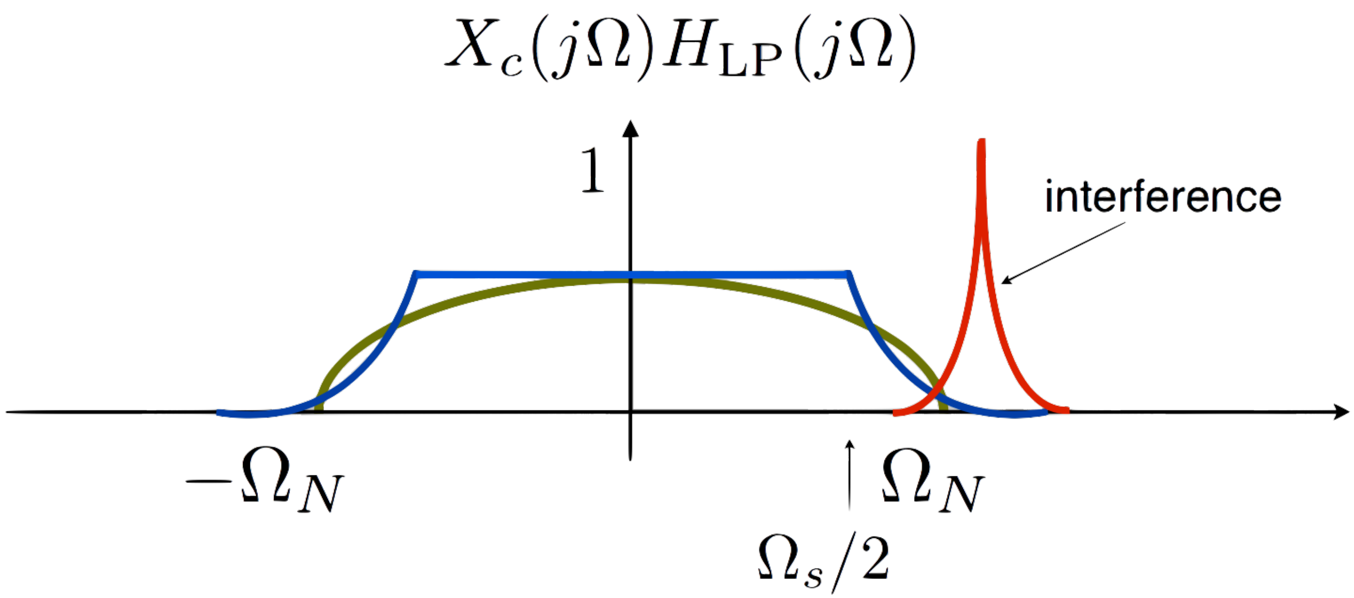
Anti-Aliasing Filter with ADC

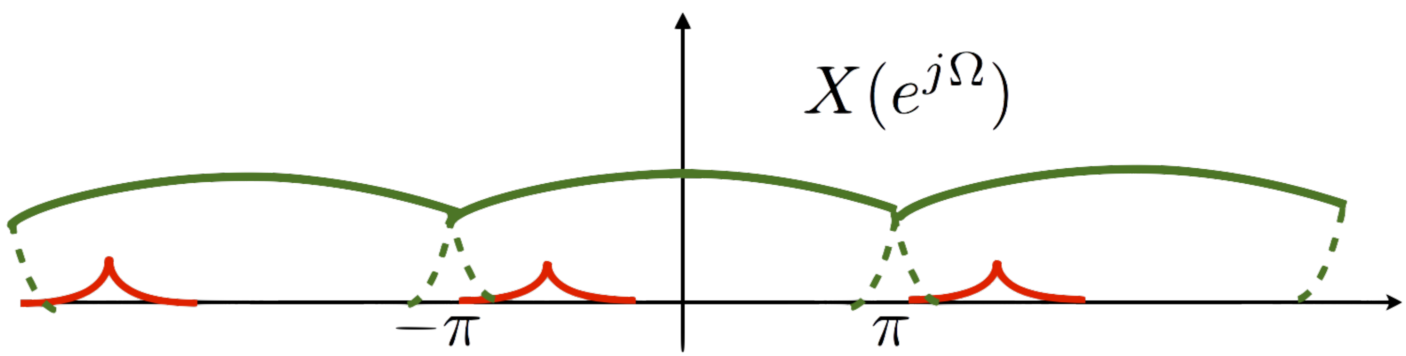


and $\Omega_s < 2\Omega_N$

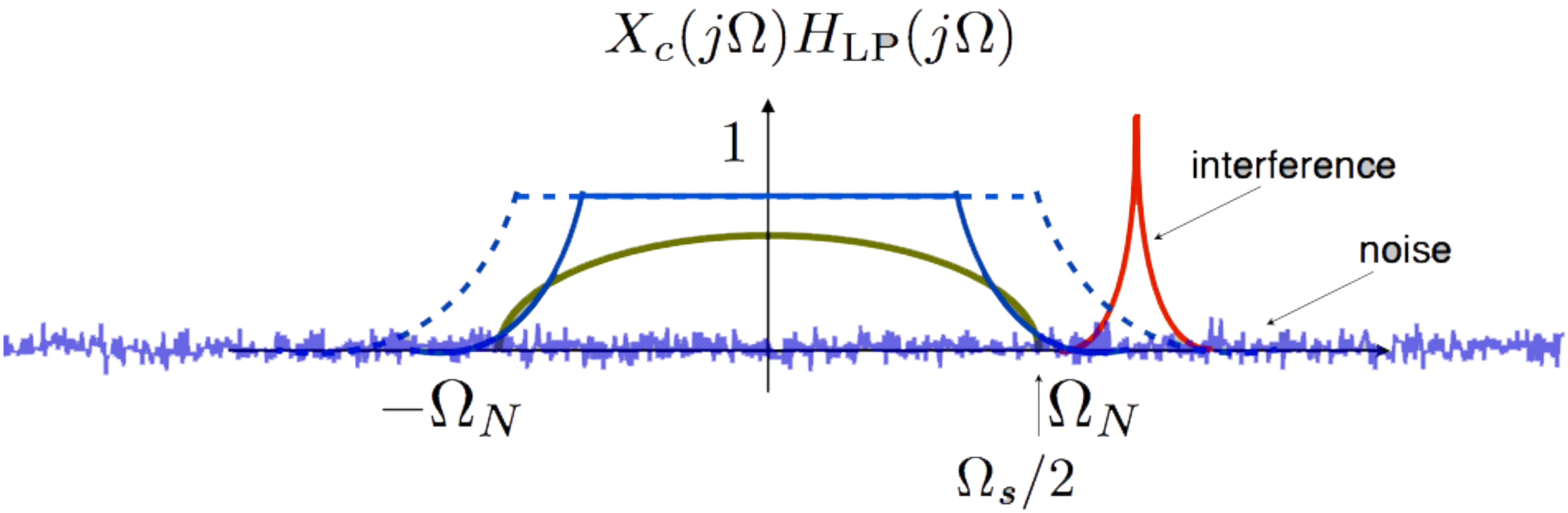


Non-Ideal Anti-Aliasing Filter



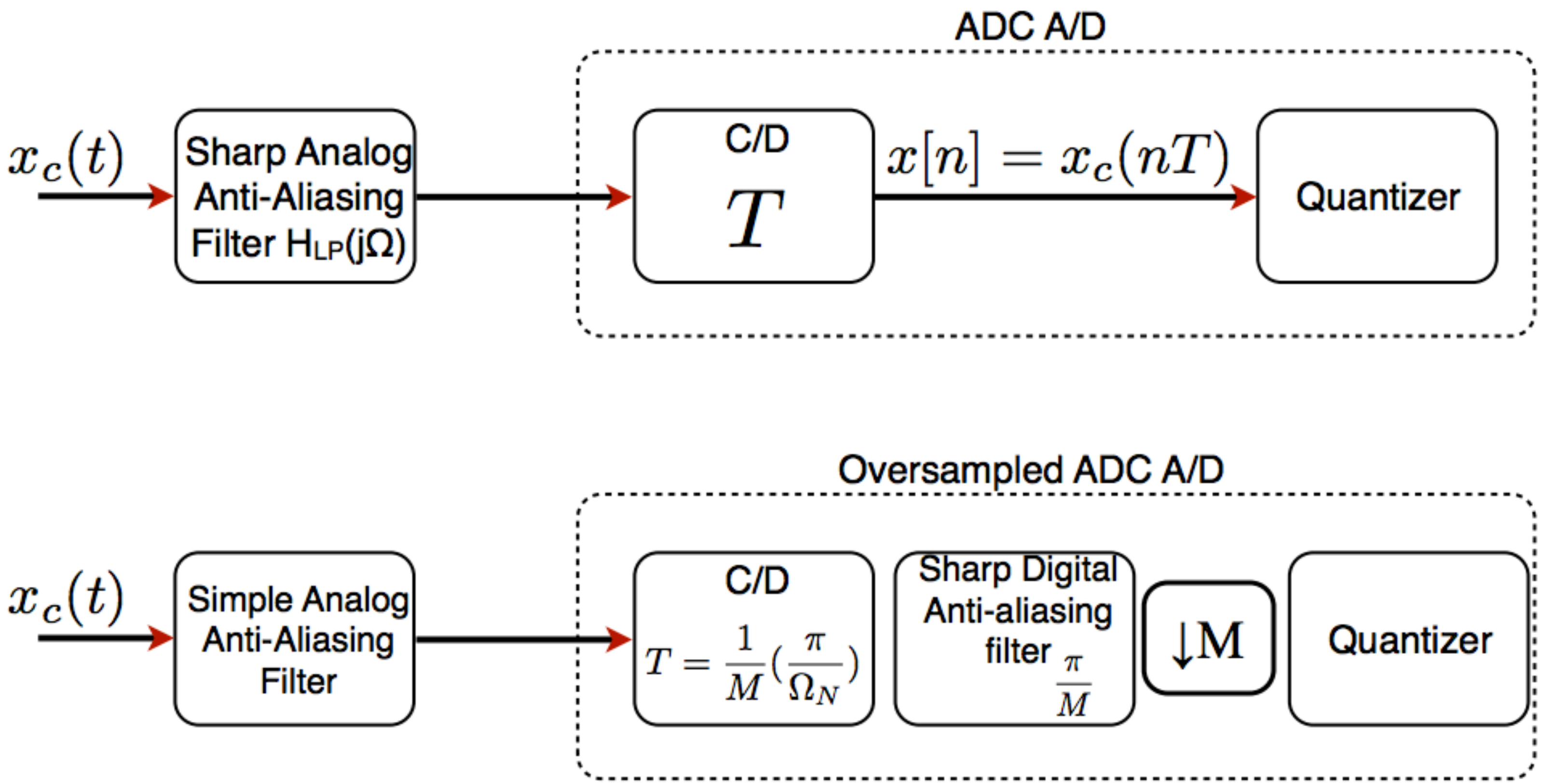


Non-Ideal Anti-Aliasing Filter

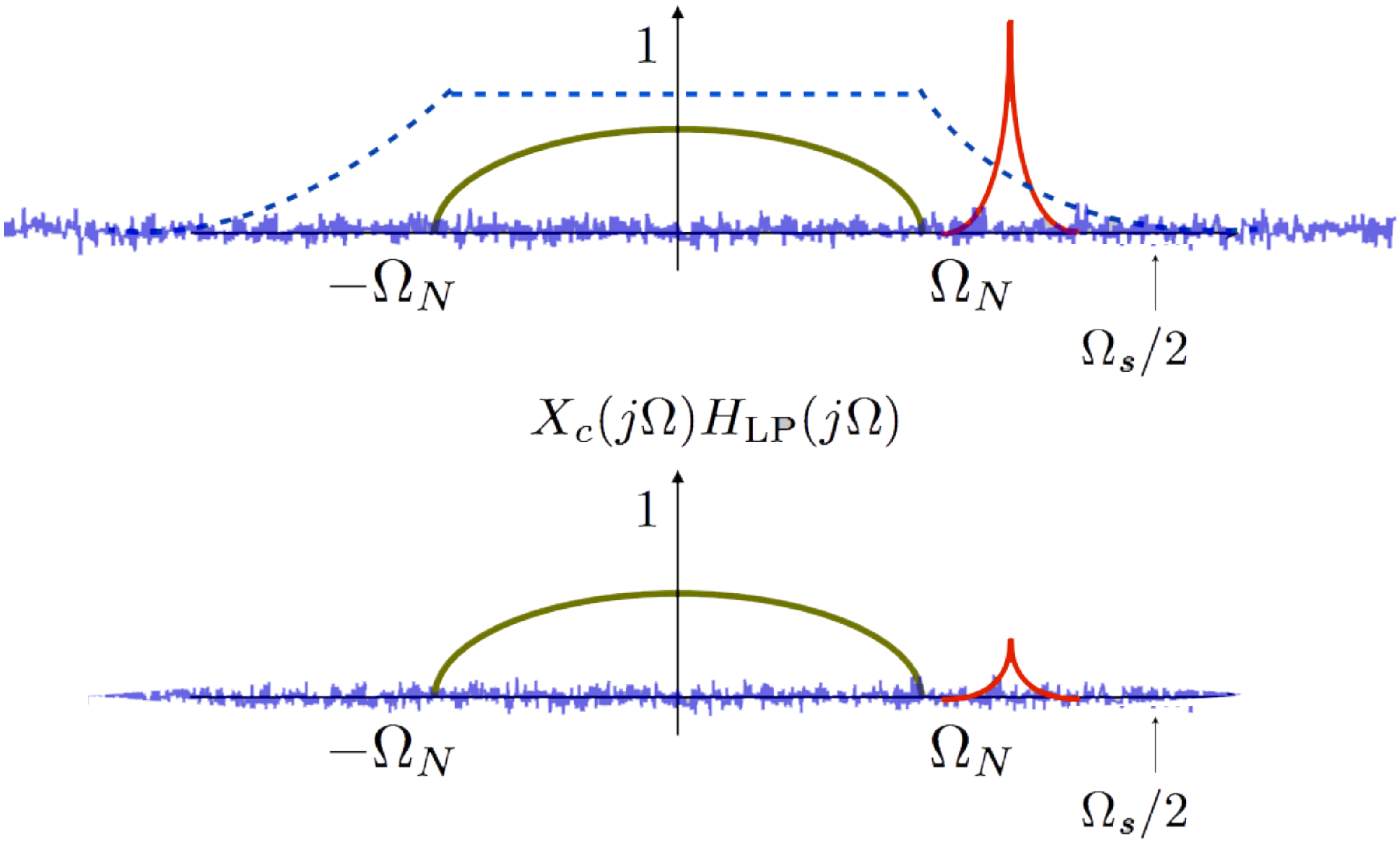


- Problem: Hard to implement sharp analog filter
- Solution: Crop part of the signal and suffer from noise and interference

