Analog-to-Digital Converters

Richard Heusdens

EE2S31

TU Delft

Digital Signal Processing

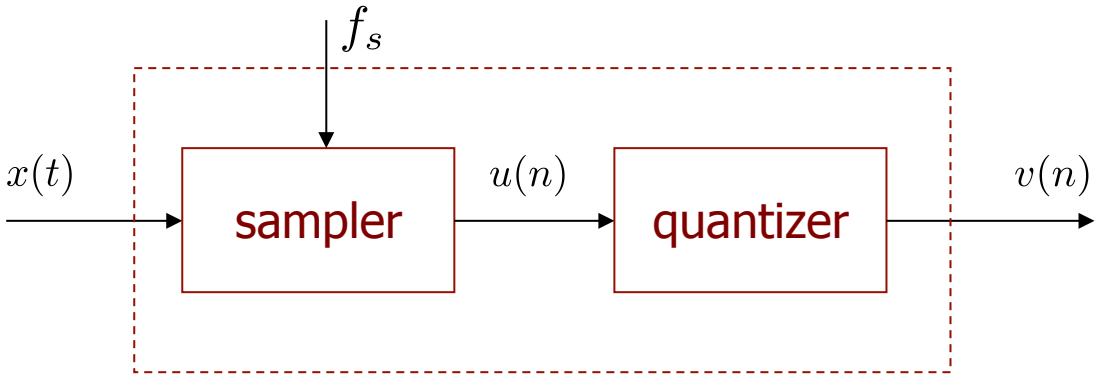
In its most general form, *digital signal processing* (DSP) refers to processing of analog signals by means of discrete-time (discrete-space) operations implemented on digital hardware.

From a system point of view, DSP is concerned with mixed systems:

- the input and output signals are analog
- the processing is done on the equivalent digital signals

Analog-to-Digital Converter

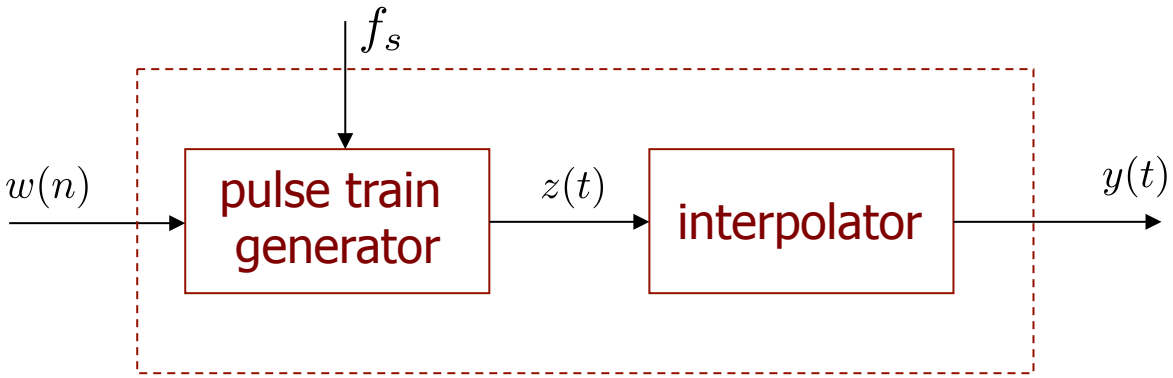
Two-step approach:



- Sampler: $u(n) = x(nT_s)$ where $T_s = 1/f_s$, the sampling period
- Quantizer: $v(n) = (Qu)(n)$, where Q is a (nonlinear) mapping from intervals of the real line (quantization cells) to reproduction levels

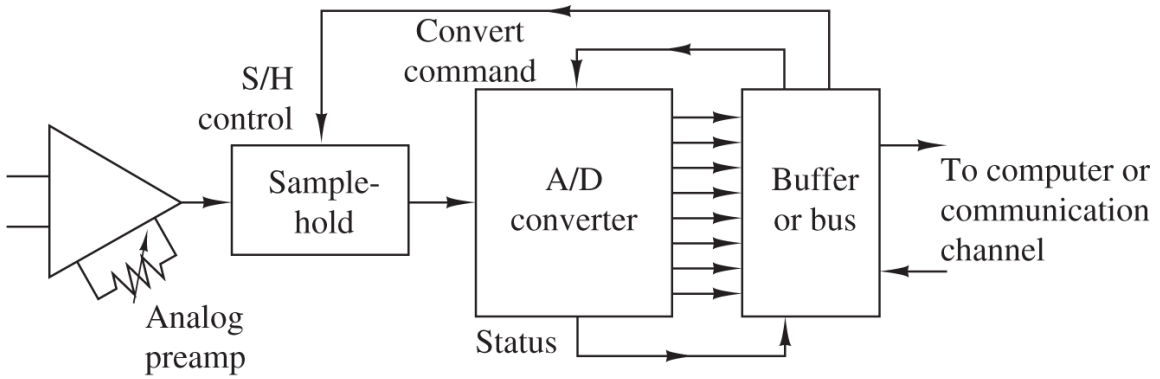
Digital-to-Analog Converter

Two-step approach:

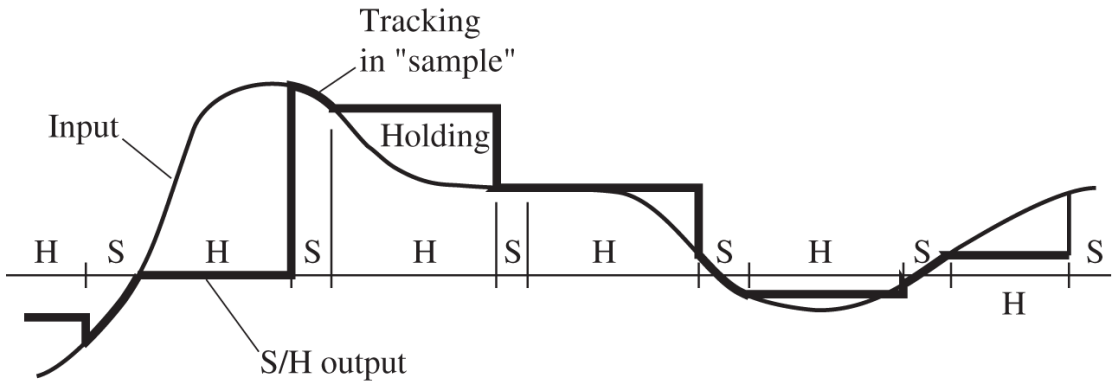


- The pulse-train generator transforms the sequence of numbers $w(n)$ into a sequence of scaled, analog pulses (spaced $T_s = 1/f_s$ seconds apart)
- The interpolator removes high-frequency components in z (via low-pass filtering) to produce a smooth analog output signal

Basic Elements A/D Converter



(a)



(b)

Spectral Analysis

A/D converter:

1) quantization

2) coding

Quantization is a non-linear and non-invertible process that maps a given amplitude $x(n) = x_a(nT_s)$ at time $t = nT_s$ into an amplitude x_k taken from a finite set of values (alphabet)

Coding assigns a unique binary number (code) to each and every quantization level. This process is reversible (lossless)

Quantization

An L-level quantizer is characterized by a set of $L+1$ *decision levels* or *decision thresholds* $x_1 < x_1 < \cdots < x_{L+1}$ and a set $\hat{\mathcal{X}} = \{\hat{x}_k,\, k = 1, \ldots, L\}$ such that $\hat{x} = \hat{x}_k$ if and only if $x_k \leq x < x_{k+1}$, where $x_1 = -\infty$ and $x_{L+1} = \infty$.

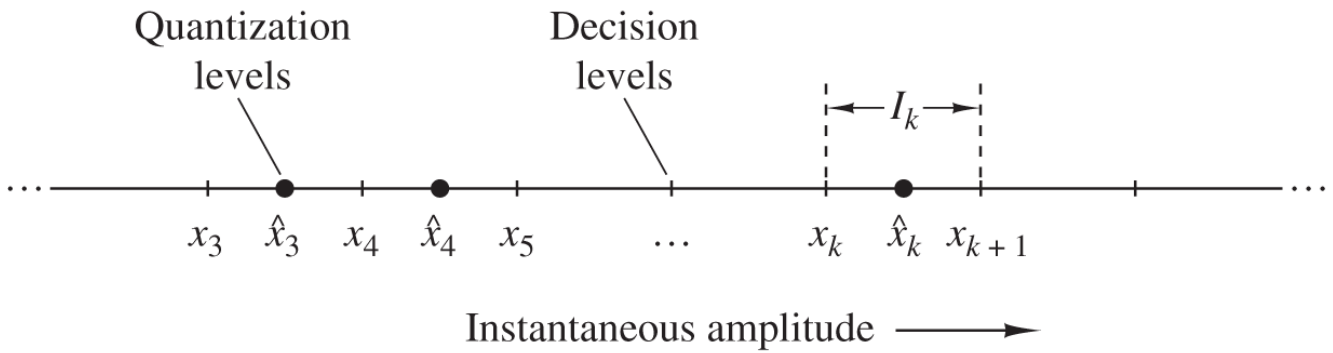
The numbers $\hat{x}_k$ are called the *reconstruction values* or *quantization levels* and the intervals $I_k = [x_k,\, x_{k+1})$ are usually referred to as the *decision intervals* or *quantization cells*.