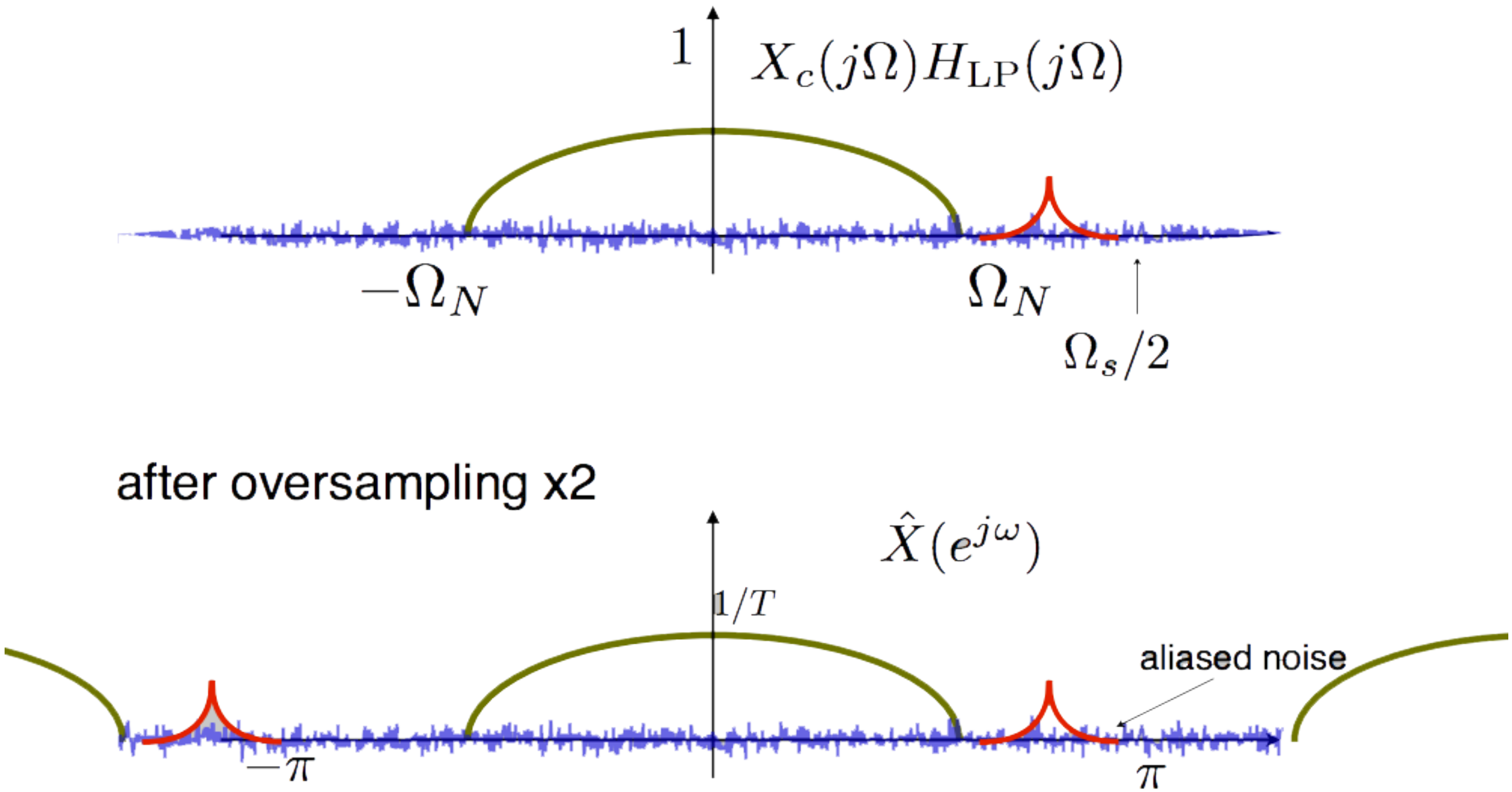
Oversampled ADC



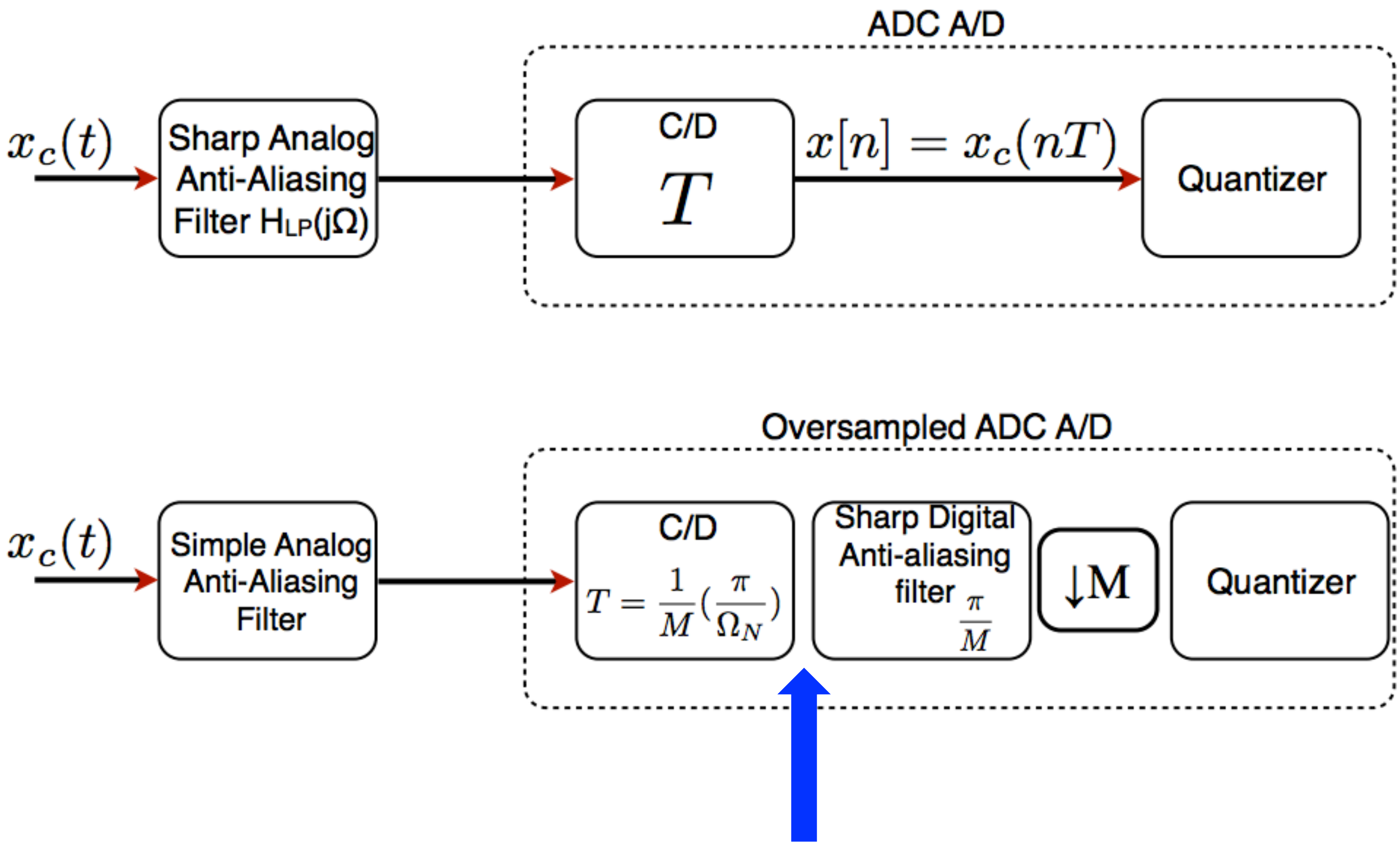
Oversampled ADC (2x)



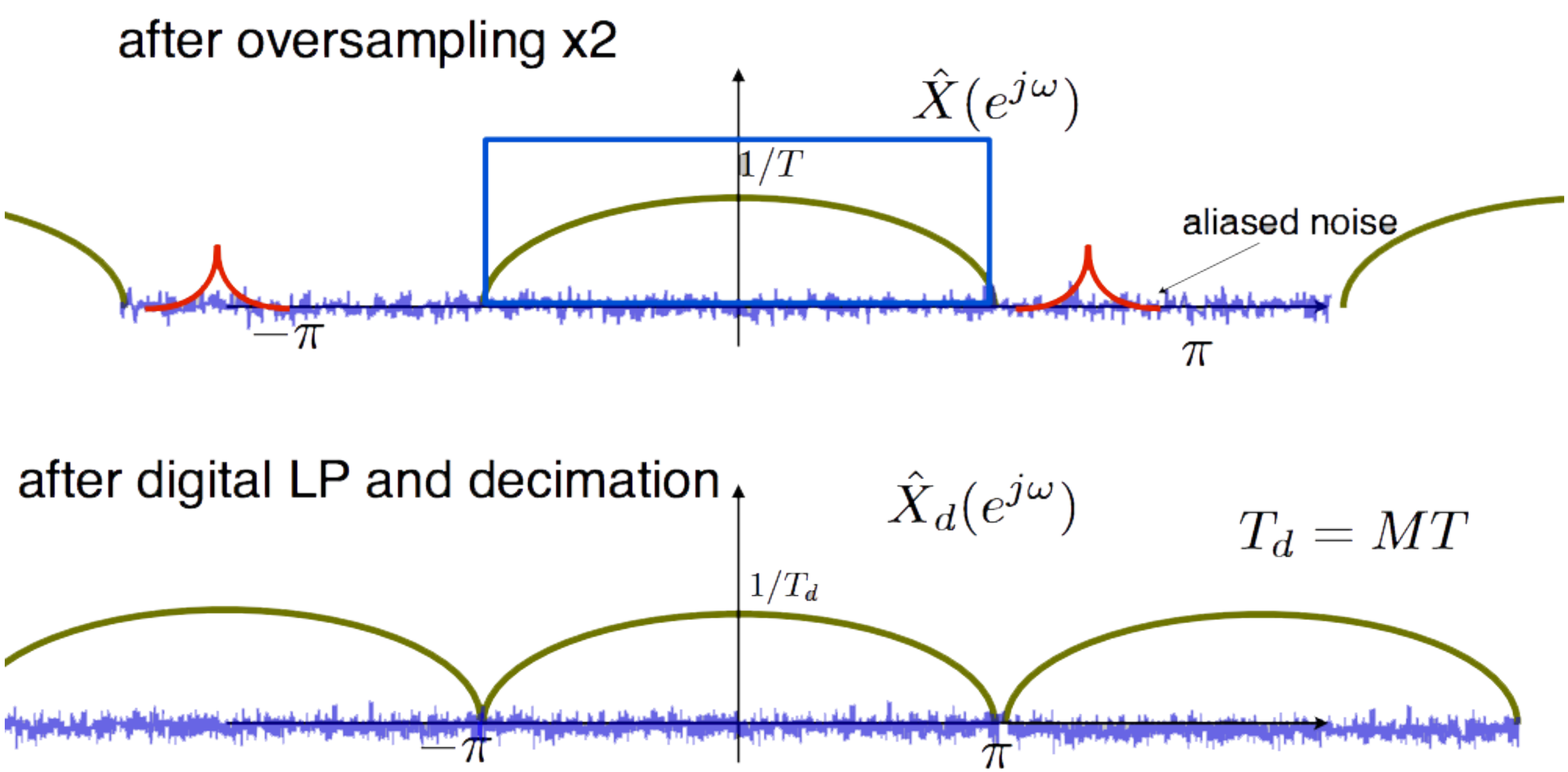
Oversampled ADC



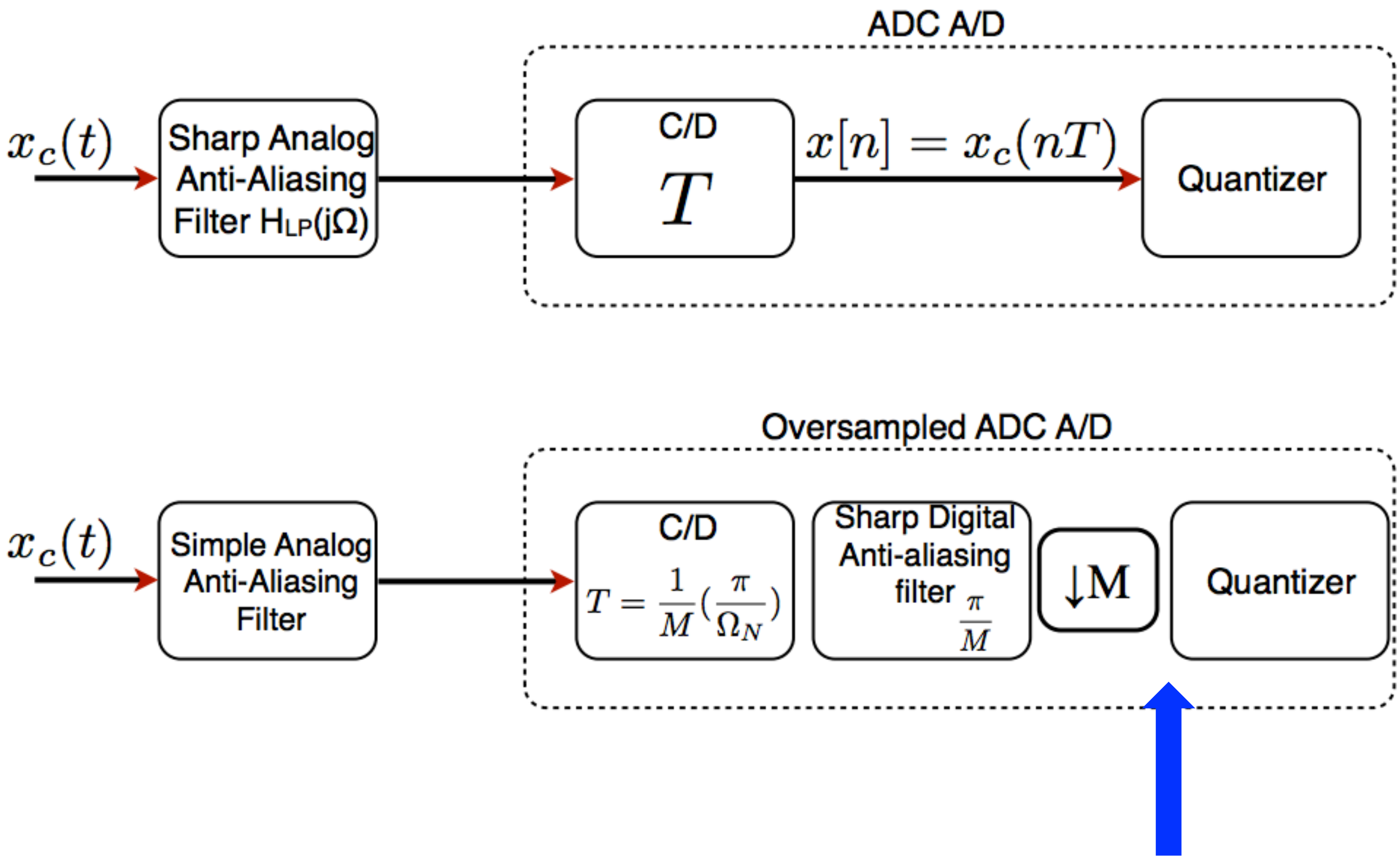
Oversampled ADC



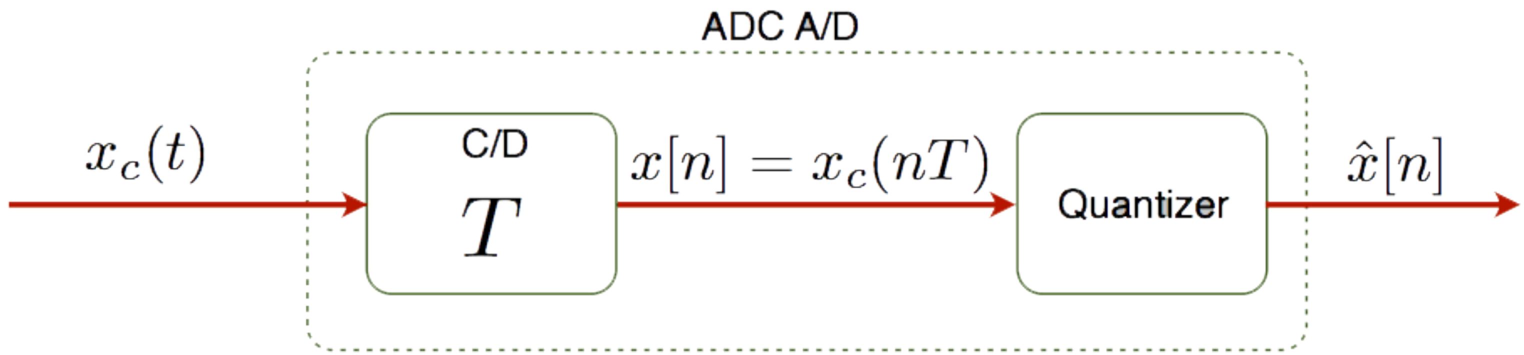
Oversampled ADC

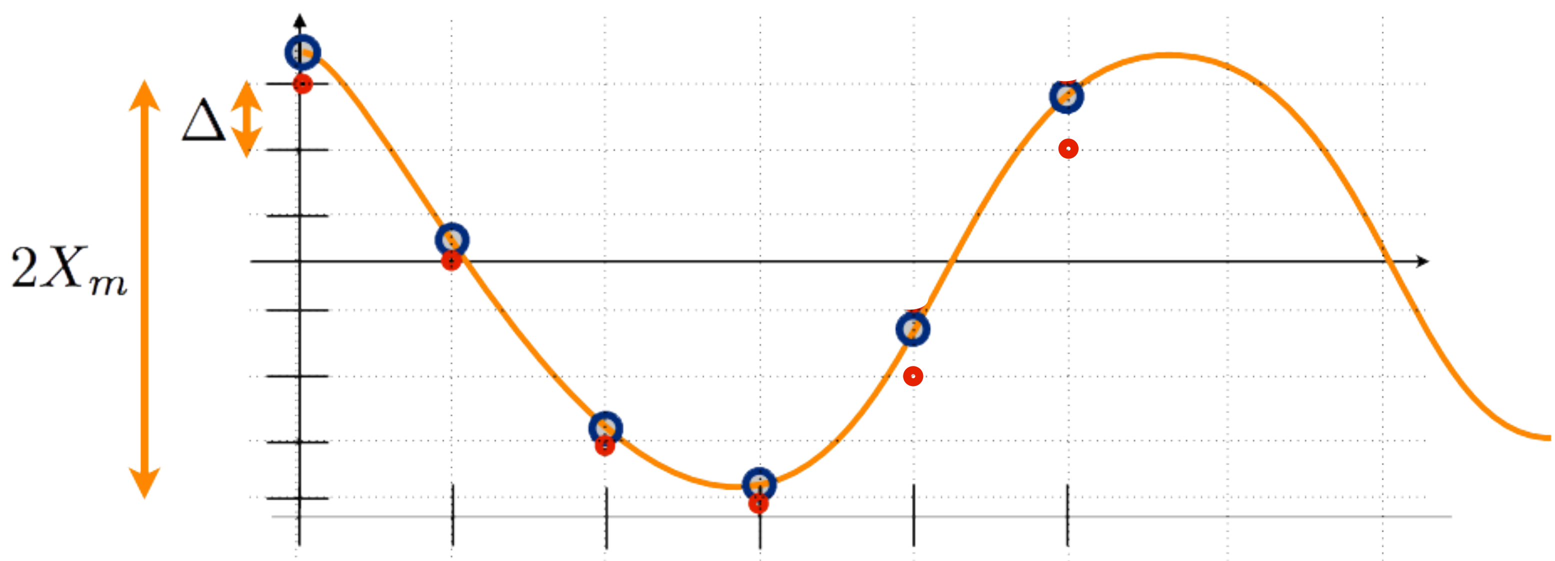


Oversampled ADC



Sampling and Quantization

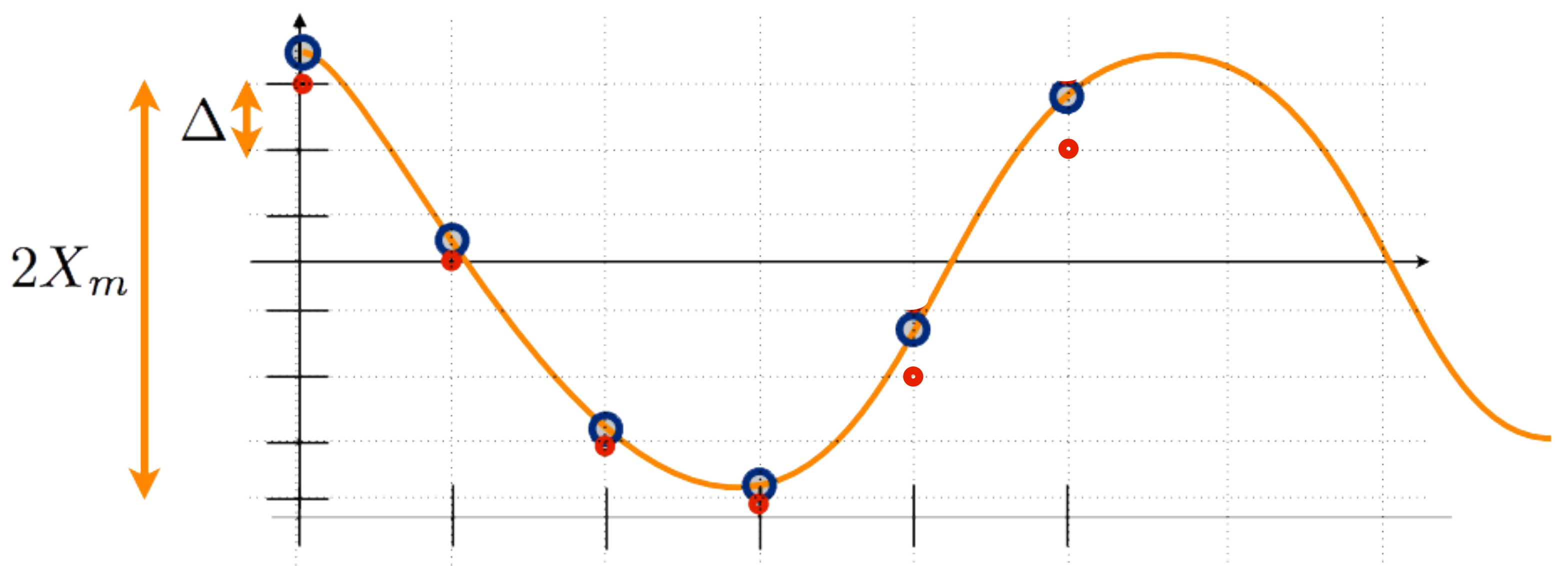




Sampling and Quantization

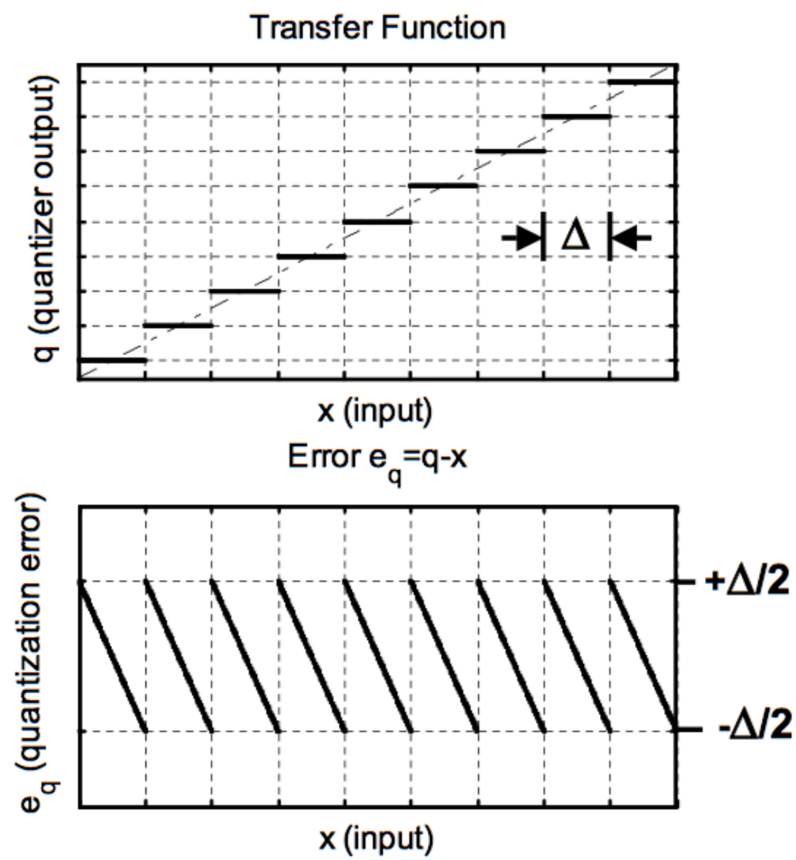
for 2's complement with B+1 bits $-1 \leq \hat{x}_B[n] < 1$

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$



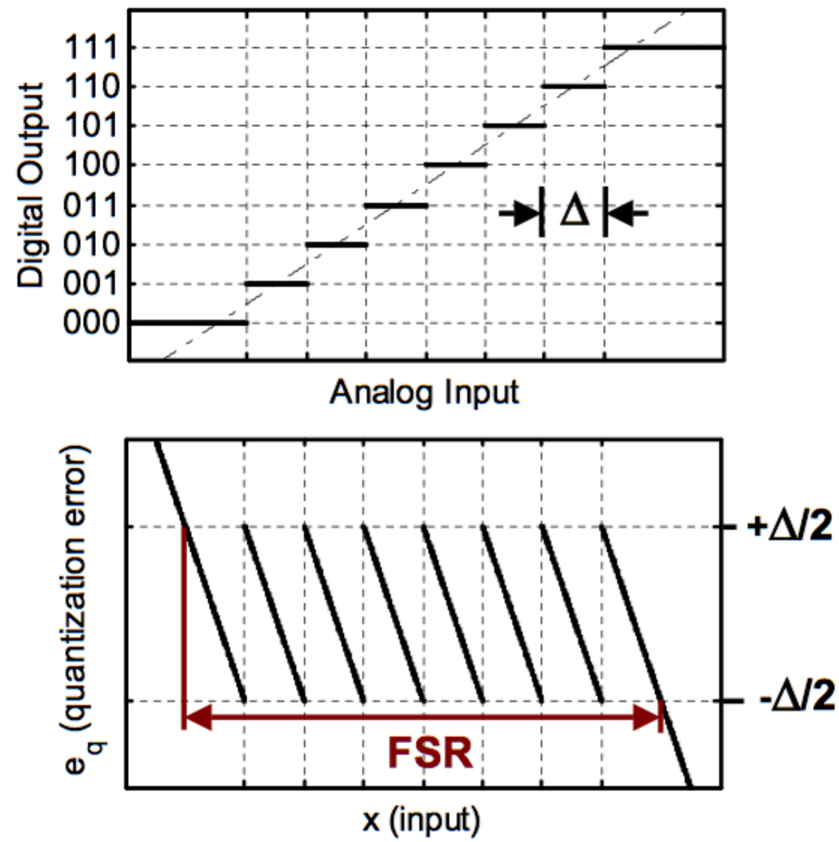
Ideal Quantizer

- Quantization step Δ
- Quantization error has sawtooth shape,
  - Bounded by $-\Delta/2, +\Delta/2$
- Ideally infinite input range and infinite number of quantization levels