The map $Q : \mathcal{X} \mapsto \hat{\mathcal{X}}$ is given by

$$Q(x) = \hat{x}_k \quad \text{for } x \in I_k, \qquad k = 1, \ldots, L,$$

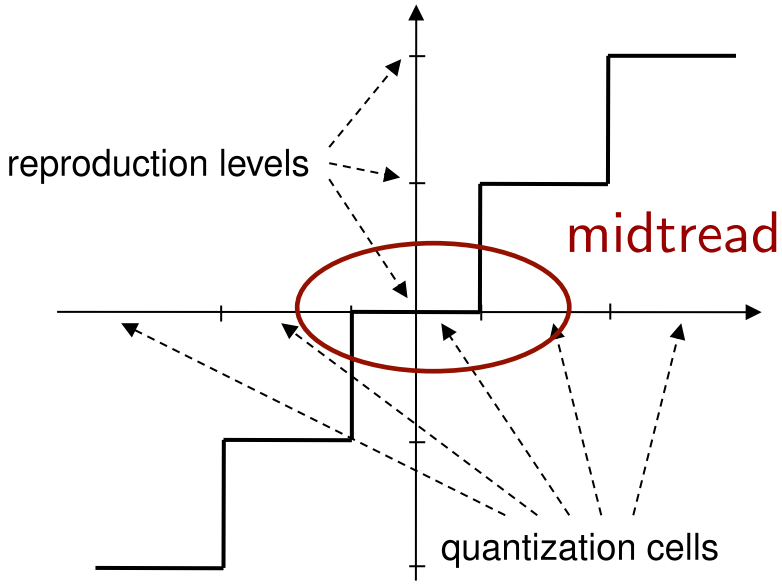
is a staircase function by definition

Quantization

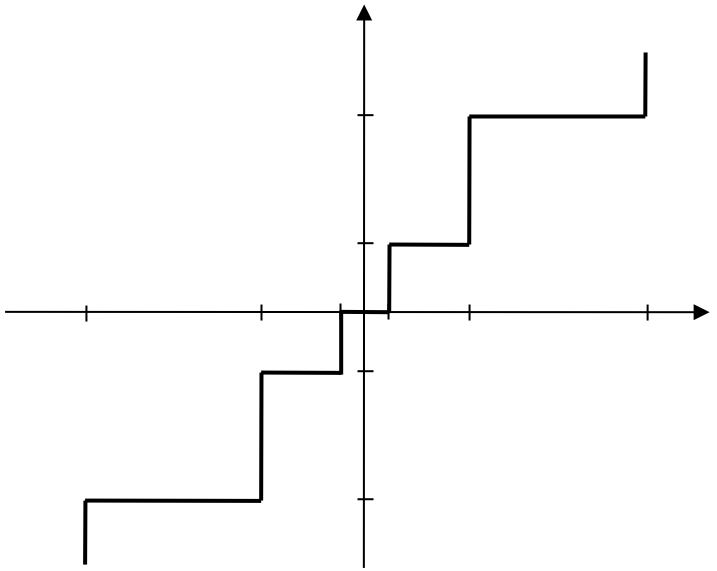


Quantization

uniform quantization



nonuniform quantization



Quantization

Uniform (or linear) quantizer:

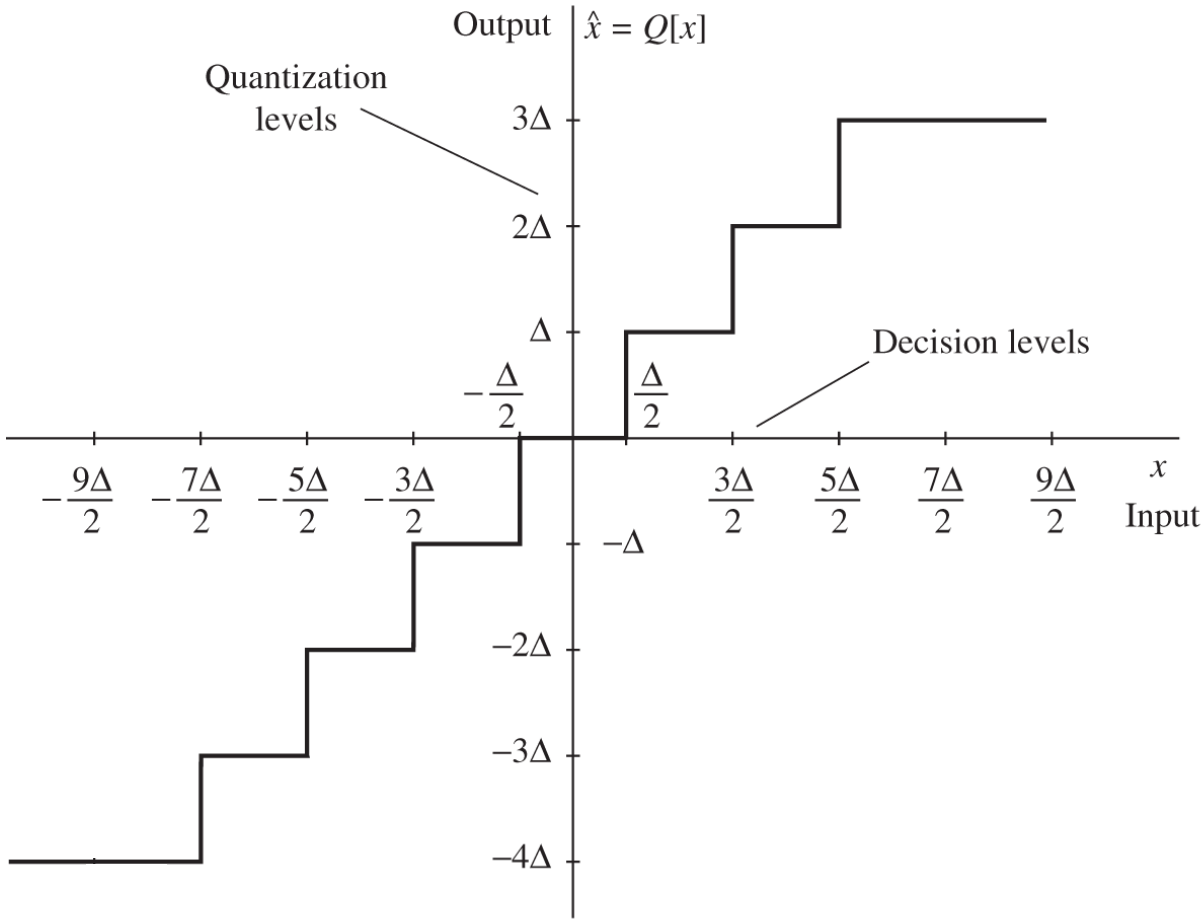
- $x_{k+1} - x_k = \Delta$
- $\hat{x}_k = (x_{k+1} - x_k)/2 \quad \Rightarrow \quad \hat{x}_{k+1} - \hat{x}_k = \Delta$

Δ is called the *step size* of the quantizer

The quantization error $z(n) = x(n) - \hat{x}(n)$ satisfies

$$-\frac{\Delta}{2} \leq z(n) < \frac{\Delta}{2}$$

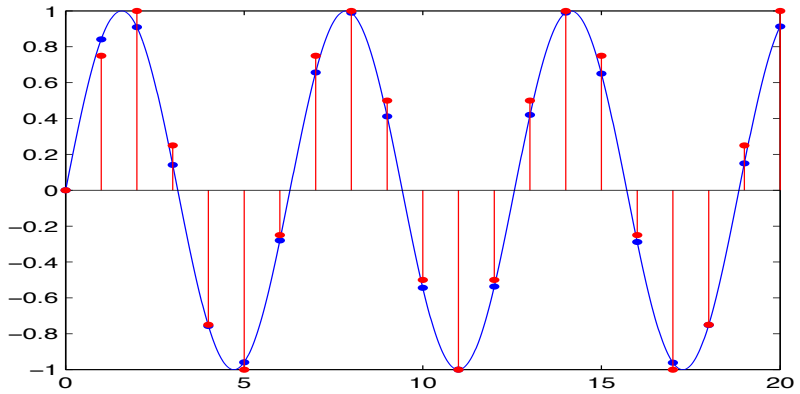
Quantization



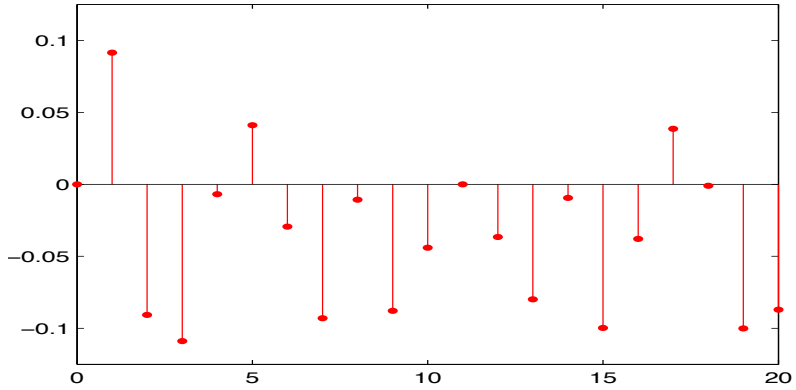
Quantization

Example:

sampled signal (original (blue) and quantized (red))

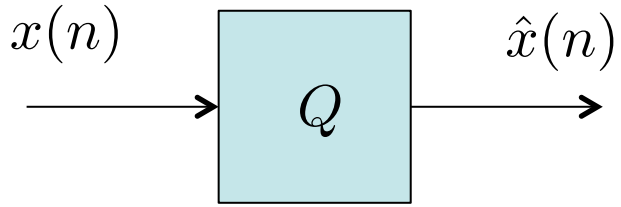


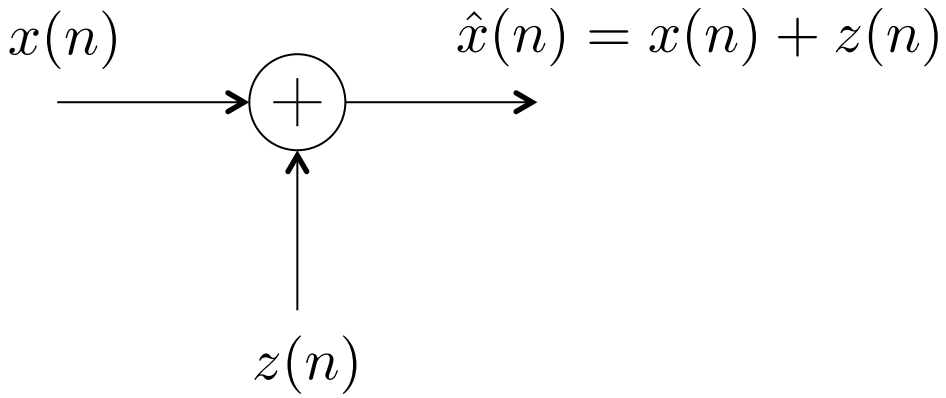
quantization error



Quantization

Mathematical model of quantization:



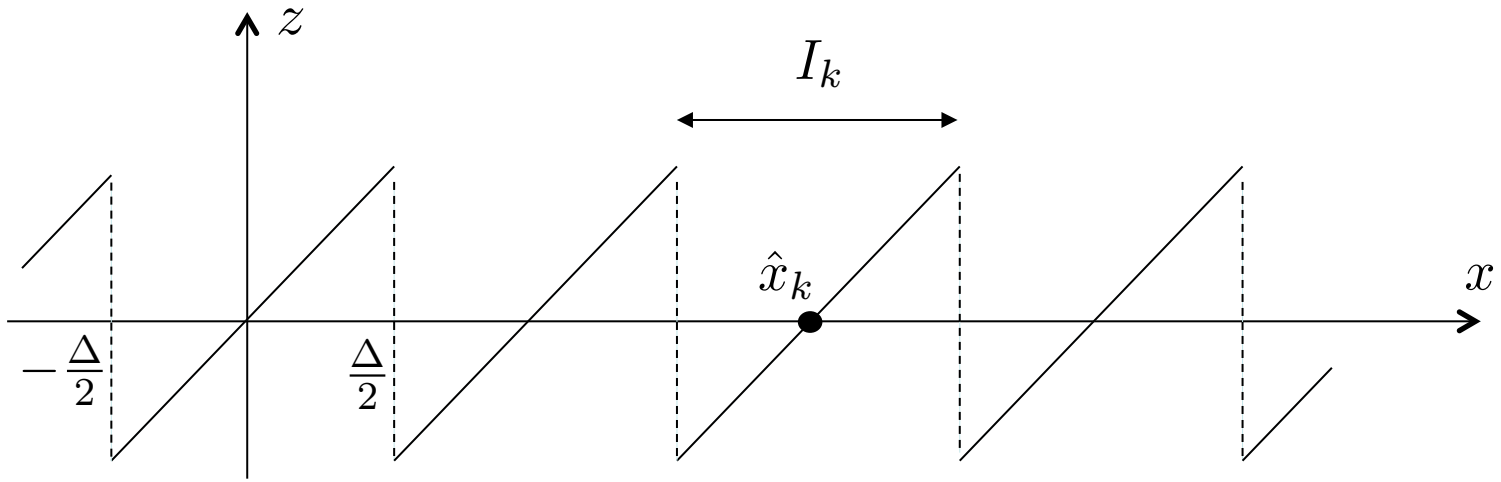


Quantization

Assumptions:

- The error z is uniformly distributed over the interval $-\Delta/2 \leq z(n) < \Delta/2$
- The input signal x is a realization of a zero-mean wide-sense stationary process
- The quantization noise is "white" (uncorrelated)
- The quantization noise is uncorrelated with the input x

Quantization Error

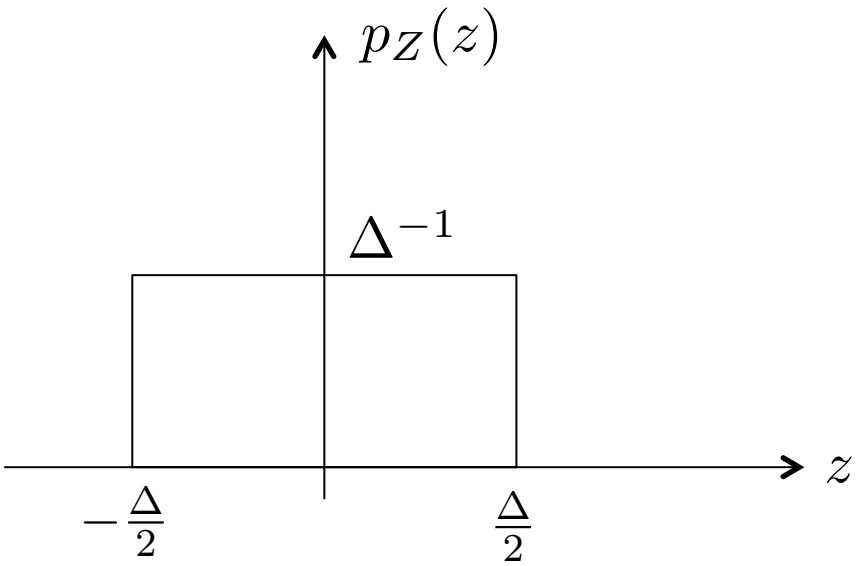


Distribution of the quantization noise: $Z = X - Q(X)$

approximately

$$p_Z(z) = \sum_{k=1}^{L} p_X(z + \hat{x}_k) \sim U\left(0, \frac{\Delta^2}{12}\right)$$

Quantization Error



$$\sigma_Z^2 = \int_{-\Delta/2}^{\Delta/2} z^2 \Delta^{-1} dz = \Delta^{-1} \left. \frac{1}{3} z^3 \right|_{-\Delta/2}^{\Delta/2} = \frac{\Delta^2}{12}$$

Quantization

Signal-to-quantization-noise-ratio (SQNR):

$$SQNR = 10\log_{10}\left(\frac{\sigma_X^2}{\sigma_Z^2}\right) = 10\log_{10}\left(\frac{12\sigma_X^2}{\Delta^2}\right)$$

Let R denote the (input) range of the quantizer and represent the quantized values with $B+1$ bits. We then have

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

and thus

every additional bit improves the SQNR by 6 dB

$$SQNR = 6.02B + 16.81 + 20\log_{10}\left(\frac{\sigma_X}{R}\right)$$

Coding

Fixed-point representation:

- sign-magnitude

$$x = (b_K \cdots b_0\, b_{-1} \cdots b_{-L})_{\mathrm{SM}} = (-1)^{b_K} \times \sum_{i=-L}^{K-1} b_i 2^i$$

- two's-complement

$$x = (b_K \cdots b_0\, b_{-1} \cdots b_{-L})_{\mathrm{2C}} = -b_K 2^K + \sum_{i=-L}^{K-1} b_i 2^i$$

Coding

Example:

bits	SM	2C
011	3	3
010	2	2
001	1	1
000	0	0
111	-3	-1
110	-2	-2
101	-1	-3
100	0	-4

$\longrightarrow$ $-4 + 1 = -3$

$2^K = 2^2 = 4$

Coding

Properties:

- resolution 2^{-L}
- $2^{B+1} = 2^{K+L+1}$ levels (one less for sign-magnitude)

Range:

- sign-magnitude $[-2^K + 2^{-L}, 2^K - 2^{-L}]$
- two's complement $[-2^K, 2^K - 2^{-L}]$

Most fixed-point processors use two's complement arithmetic. With two's complement representation, addition can be down bit-by-bit. In the sign-magnitude format this is more complex.

Coding

Addition:

SM:
$$\begin{array}{r} 001 \\ 111 \\ \hline 000 \end{array} \oplus \begin{array}{r} (1) \\ (-3) \\ \hline (0) \end{array}$$

2C:
$$\begin{array}{r} 001 \\ 101 \\ \hline 110 \end{array} \oplus \begin{array}{r} (1) \\ (-3) \\ \hline (-2) \end{array}$$

- Two's complement arithmetic is arithmetic modulo 2^{K+1}
- If the final sum of numbers is within the range, the sum will be computed correctly, even when individual partial sums result in overflows

In general, multiplication of two fixed-point numbers each B bits in length results in a product of $2B$ bits in length

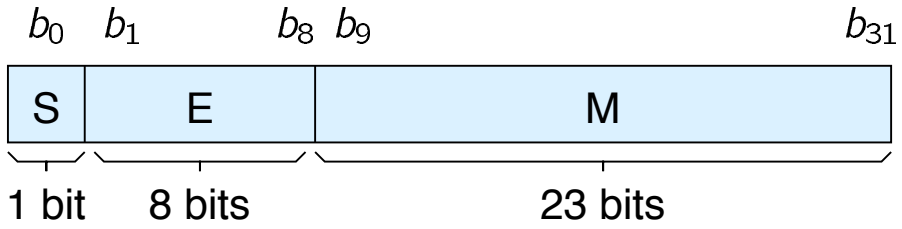
Coding

Floating-point representation:

$$x = M \times 2^E,$$

where $0.5 \leq M < 1$ the mantissa (or significand) and E the exponent

IEEE 754 floating point standaard:



$$x = (-1)^s \times (M)_2 \times 2^{E-127}$$

Coding

Example:

$$X_1 = 5 \quad \Rightarrow \quad M_1 = 0.101000, E_1 = 011$$

$$X_2 = \frac{3}{8} \quad \Rightarrow \quad M_2 = 0.110000, E_2 = 101$$

SM

Multiplication:

$$\begin{aligned} X_1 X_2 &= M_1 M_2 \times 2^{E_1+E_2} \\ &= (0.011110)_2 \times 2^{(010)_2} \\ &= (0.111100)_2 \times 2^{(001)_2} \quad (0.9375 \times 2 = 1.8750) \end{aligned}$$

Coding

Example:

$$X_1 = 5 \quad \Rightarrow \quad M_1 = 0.101000, E_1 = 011$$

$$X_2 = \frac{3}{8} \quad \Rightarrow \quad M_2 = 0.110000, E_2 = 101$$

Addition: when $E_2 = E_1 = E$ we can add the two numbers

$$\begin{aligned} X_1 + X_2 &= (M_1 + M_2) \times 2^E \\ &= ((0.101000)_2 + (0.000011)_2) \times 2^{(011)_2} \\ &= (0.101011)_2 \times 2^{(011)_2} \qquad (0.671875 \times 2^3 = 5.3750) \end{aligned}$$

Scaling (shifting) will, in general, result in a loss of precision

Coding

Properties:

- larger dynamic range than fixed point
- variable resolution
- requires more expensive hardware
- extensions: $x = 0\ (E = 0, M = 0),\ x = \infty\ (E = 256, M = 0),\ x = \mathrm{NaN}\ (E = 256, M \neq 0)$

Quantization of Filter Coefficients

Transfer function:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{M} a_k z^{-k}}$$

After quantization ($\hat{a}_k = a_k + \Delta a_k,\ \hat{b}_k = b_k + \Delta b_k$)

$$\hat{H}(z) = \frac{\sum_{k=0}^{M} \hat{b}_k z^{-k}}{1 - \sum_{k=1}^{M} \hat{a}_k z^{-k}} = H(z) + \Delta H(z)$$

Quantization of Filter Coefficients

As a consequence, the pole and zero positions will change:

- $\hat{p}_k = p_k + \Delta p_k$
- $\hat{z}_k = z_k + \Delta z_k$

We have

$$\Delta p_k \approx -\sum_{l=1}^{N} \frac{p_k^{N-l} \Delta a_l}{\prod_{l \neq k} p_k - p_l}$$

Closely spaced poles can give rise to large errors. Similar results hold for the zeros.