Ideal B-bit Quantizer

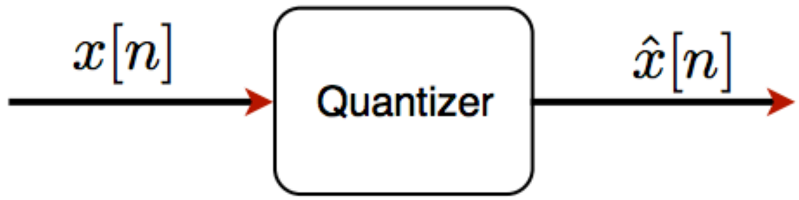
- Practical quantizers have a limited input range and a finite set of output codes
- E.g. a 3-bit quantizer can map onto $2^3=8$ distinct output codes
  - Diagram on the right shows "offset-binary encoding"
    - See Gustavsson (p.2) for other coding formats
- Quantization error grows out of bounds beyond code boundaries
- We define the full scale range (FSR) as the maximum input range that satisfies $|e_q| \le \Delta/2$
  - Implies that FSR = $2^B \cdot \Delta$



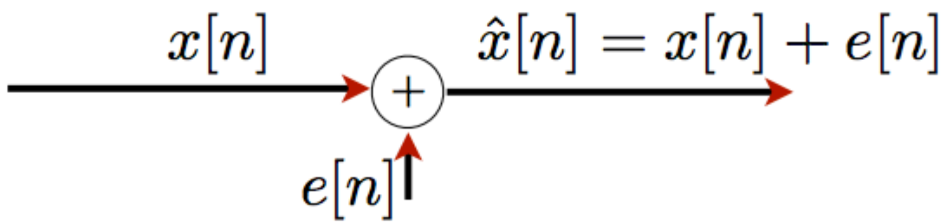
Effect of Quantization Error on Signal

- Quantization error is a deterministic function of the signal
  - Consequently, the effect of quantization strongly depends on the signal itself
- Unless, we consider fairly trivial signals, a deterministic analysis is usually impractical
  - More common to look at errors from a statistical perspective
  - "Quantization noise”
- Two aspects
  - How much noise power (variance) does quantization add to our samples?
  - How is this noise distributed in frequency?

Quantization Error



- Model quantization error as noise



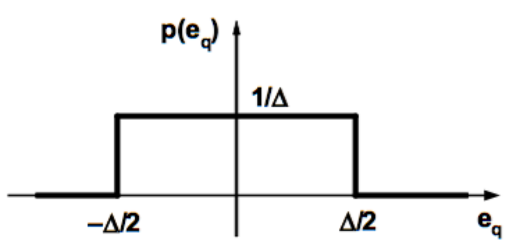
- In that case:

$$-\Delta/2 \leq e[n] < \Delta/2$$

$$(-X_m - \Delta/2) < x[n] \leq (X_m - \Delta/2)$$

Quantization Error Statistics

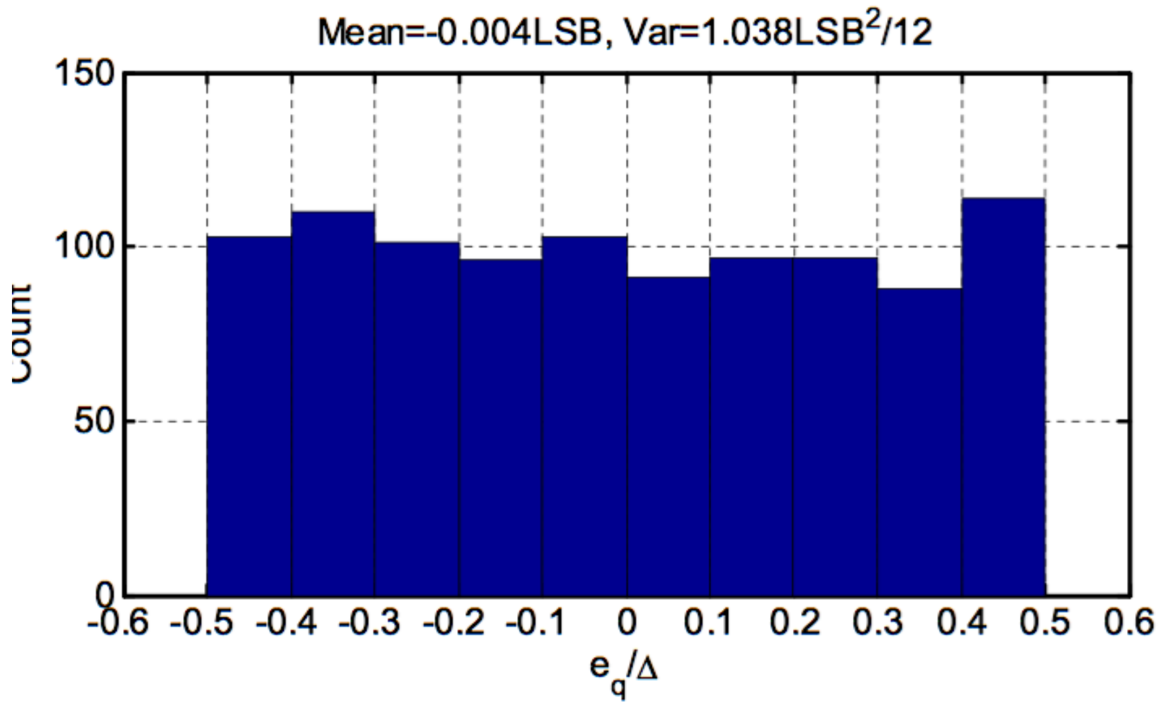
- Crude assumption: $e_q(x)$ has uniform probability density
- This approximation holds reasonably well in practice when
  - Signal spans large number of quantization steps
  - Signal is "sufficiently active”
  - Quantizer does not overload



Mean $$\bar{e} = \int_{-\Delta/2}^{+\Delta/2} \frac{e}{\Delta} de = 0$$

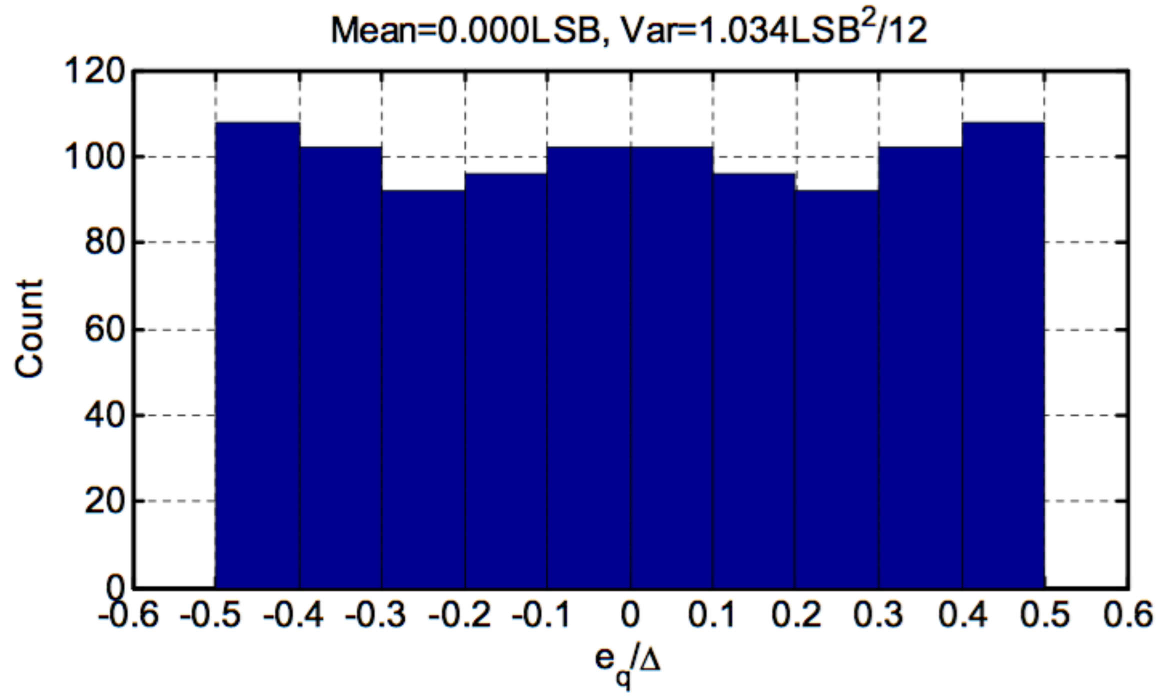
Variance $$\overline{e^2} = \int_{-\Delta/2}^{+\Delta/2} \frac{e^2}{\Delta} de = \frac{\Delta^2}{12}$$

Reality Check



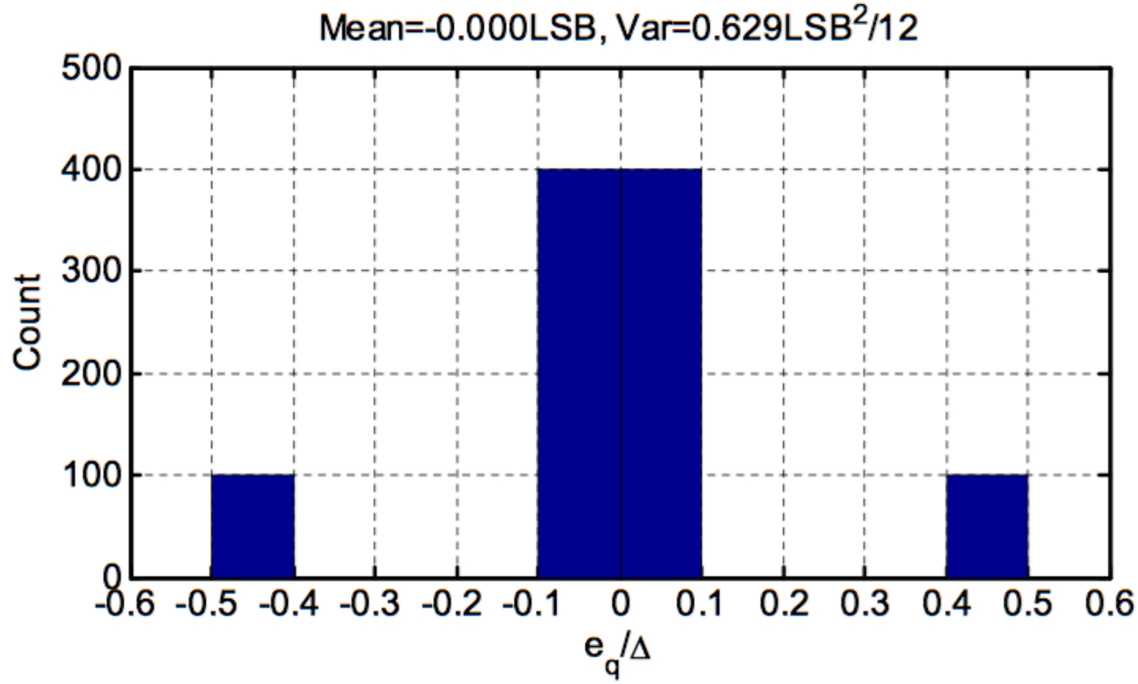
- Shown below is a histogram of e_q in an 8-bit quantizer
  - Input sequence consists of 1000 samples with Gaussian distribution, 4σ =FSR

Reality Check



- Same as before, but now using a sinusoidal input signal with f_{sig}/f_s=101/1000

Reality Check



- Same as before, but now using a sinusoidal input signal with f_{sig}/f_s=100/1000
- What went wrong?

Analysis

$$\mathrm{v}_{\mathrm{sig}}(n) = \cos\left(2\pi \cdot \frac{f_{sig}}{f_S} \cdot n\right)$$

- Signal repeats every m samples, where m is the smallest integer that satisfies

$$m \cdot \frac{f_{sig}}{f_S} = \text{integer}$$

$$m \cdot \frac{101}{1000} = \text{integer} \Rightarrow \text{m=1000}$$

$$m \cdot \frac{100}{1000} = \text{integer} \Rightarrow \text{m=10}$$

- This means that in the last case $e_q(n)$ consists at best of 10 different values, even though we took 1000 samples

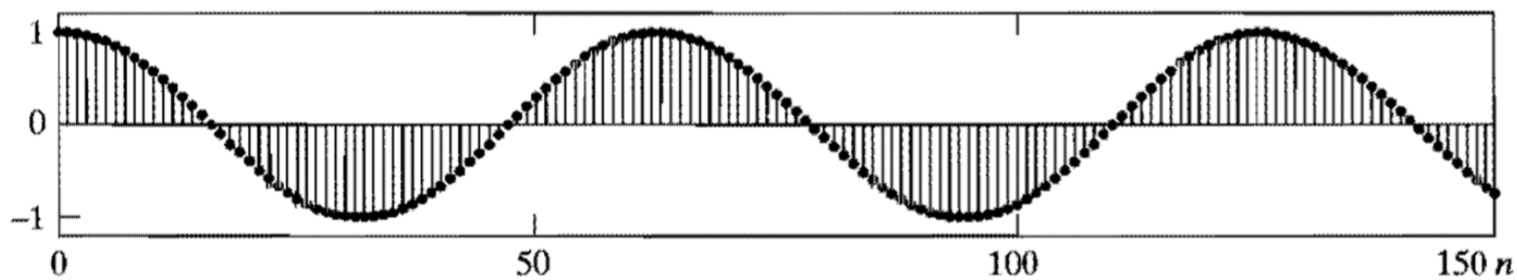
Noise Model for Quantization Error

- Assumptions:
  - Model e[n] as a sample sequence of a stationary random process
  - e[n] is not correlated with x[n]
  - e[n] not correlated with e[m] where m≠n (white noise)
  - e[n] ~ U[-Δ/2, Δ/2] (uniform pdf)

- Result:
- Variance is: $\sigma_e^2 = \frac{\Delta^2}{12}$, or $\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}$ since $\Delta = 2^{-B} X_m$
- Assumptions work well for signals that change rapidly, are not clipped, and for small Δ

Quantization Noise

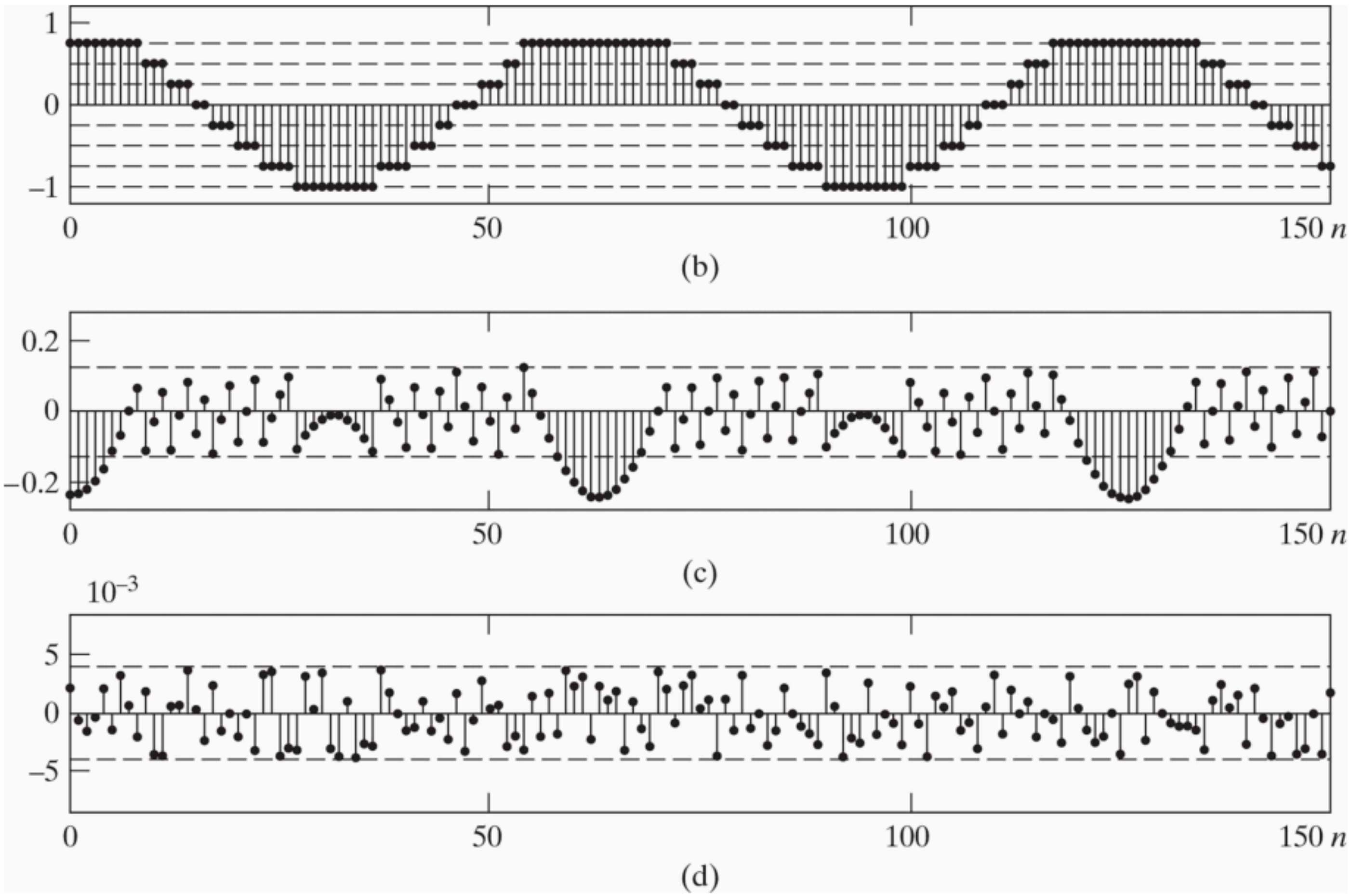
- **Figure 4.57** Example of quantization noise. (a) Unquantized samples of the signal $x[n] = 0.99\cos(n/10)$.



(a)

Quantization Noise

Figure 4.57 (*continued*) (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).



Signal-to-Quantization-Noise Ratio

- For uniform B+1 bits quantizer

$$
\begin{aligned}
SNR_Q &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \\
&= 10\log_{10}\left(\frac{12\cdot 2^{2B}\sigma_x^2}{X_m^2}\right)
\end{aligned}
$$

$$
\mathrm{SNR}_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right)
$$

Quantizer range (X_m)

rms of amp (σ_x)

Signal-to-Quantization-Noise Ratio

$$\mathrm{SNR}_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right)$$

Quantizer range (X_m), rms of amp (σ_x)

- Improvement of 6dB with every bit
- The range of the quantization must be adapted to the rms amplitude of the signal
  - Tradeoff between clipping and noise!
  - Often use pre-amp
  - Sometimes use analog auto gain controller (AGC)

Signal-to-Quantization-Noise Ratio

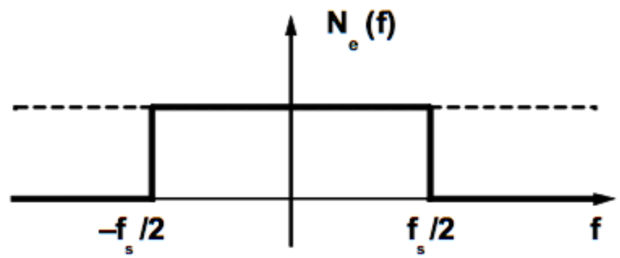
- Assuming full-scale sinusoidal input, we have

$$SQNR = \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{2^B \Delta}{2}\right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2B} = 6.02B + 1.76 \ \ dB$$

B (Number of Bits)	SQNR
8	50dB
12	74dB
16	98dB
20	122dB

Quantization Noise Spectrum

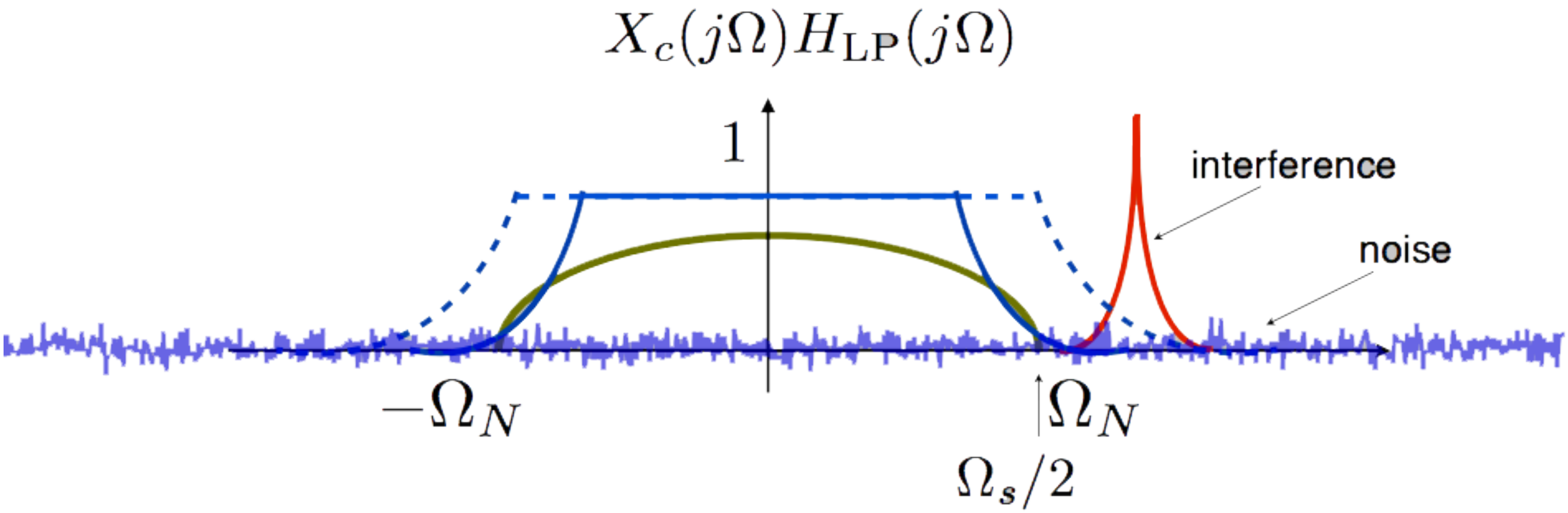
- If the quantization error is "sufficiently random", it also follows that the noise power is uniformly distributed in frequency



$$\frac{\Delta^2}{12} \cdot \frac{1}{f_s}$$

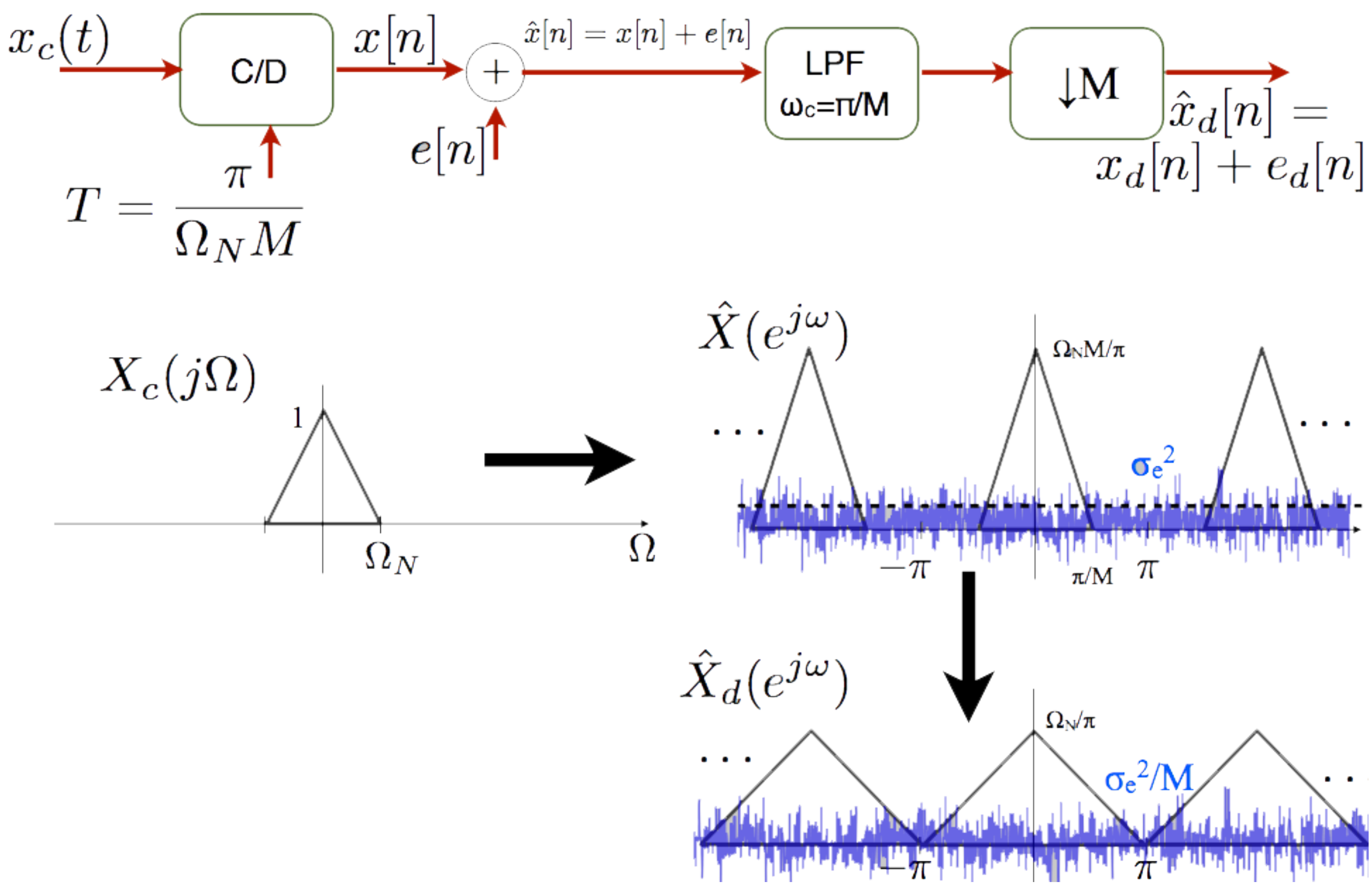
- References
  - W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., pp. 446-72, July 1988.
  - B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," IRE Trans. Circuit Theory, vol. CT-3, pp. 266-76, 1956.