Quantization of Filter Coefficients

Error in Δp_k can be minimized by maximizing $|p_k - p_l|$:

- realize high-order filter with either one or two-pole filter sections
- in general, one-pole filter sections require complex-valued arithmetic
- not necessary by combining complex-conjugated poles (and zeros)
- since complex-conjugated poles are usually sufficiently far apart, the perturbation error Δp_k will be small

Quantization of Filter Coefficients

Even in the case of a two-pole filter section, the structure used to realize the filter section plays an important role in the error caused by coefficient quantization

Example:

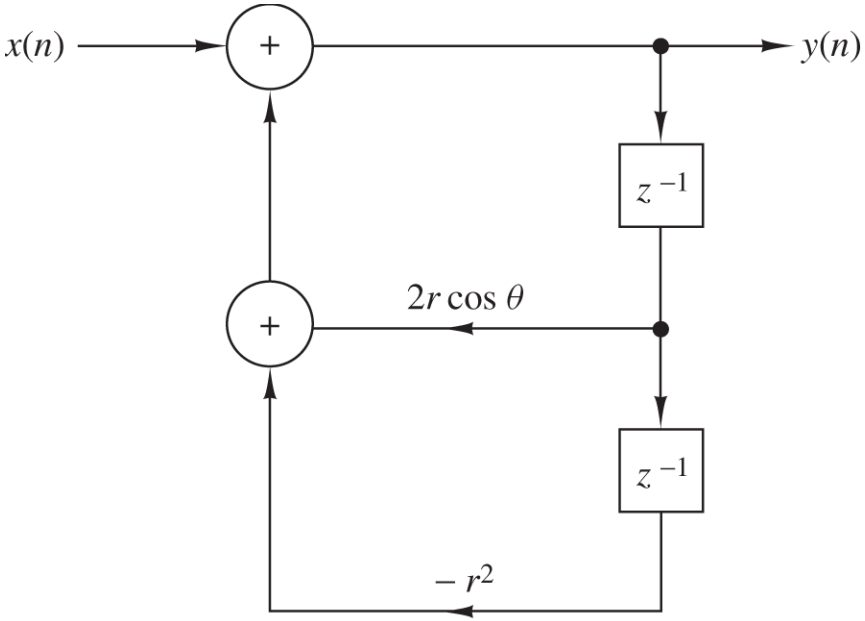
poles at $z = re^{\pm j\theta}$

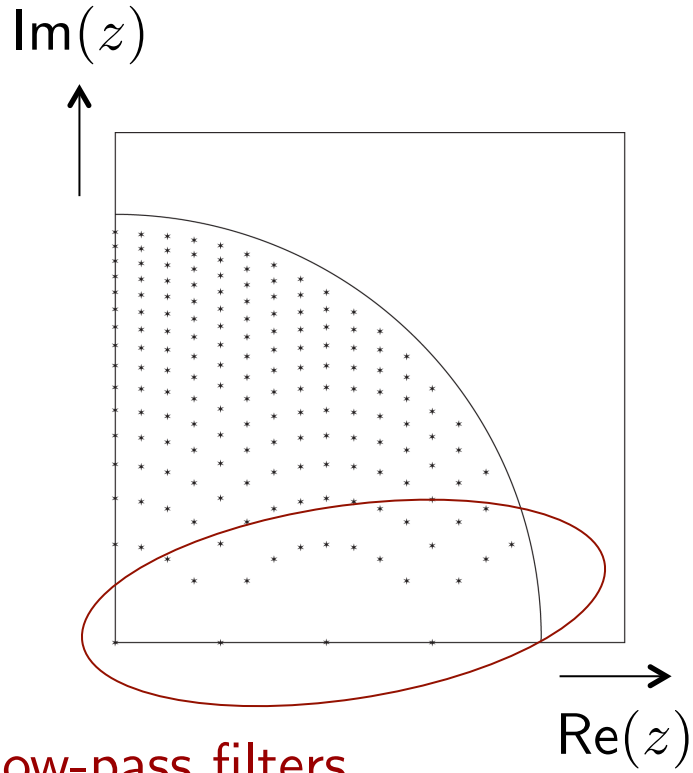
$$H(z) = \frac{1}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}}$$

With finite precision, the possible pole positions are finite as well

Quantization of Filter Coefficients

Example:

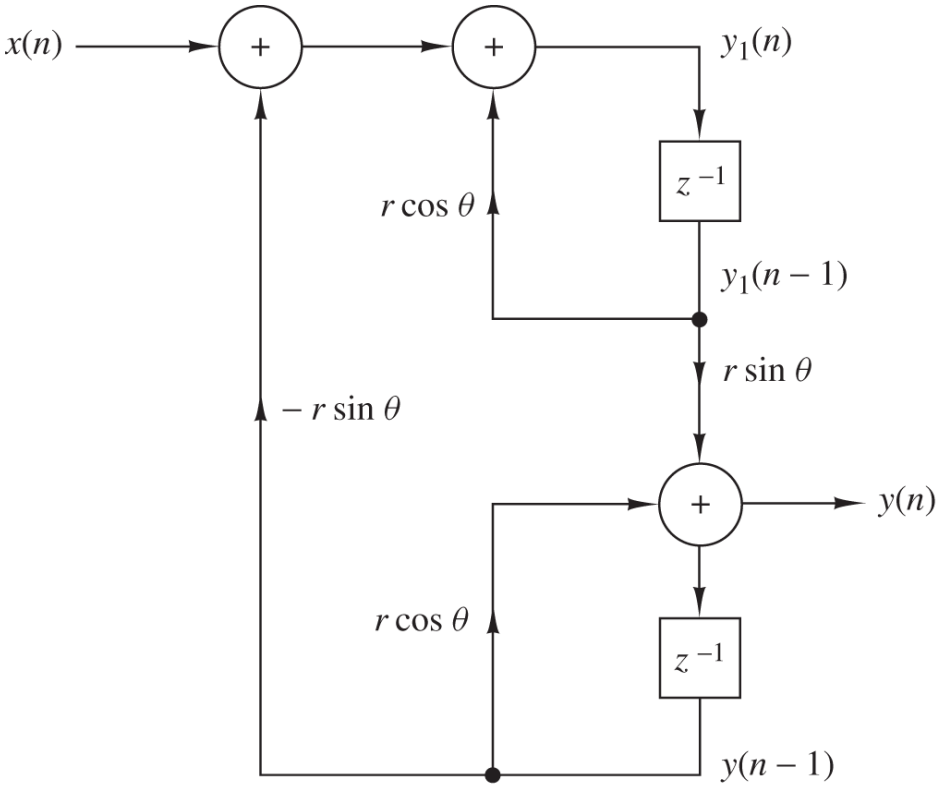


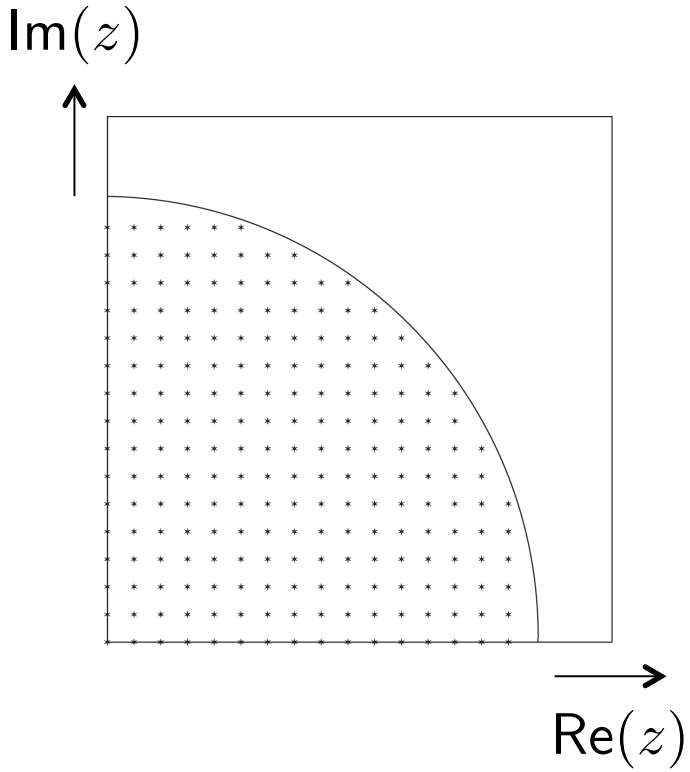


problems with low-pass filters

Quantization of Filter Coefficients

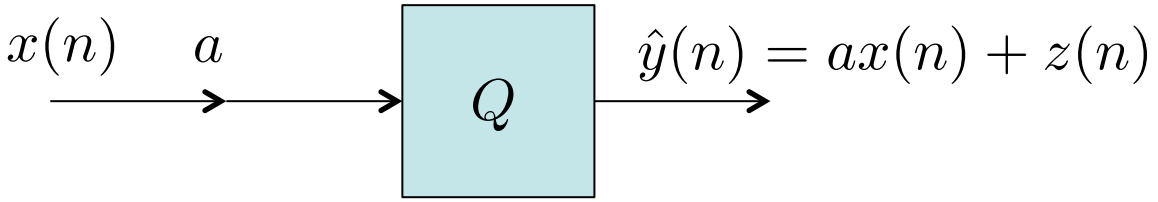
Example:

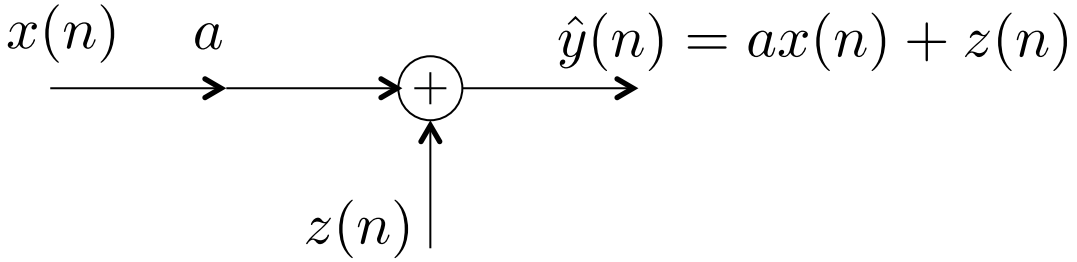




Round-Off Effects in Digital Filters

Round-off errors in multipliers can be modeled as additive white uniformly distributed noise:

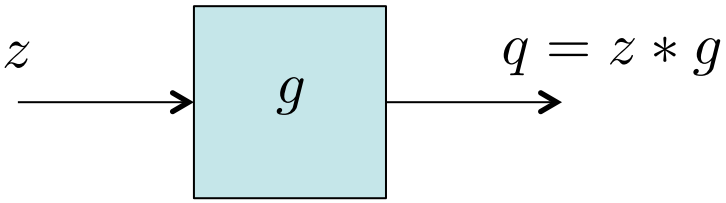




Round-Off Effects in Digital Filters

In order to quantify the effect of quantization noise, we need to know the transfer function of the noise source to the output of our filter.

Let g denote the impulse response of a LTI system:



We then have

$$\sigma_Q^2 = \sigma_Z^2 \sum_{n=-\infty}^{\infty} |g(n)|^2 = \frac{\sigma_Z^2}{2\pi} \int_0^{2\pi} |G(e^{j\omega})|^2 d\omega$$

Round-Off Effects in Digital Filters

Since the filters we consider are LTI, the superposition principle applies. Assuming all p quantization noise sources are statistically independent, we have

$$\sigma_{Q,\text{total}} = \sum_{k=1}^{p} \sigma_{Q_k}^2$$

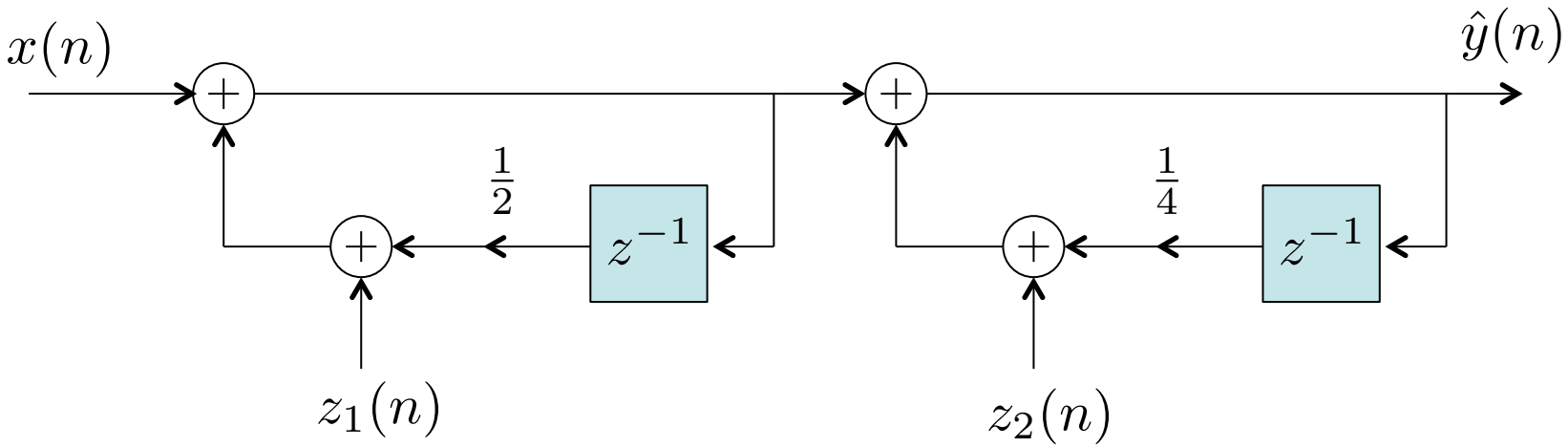
Example:

$$H(z) = \frac{z^2}{(z-\frac{1}{2})(z-\frac{1}{4})} = \frac{2z}{z-\frac{1}{2}} - \frac{z}{z-\frac{1}{4}}$$

and thus

$$h(n) = 2\left(\frac{1}{2}\right)^n u(n) - \left(\frac{1}{4}\right)^n u(n)$$

Round-Off Effects in Digital Filters

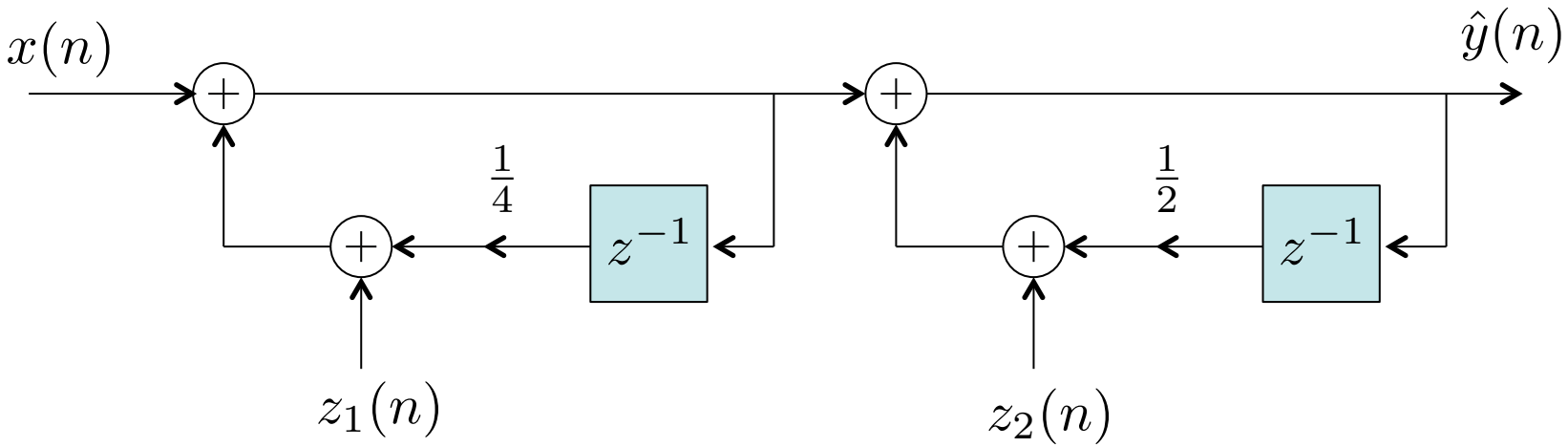


$$\sigma_{Q_1}^2 = \frac{\Delta^2}{12} \sum_{n \geq 0} \left(2 \left(\frac{1}{2} \right)^n - \left(\frac{1}{4} \right)^n \right)^2 \approx 1.83 \frac{\Delta^2}{12}$$

$$\sigma_{Q_2}^2 = \frac{\Delta^2}{12} \sum_{n \geq 0} \left(\frac{1}{4} \right)^{2n} \approx 1.07 \frac{\Delta^2}{12}$$

$$\Rightarrow \sigma_{Q,\text{total}}^2 \approx 2.90 \frac{\Delta^2}{12}$$

Round-Off Effects in Digital Filters



$$\sigma_{Q_1}^2 = \frac{\Delta^2}{12} \sum_{n \geq 0} \left(2 \left(\frac{1}{2} \right)^n - \left(\frac{1}{4} \right)^n \right)^2 \approx 1.83 \frac{\Delta^2}{12}$$

$$\sigma_{Q_2}^2 = \frac{\Delta^2}{12} \sum_{n \geq 0} \left(\frac{1}{2} \right)^{2n} \approx 1.33 \frac{\Delta^2}{12}$$

$\sigma_{Q,\text{total}}^2 = ?$

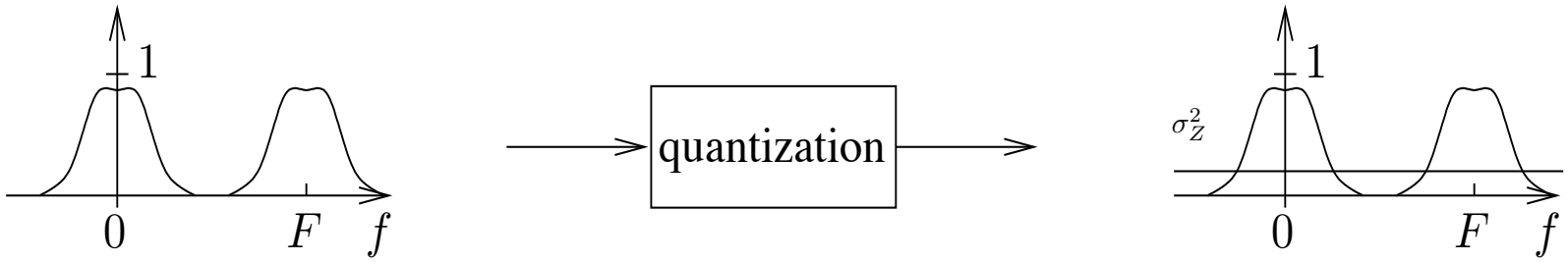
$$\Rightarrow \sigma_{Q,\text{total}}^2 \approx 3.16 \frac{\Delta^2}{12}$$