Non-Ideal Anti-Aliasing Filter



- Problem: Hard to implement sharp analog filter
- Solution: Crop part of the signal and suffer from noise and interference

Quantization Noise with Oversampling



Quantization Noise with Oversampling

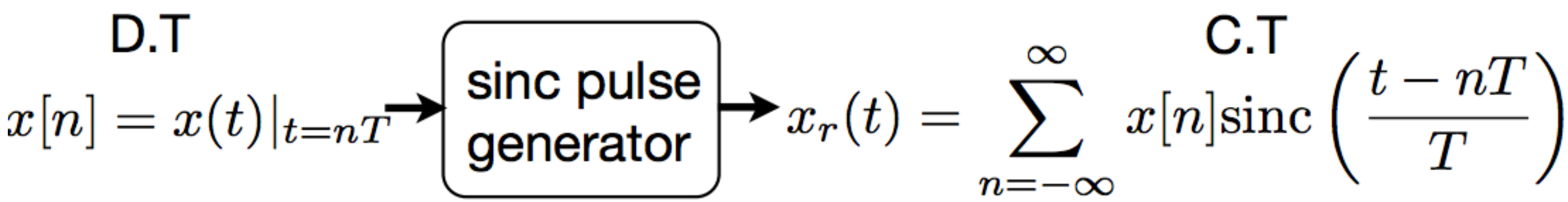
- Energy of $x_d[n]$ equals energy of x[n]
  - No filtering of signal!
- Noise variance is reduced by factor of M

$$\mathrm{SNR}_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) + 10\log_{10} M$$

- For doubling of M we get 3dB improvement, which is the same as 1/2 a bit of accuracy
  - With oversampling of 16 with 8bit ADC we get the same quantization noise as 10bit ADC!

Practical DAC

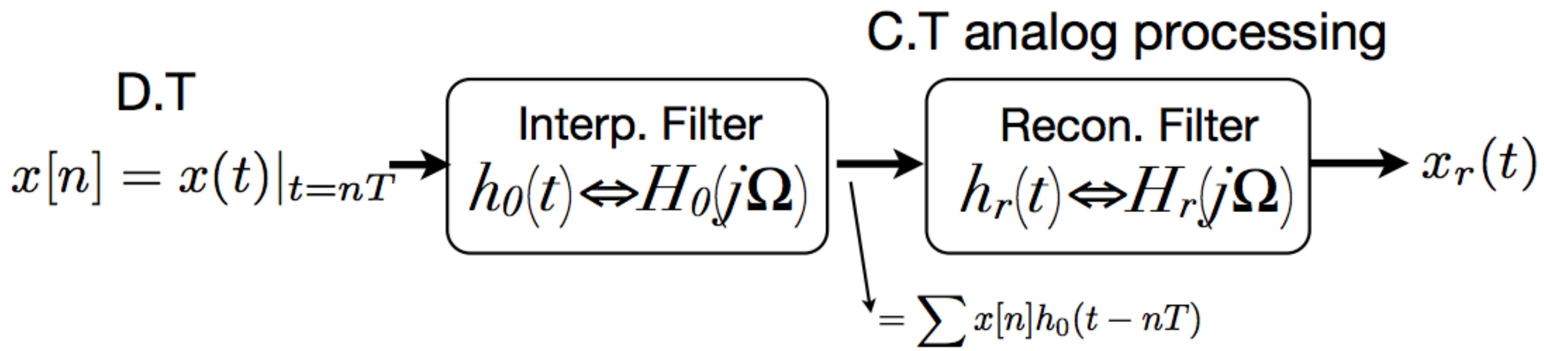
Practical DAC



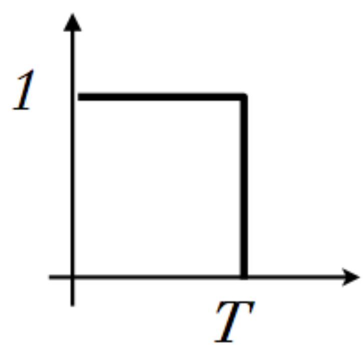
$$x[n] = x(t)|_{t=nT} \rightarrow \text{sinc pulse generator} \rightarrow x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\,\text{sinc}\left(\frac{t-nT}{T}\right)$$

- Scaled train of sinc pulses
- Difficult to generate sinc → Too long!

Practical DAC



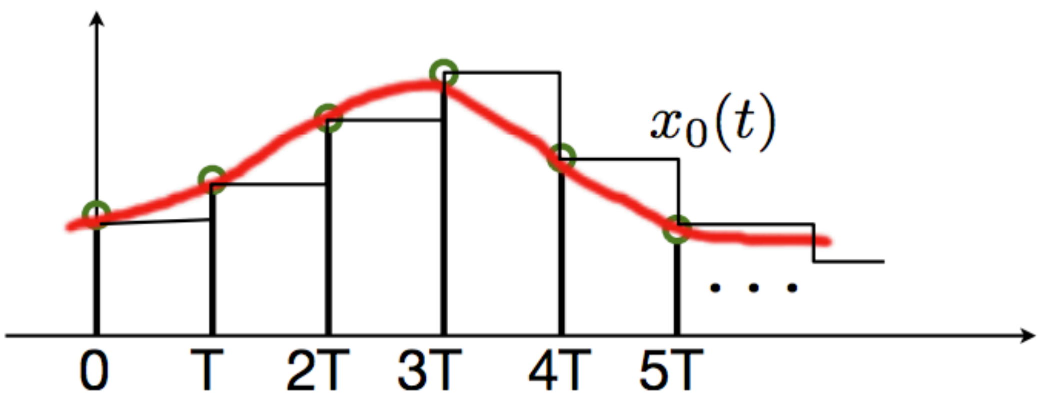
- $h_0(t)$ is finite length pulse → easy to implement
- For example: zero-order hold



$$H_0(j\Omega) = Te^{-j\Omega\frac{T}{2}}\text{sinc}(\frac{\Omega}{\Omega_s})$$

Practical DAC

Zero-Order-Hold interpolation



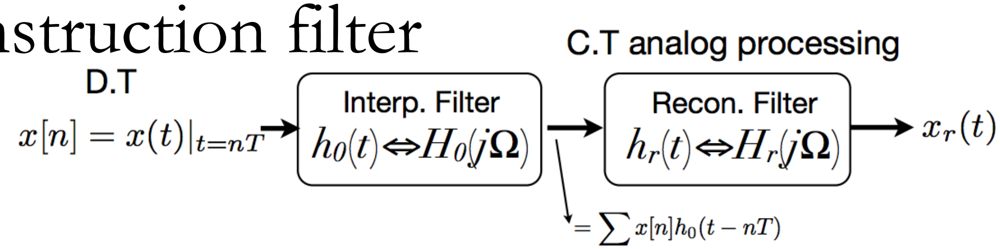
$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t - nT) = h_0(t) * x_s(t)$$

Taking a FT:

$$\begin{aligned} X_(j\Omega) &= H_0(j\Omega)X_s(j\Omega) \\ &= H_0(j\Omega)\frac{1}{T}\sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s)) \end{aligned}$$

Practical DAC

- Output of the reconstruction filter

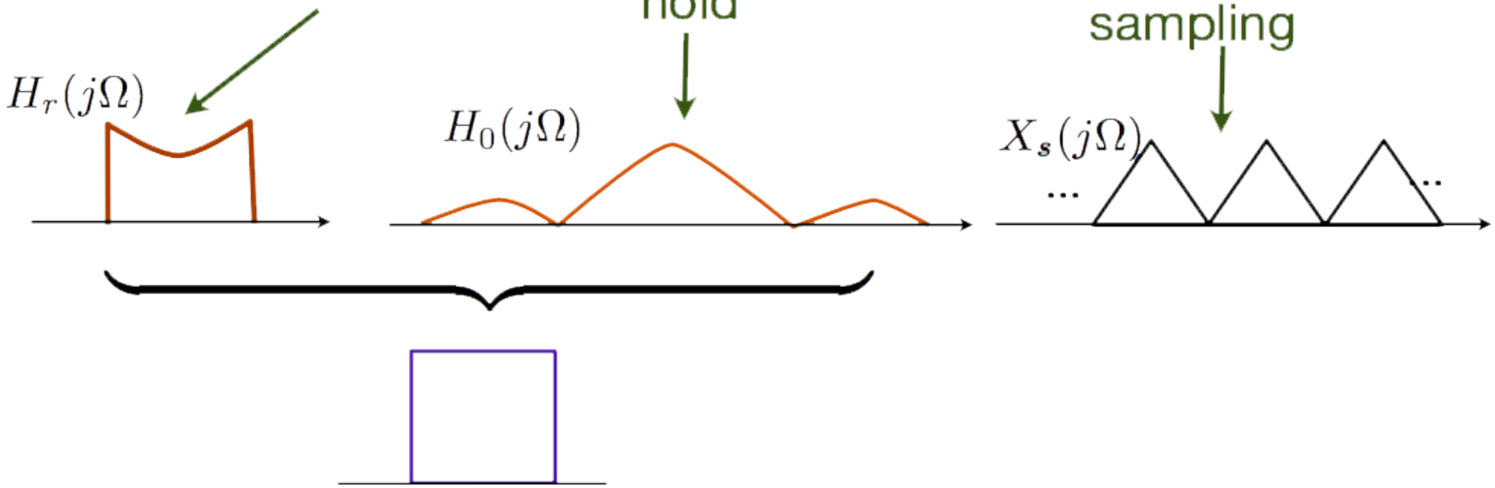


$$
\begin{aligned}
X_r(j\Omega) &= H_r(j\Omega) \cdot H_0(j\Omega) \cdot X_s(j\Omega) \\
&= \underbrace{H_r(j\Omega)}_{\text{recon filter}} \cdot \underbrace{T e^{-j\Omega \frac{T}{2}} \operatorname{sinc}\left(\frac{\Omega}{\Omega_s}\right)}_{\text{from zero-order hold}} \cdot \underbrace{\frac{1}{T} \sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s))}_{\text{Shifted copies from sampling}}
\end{aligned}
$$

recon filter

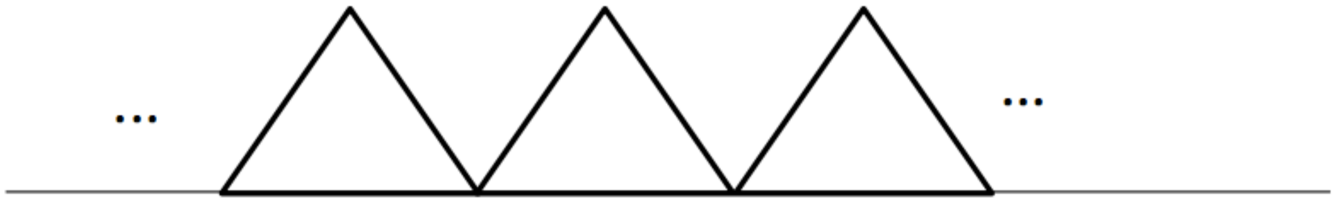
from zero-order hold

Shifted copies from sampling



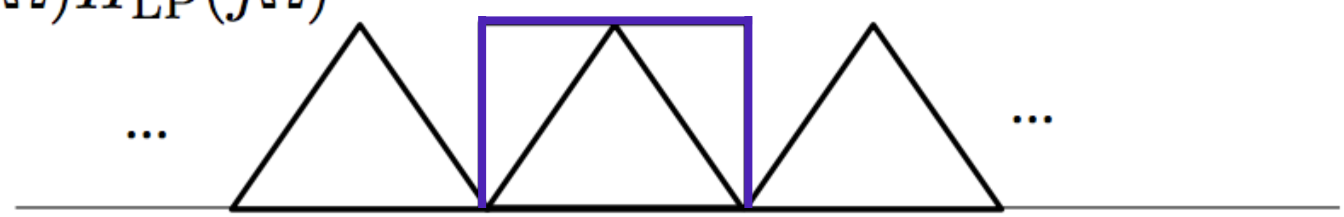
Practical DAC

$X_s(j\Omega)$

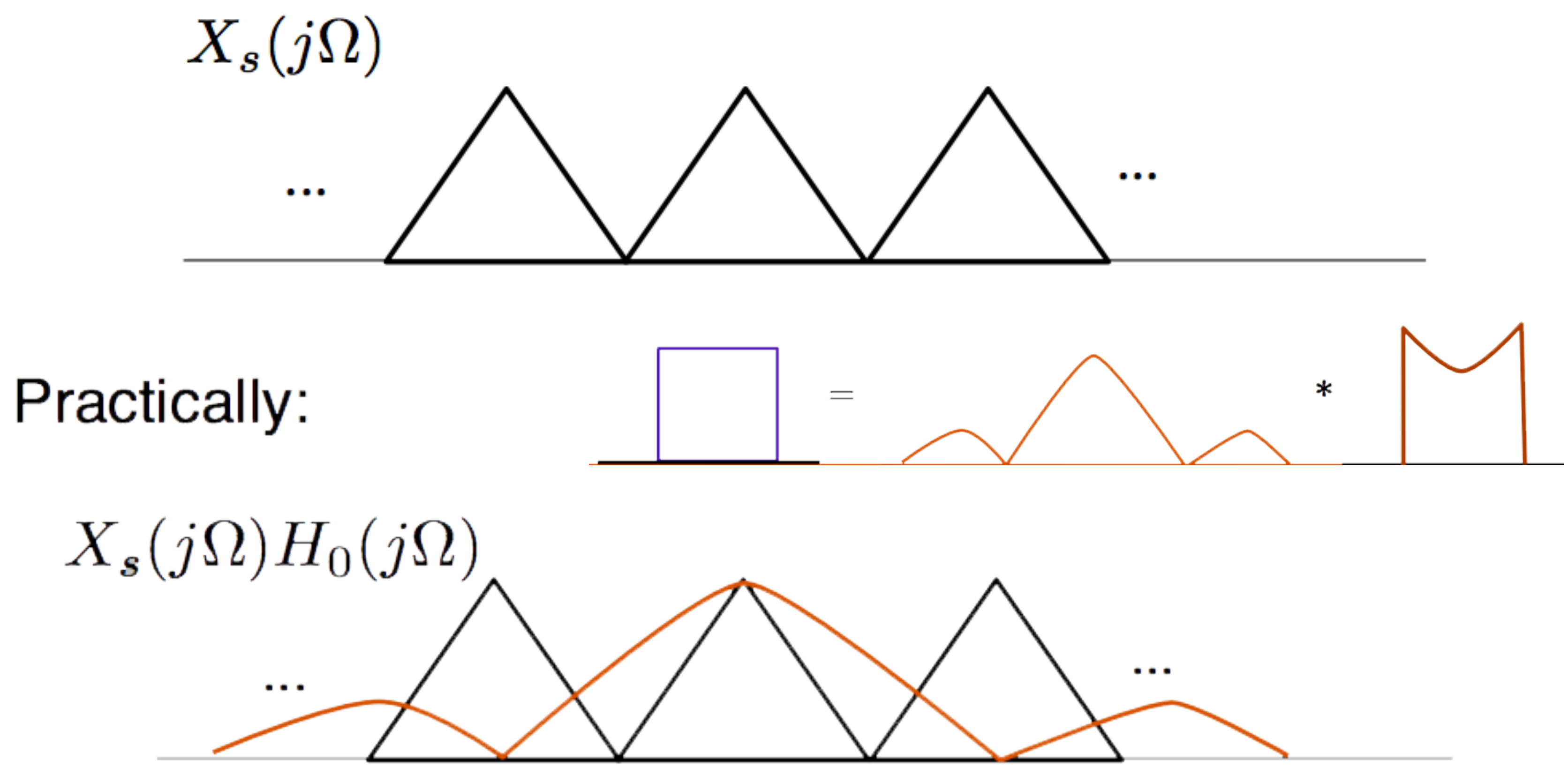


Ideally:

$X_s(j\Omega)H_{\mathrm{LP}}(j\Omega)$

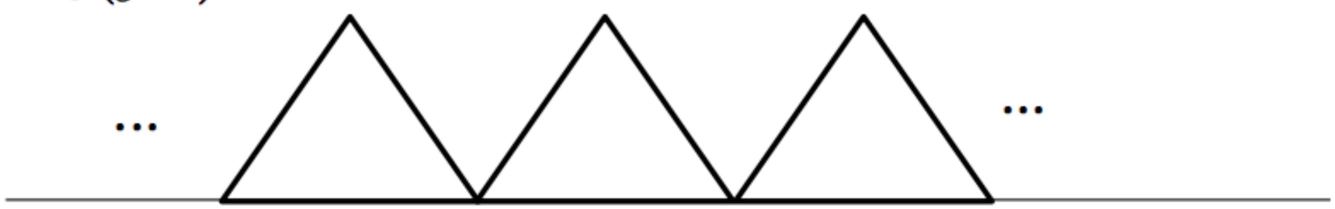


Practical DAC



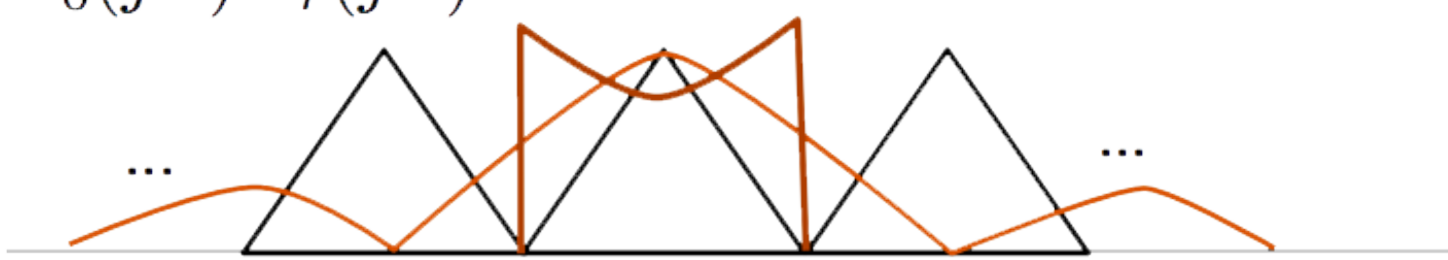
Practical DAC

$X_s(j\Omega)$

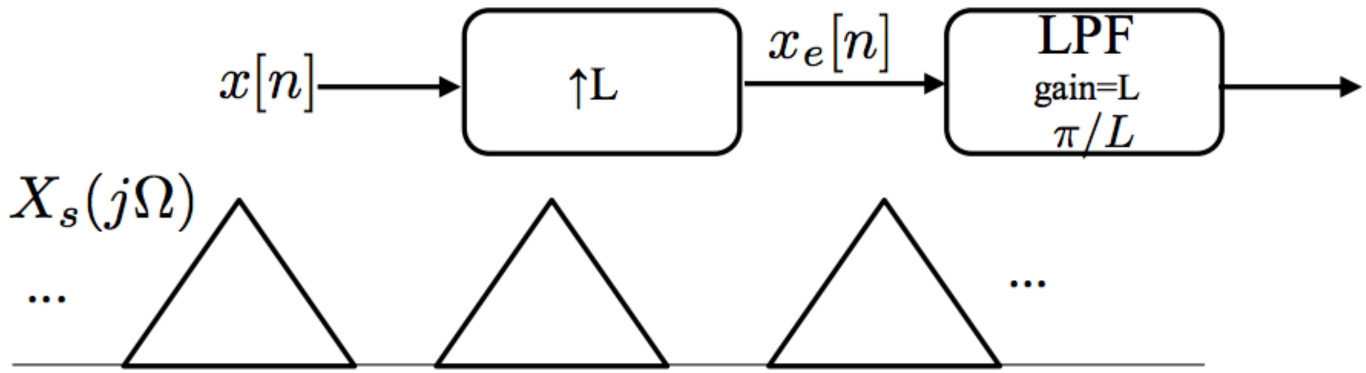


Practically:

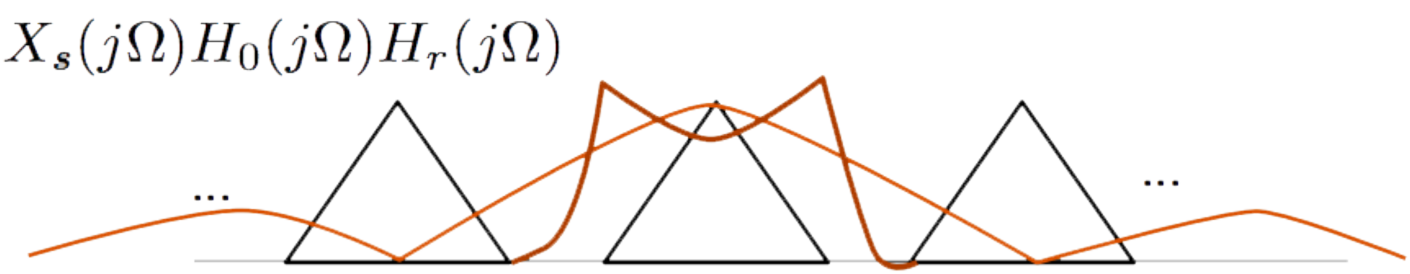
$X_s(j\Omega)H_0(j\Omega)H_r(j\Omega)$



Practical DAC with Upsampling

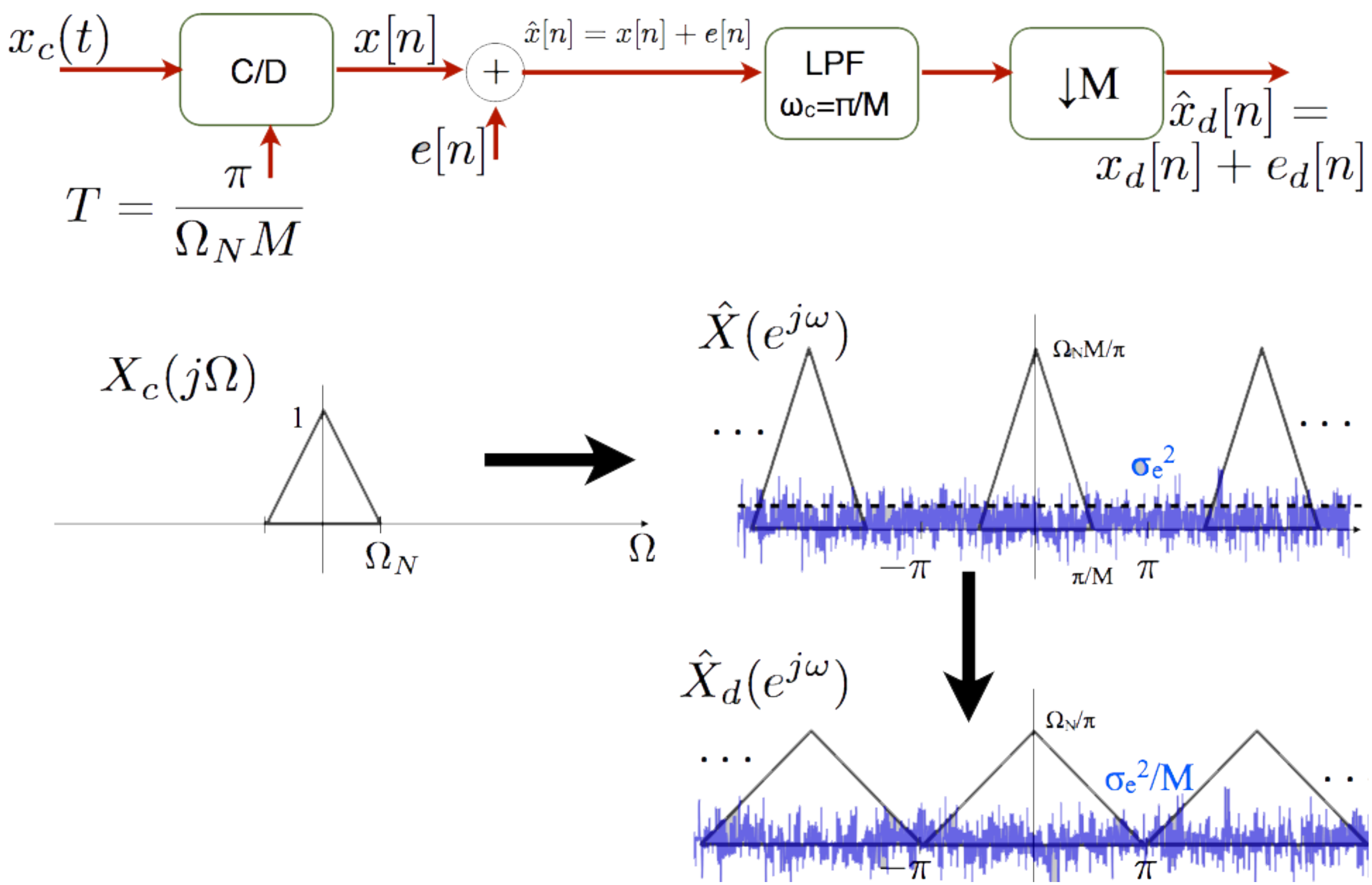


Practically:



Noise Shaping

Quantization Noise with Oversampling



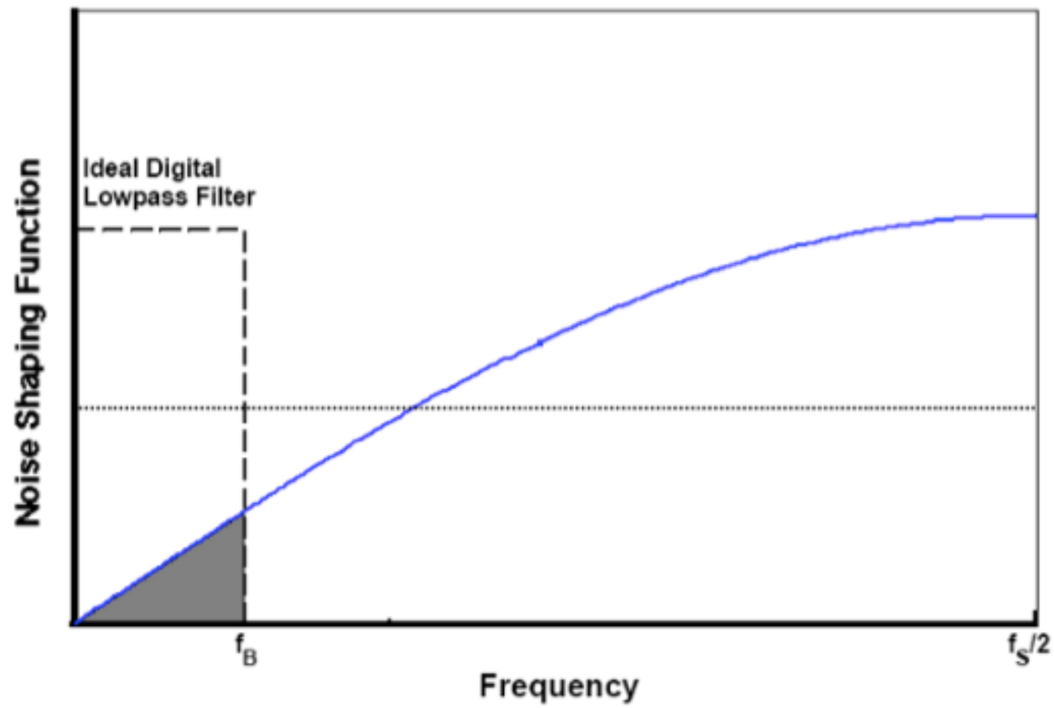
Quantization Noise with Oversampling

- Energy of $x_d[n]$ equals energy of x[n]
  - No filtering of signal!
- Noise variance is reduced by factor of M

$$\mathrm{SNR}_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) + 10\log_{10} M$$

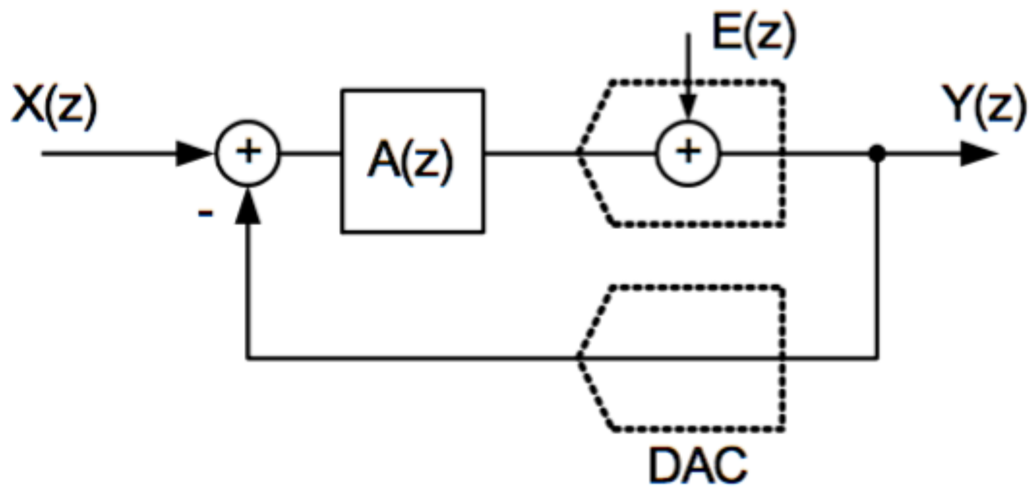
- For doubling of M we get 3dB improvement, which is the same as 1/2 a bit of accuracy
  - With oversampling of 16 with 8bit ADC we get the same quantization noise as 10bit ADC!