Oversampled A/D and D/A Convertors

Recall that for a uniform midtread quantizer, we have

$$\sigma_Z^2 = \frac{\Delta}{12}, \quad \text{where} \quad \Delta = \frac{R}{2^{B+1}}$$

- σ_X should match the range R of the quantizer
- $\Rightarrow \sigma_Z^2 \propto \sigma_X$

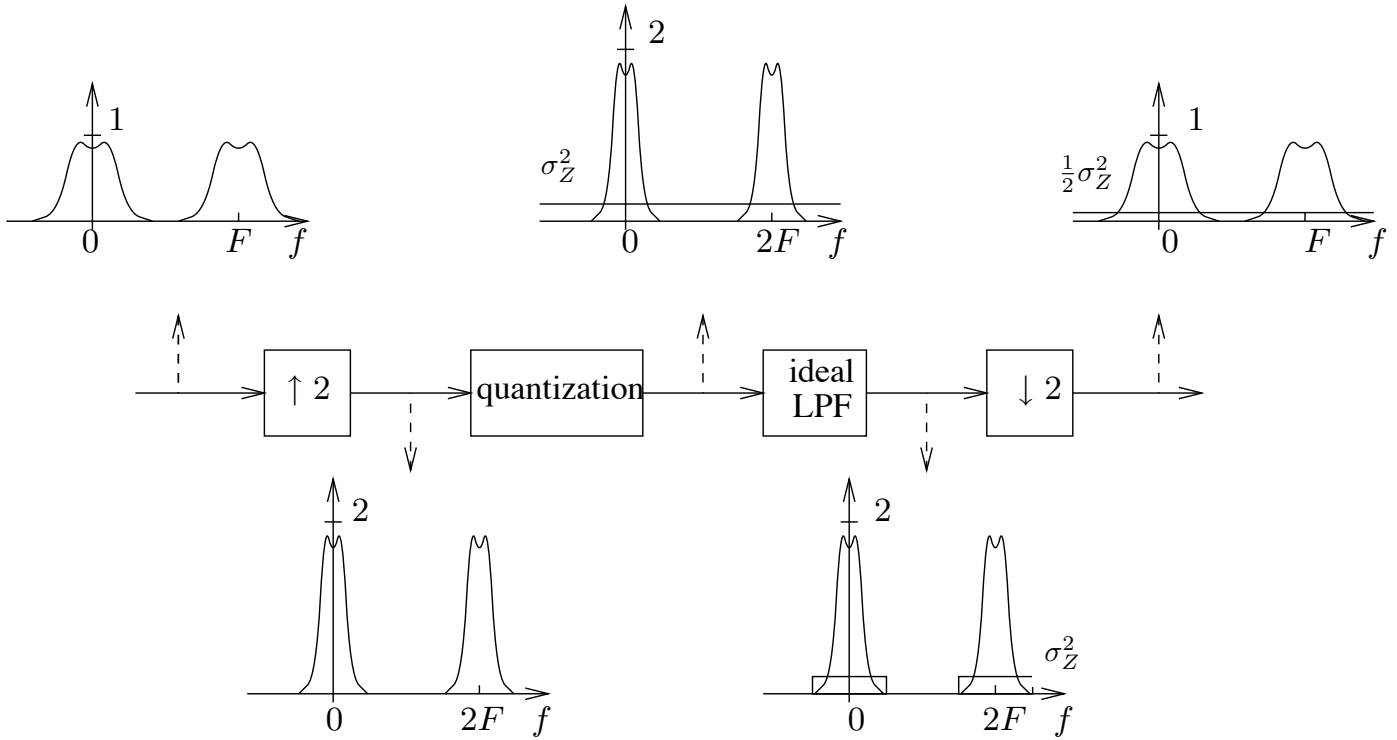


Oversampled A/D and D/A Convertors

- We can improve the SQNR by oversampling (3 dB gain per doubling of f_s)
- We can improve the SQNR by reducing the variance of the input of the quantizer. Or, equivalently, we can reduce the number of bits while keeping the SQNR constant!

Oversampling

We can improve the SQNR by oversampling:



Differential Quantization

Let $d(n) = x(n) - x(n-1)$. The variance of the difference signal is

$$\begin{aligned}\sigma_D^2 &= ED_n^2 = E(X_n - X_{n-1})^2 \\ &= EX_n^2 - 2E(X_n X_{n-1}) + EX_{n-1}^2 \\ &= 2\sigma_X^2(1 - r_X(1)),\end{aligned}$$

where $r_X(1) = R_X(1)/R_X(0) \leq 1$, the normalized autocorrelation coefficient with lag 1

If $r_X(1) > 0.5$, we have that $\sigma_D^2 < \sigma_X^2$ and it is beneficial to quantize the difference signal d instead of x directly (requires less bits!)

Differential Quantization

Even better:

first-order predictor

$$d(n) = x(n) - ax(n-1)$$

We have

$$\sigma_D^2 = (1 + a^2)\sigma_X^2 - 2aR_X(1)$$

Opimal value:

$$a = r_X(1)$$

and

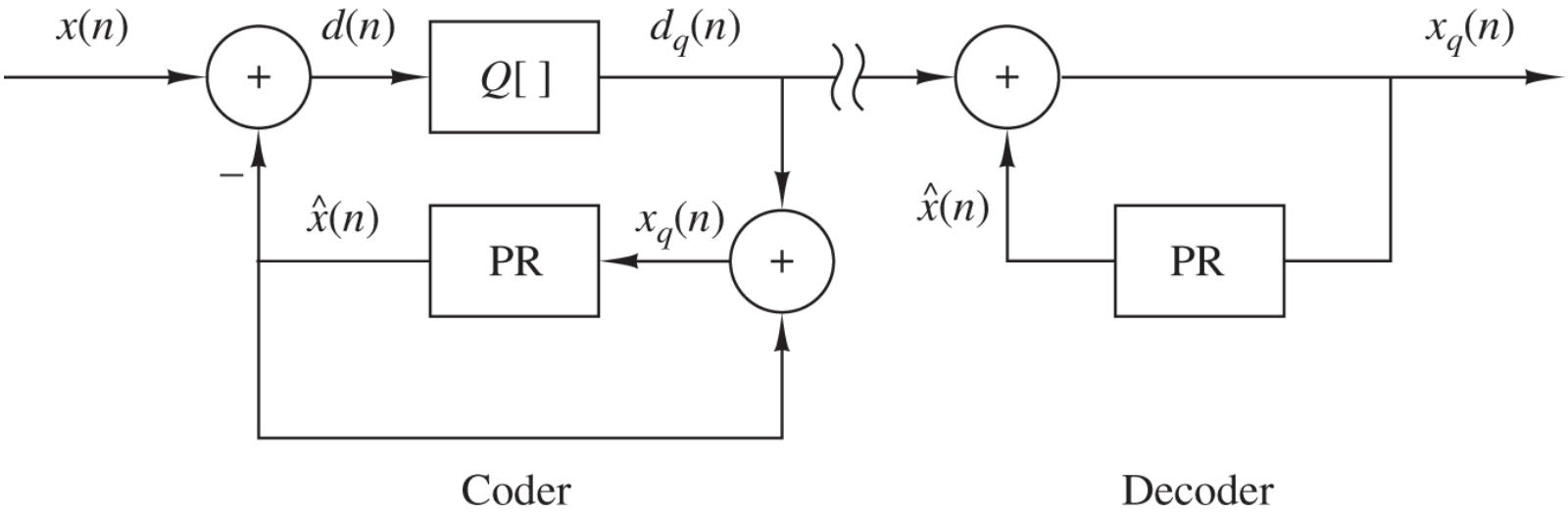
$$\sigma_D^2 = \sigma_X^2(1 - r_X(1)^2) \leq \sigma_X^2$$

$r_X(1) \leq 1$

Differential Quantization

More general: $$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k)$$

DPCM quantizer



$$x_q(n) - x(n) = d_q(n) + \hat{x}(n) - (d(n) + \hat{x}(n)) = d_q(n) - d(n)$$

Differential Quantization

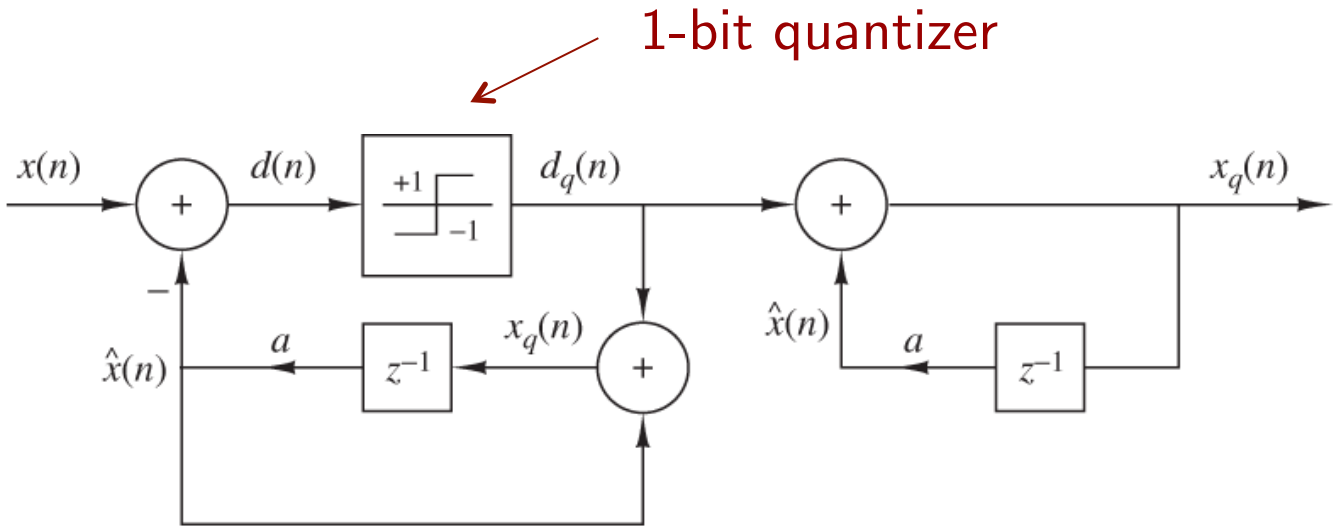
Combining both principles:

- oversampling improves SQNR (3 dB per doubling of f_s)
- SQNR can be further improved by noise shaping (with noise-shape filter of order p)
- oversampling increases the correlation of neighboring samples $\Rightarrow$ reduction of number bits B

Rule of thumb: $\Delta SQNR = 3 + 6p$ dB

Delta Modulation

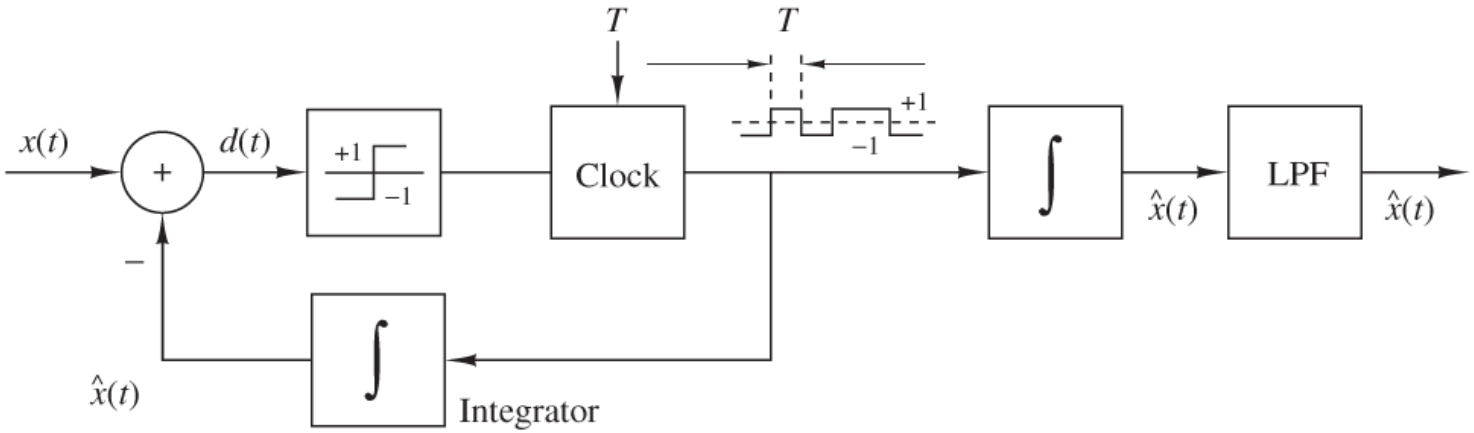
Simplest 1-bit differential predictive quantizer: Delta modulator (DM)



For $a = 1$ we have an ideal accumulator (integrator)

Delta Modulation

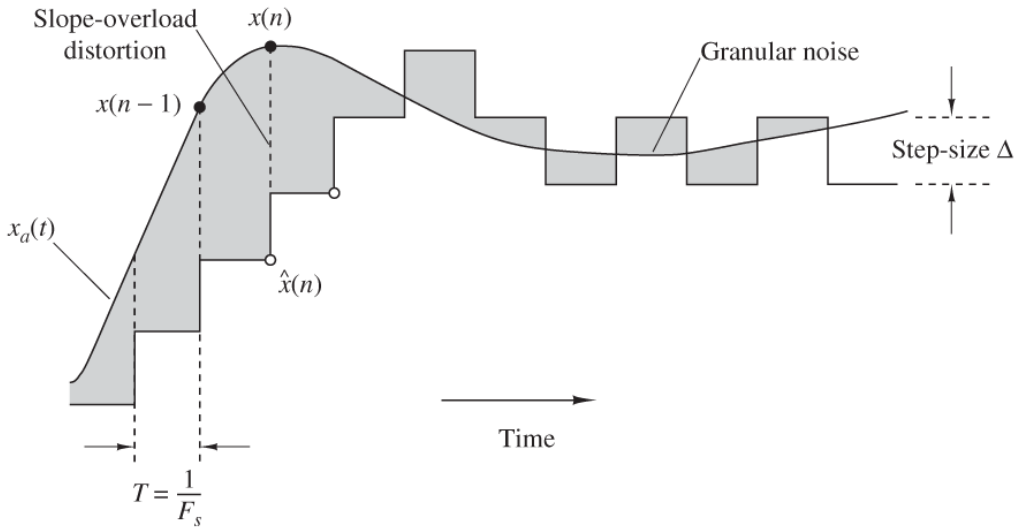
Practical implementation of the delta modulator:



Delta Modulation

Two types of error:

- granular noise
- slope-overload distortion

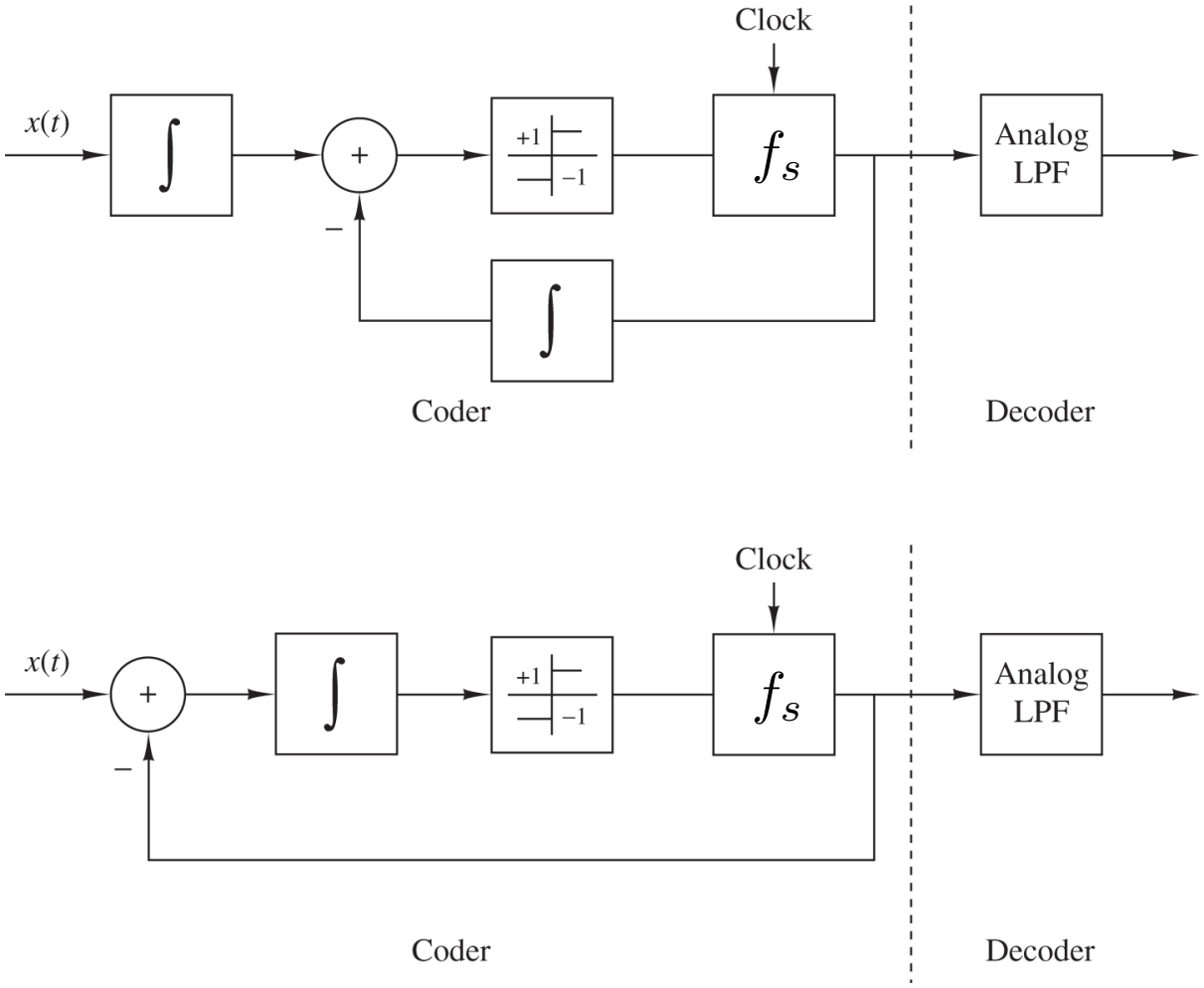


Delta Modulation

Both type of errors can be reduced by using an integrator in front of the DM:

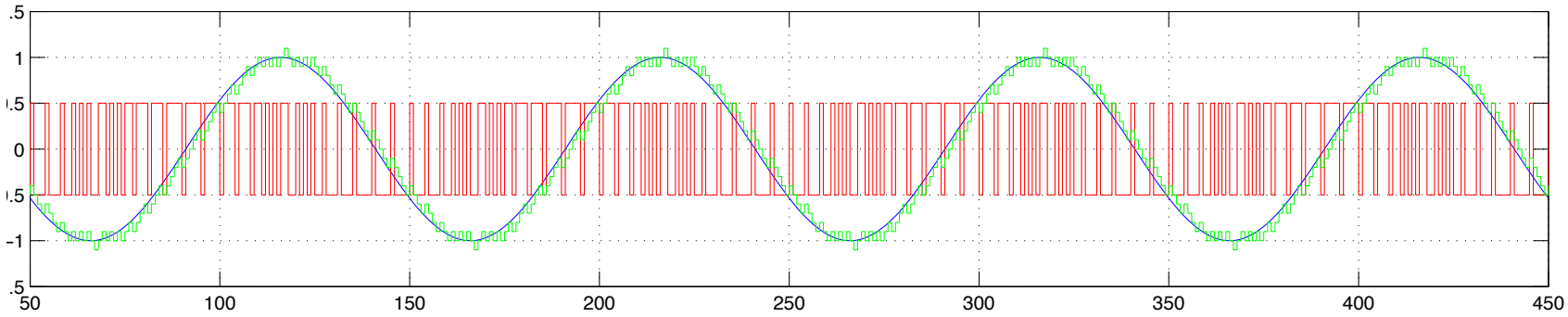
- reduces errors in the low-frequency band (noise shaping)
- increases correlation of quantizer input signal
- simplifies the decoder (just a low-pass filter)