Noise Shaping



- Idea: "Somehow" build an ADC that has most of its quantization noise at high frequencies
- Key: Feedback

Noise Shaping Using Feedback



$$Y(z) = E(z) + A(z)X(z) - A(z)Y(z)$$

$$= E(z)\frac{1}{1+A(z)} + X(z)\frac{A(z)}{1+A(z)}$$

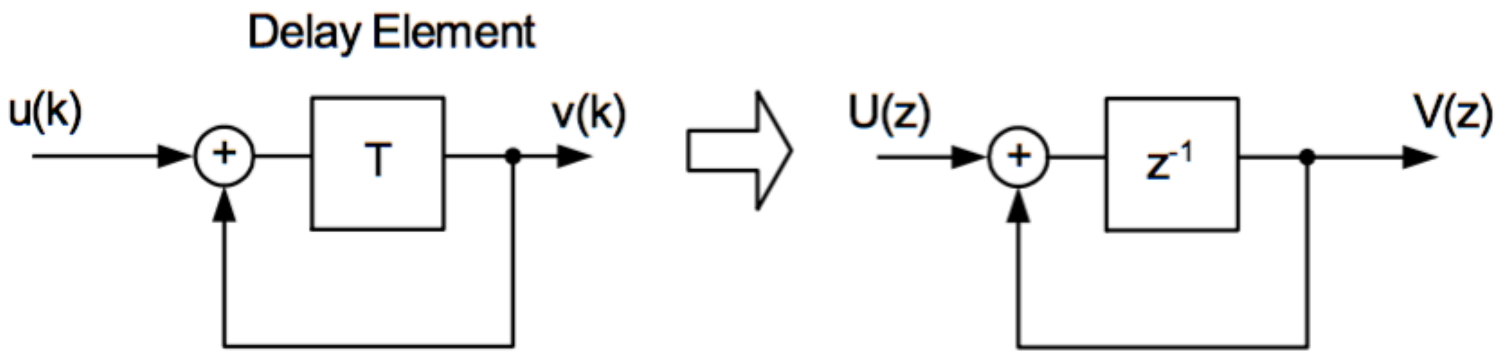
$$= E(z)\underbrace{H_E(z)}_{\substack{\text{Noise}\\\text{Transfer}\\\text{Function}}} + X(z)\underbrace{H_X(z)}_{\substack{\text{Signal}\\\text{Transfer}\\\text{Function}}}$$

Noise Shaping Using Feedback

$$Y(z) = E(z)\underbrace{\frac{1}{1+A(z)}}_{\substack{Noise\\Transfer\\Function}} + X(z)\underbrace{\frac{A(z)}{1+A(z)}}_{\substack{Signal\\Transfer\\Function}}$$

- Objective
  - Want to make STF unity in the signal frequency band
  - Want to make NTF "small" in the signal frequency band
- If the frequency band of interest is around DC ($0 \ldots f_B$) we achieve this by making $|A(z)| >> 1$ at low frequencies
  - Means that NTF << 1
  - Means that STF ≅ 1

Discrete Time Integrator



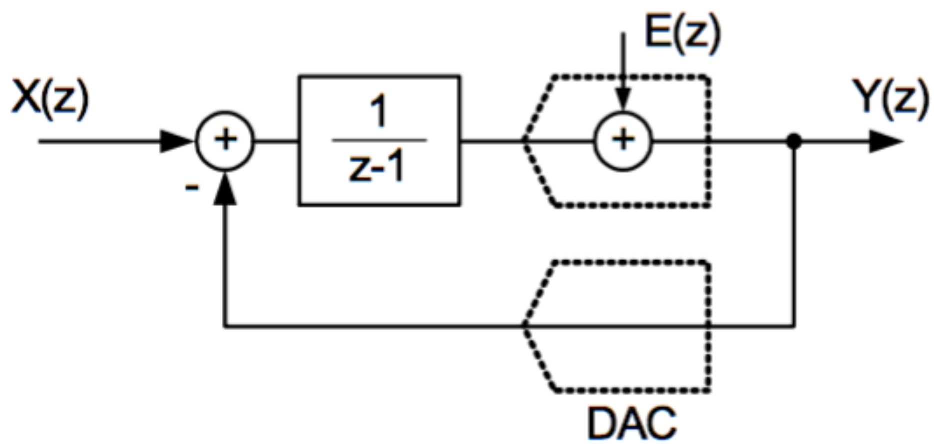
$$v(k) = u(k-1) + v(k-1)$$

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

$$\frac{V(z)}{U(z)} = \frac{z^{-1}}{1-z^{-1}} = \frac{1}{z-1} \qquad z = e^{j\omega T}$$

- "Infinite gain" at DC (ω=0, z=1)

First Order Sigma-Delta Modulator



$$Y(z) = E(z)\frac{1}{1+\frac{1}{z-1}} + X(z)\frac{\frac{1}{z-1}}{1+\frac{1}{z-1}}$$

$$= E(z)\left(1-z^{-1}\right) + X(z)z^{-1}$$

- Output is equal to delayed input plus filtered quantization noise

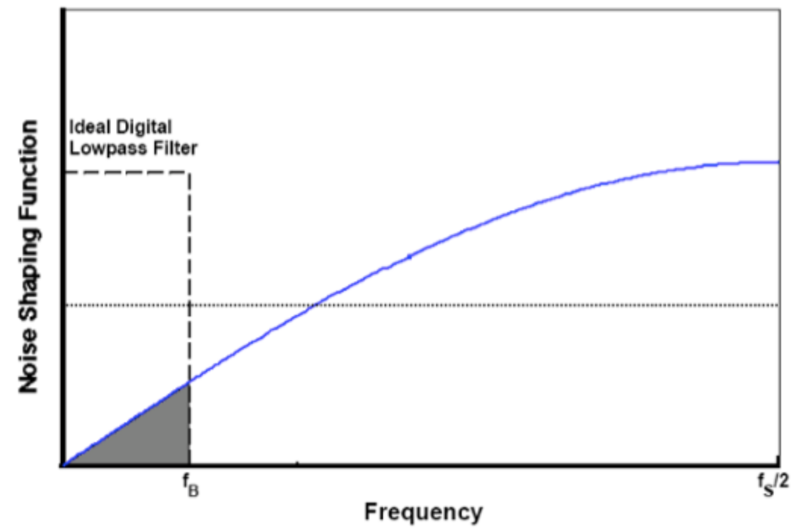
NTF Frequency Domain Analysis

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

$$H_e(j\omega) = \left(1 - e^{-j\omega T}\right) = 2e^{-j\omega T/2}\left(\frac{e^{j\omega T/2} - e^{-j\omega T/2}}{2}\right)$$

$$= 2e^{-j\frac{\omega T}{2}}\left(j\,sin\left(\frac{\omega T}{2}\right)\right) = 2\,sin\left(\frac{\omega T}{2}\right)e^{-j\frac{\omega T - \pi}{2}}$$

$$\left|H_e(f)\right| = 2\left|sin(\pi f T)\right| = 2\left|sin\left(\pi\frac{f}{f_s}\right)\right|$$



- "First order noise Shaping"
  - Quantization noise is attenuated at low frequencies, amplified at high frequencies

In-Band Quantization Noise

- Question: If we had an ideal digital lowpass, what is the achieved SQNR as a function of oversampling ratio?
- Can integrate shaped quantization noise spectrum up to f_B and compare to full-scale signal

$$P_{qnoise} = \int_0^{f_B} \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot \left[2\sin\left(\pi \frac{f}{f_s}\right)\right]^2 df$$

$$\cong \int_0^{f_B} \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot \left[2\pi \frac{f}{f_s}\right]^2 df$$

$$\cong \frac{\Delta^2}{12} \cdot \frac{\pi^2}{3} \left[\frac{2 f_B}{f_s}\right]^3 = \frac{\Delta^2}{12} \cdot \frac{\pi^2}{3} \frac{1}{M^3}$$

In-Band Quantization Noise

- Assuming a full-scale sinusoidal signal, we have

$$SQNR \cong \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{(2^B-1)\Delta}{2}\right)^2}{\frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\frac{1}{M^3}} = 1.5\times(2^B-1)^2\times\underbrace{\frac{3}{\pi^2}\times M^3}_{\substack{\text{Due to noise}\\ \text{shaping \&}\\ \text{digital filter}}}$$

$$\cong 1.76 + 6.02B - 5.2 + 30\log(M) \quad [dB] \quad (\text{for large } B)$$

- Each 2x increase in M results in 8x SQNR improvement
  - Also added ½ bit resolution

Digital Noise Filter

- Increasing M by 2x, means 3-dB reduction in quantization noise power, and thus 1/2 bit increase in resolution
  - "1/2 bit per octave"
- Is this useful?
- Reality check
  - Want 16-bit ADC, f_B=1MHz
  - Use oversampled 8-bit ADC with digital lowpass filter
  - 8-bit increase in resolution necessitates oversampling by 16 octaves

$$f_s \geq 2 \cdot f_B \cdot M = 2 \cdot 1MHz \cdot 2^{16}$$
$$\geq 131GHz$$

SQNR Improvement

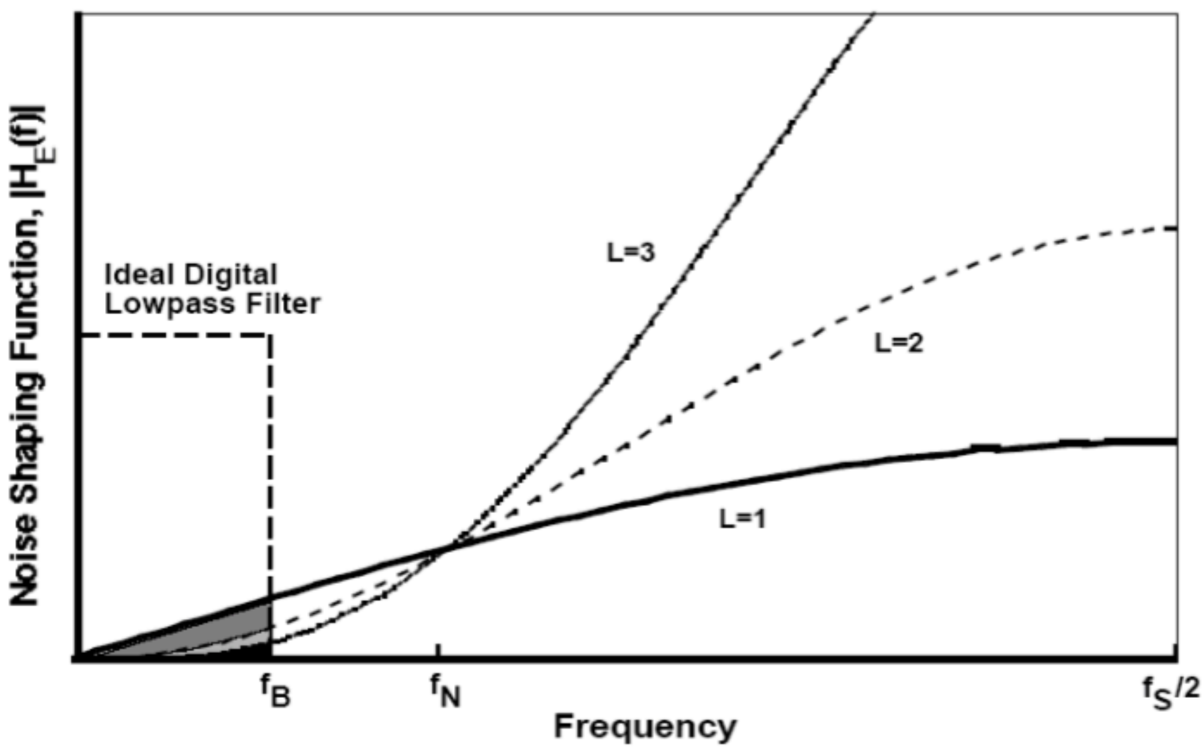
- Example Revisited
  - Want16-bit ADC, f_B=1MHz
  - Use oversampled 8-bit ADC, first order noise shaping and (ideal) digital lowpass filter
    - SQNR improvement compared to case without oversampling is -5.2dB+30log(M)
  - 8-bit increase in resolution (48 dB SQNR improvement) would necessitate M≅60 → f_S=120MHz
- Not all that bad!

M	SQNR improvement
16	31dB (~5 bits)
256	67dB (~11 bits)
1024	85dB (~14 bits)

Higher Order Noise Shaping

- L^{th} order noise transfer function

$$H_E(z) = \left(1 - z^{-1}\right)^L$$



Big Ideas

- Multi-Rate Filter Banks
  - Quadrature mirror filters eliminate aliasing from non-ideal filters.
- Data Converters
  - Oversampling to reduce interference and quantization noise → increase ENOB (effective number of bits)
  - Practical DACs use practical interpolation and reconstruction filters with oversampling
- Noise Shaping
  - Use feedback to reduce oversampling factor

Admin

- HW 5 due Friday
- Signals and Systems review resources posted