Sigma-Delta Modulation

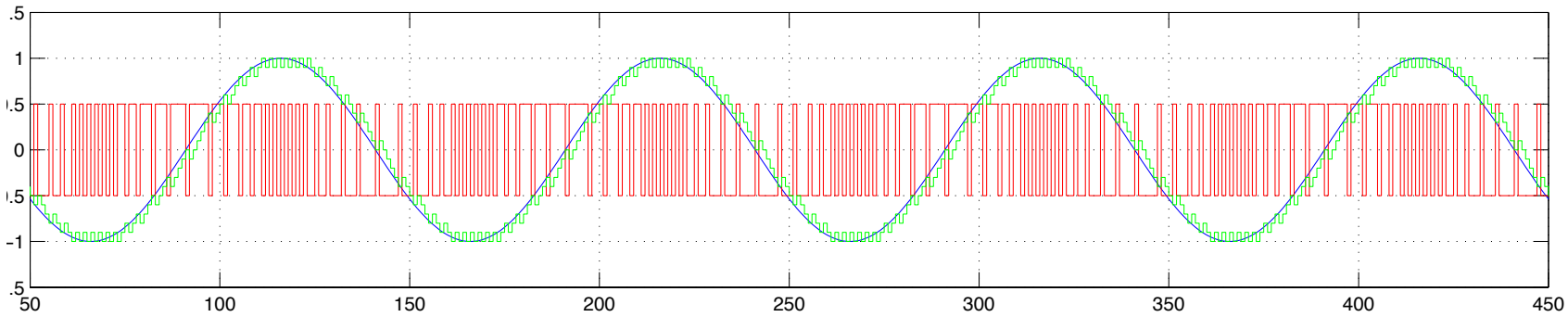


Sigma-Delta Modulation

DM:

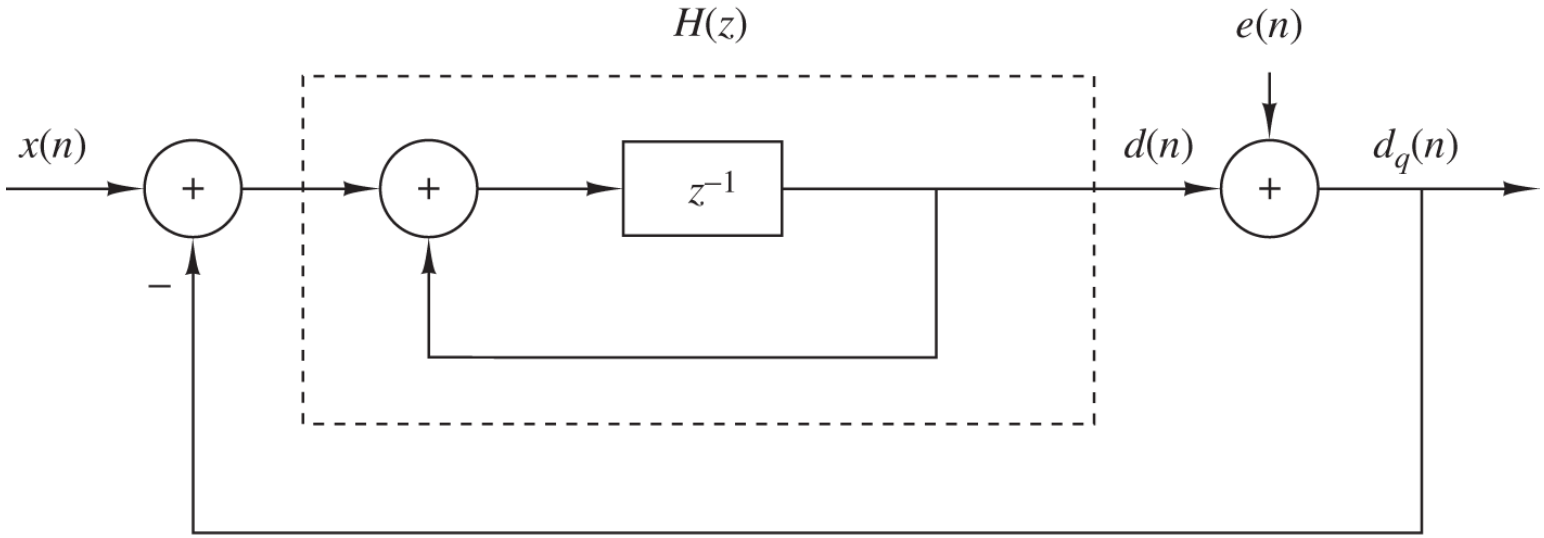


SDM:



Sigma-Delta Modulation

Discrete-time implementation of the SDM:



Sigma-Delta Modulation

Error analysis:

$$D_q(z) = \frac{H_{\text{acc}}(z)}{1 - H_{\text{acc}}(z)} X(z) + \frac{1}{1 - H_{\text{acc}}(z)} E(z)$$

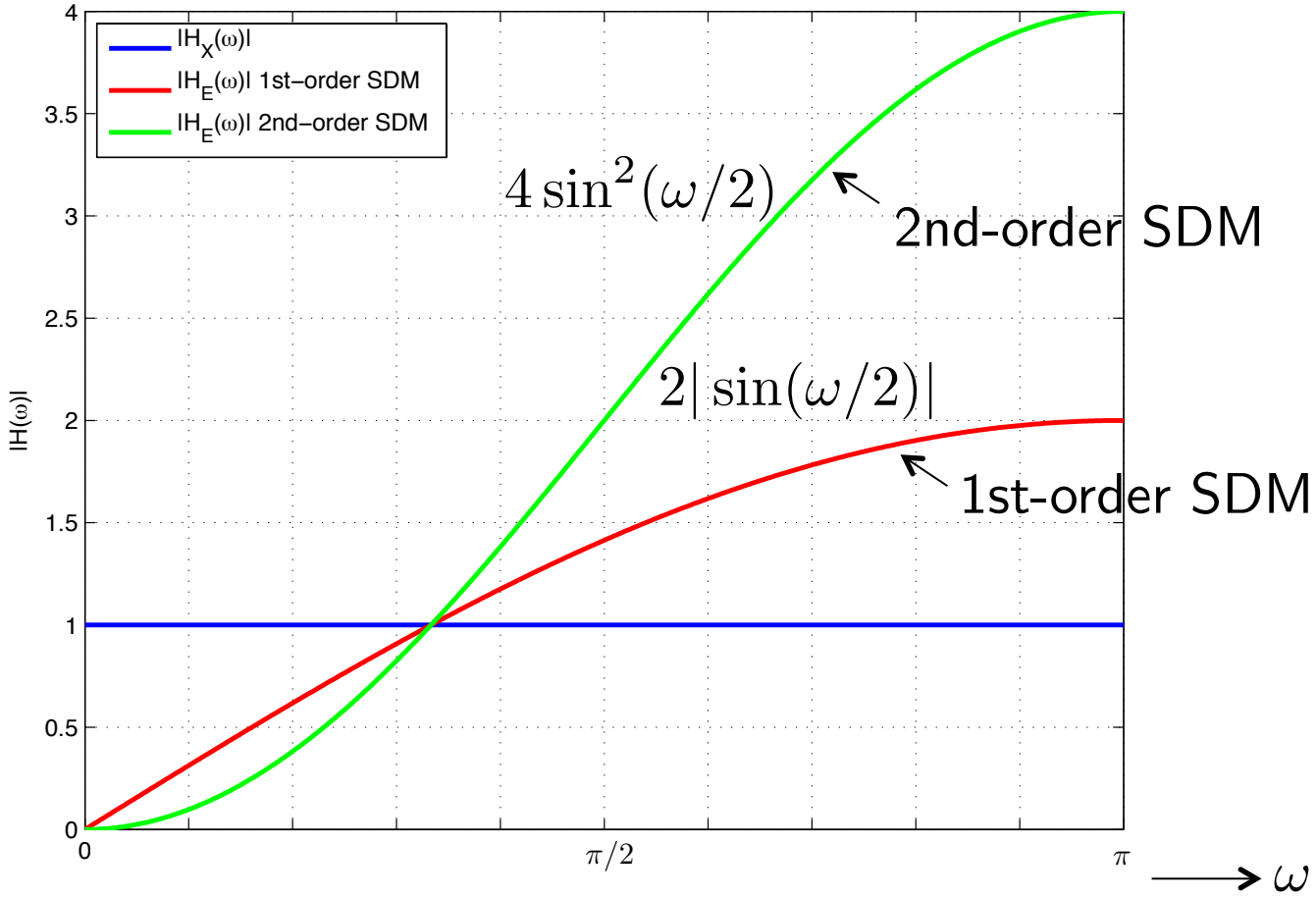
$$= \underbrace{z^{-1}}_{H_X(z)} X(z) + \underbrace{(1 - z^{-1})}_{H_E(z)} E(z)$$

noise shaping!

Hence,

$$|H_X(\omega)| = 1, \qquad |H_E(\omega)| = 2\,|\sin(\omega/2)|$$

Sigma-Delta Modulation



Sigma-Delta Modulation

Example: digital audio coding (20 - 20 kHz, $f_S = 44.1$ kHz)

SQNR PCM signal $\approx 16 \times 6 = 96$ dB

oversample factor γ:

1-bit quantizer

SDM: $\Delta\text{SQNR} == \gamma(3 + 6p) = 96 - 6 = 90$ dB

$p = 1$: $\gamma = 10$ so that $f_s = 2^{10} \times 44.1$ kHz $= 45.2$ MHz

$p = 2$: $\gamma = 6$ so that $f_s = 2^6 \times 44.1$ kHz $= 2.8$ MHz

Sigma-Delta Modulation