Quantization in Digital Filter Structures

Jose Krause Perin

Stanford University

July 30, 2017

Last lecture

- There are different forms of realizing IIR and FIR rational systems
- Their difference becomes evident when considering finite arithmetic precision
- Pipelining and parallel processing solve the problem of using a slow hardware to process a fast signal in two complementary ways.
- Pipelining adds memory (delays) to minimize the critical path. Consequently, pipelining increases latency
- In parallel processing the hardware is replicated to allow processing of multiple input samples simultaneously
- Pipelining and parallel processing can be realized together
- Pipelining and parallel processing are more difficult in IIR systems due to their inherent feedback

Today's lecture

- Fractions and integers representation with two's complement
- Coefficient quantization in FIR systems
- Coefficient quantization in IIR systems
- Roundoff noise in FIR systems
- Roundoff noise in IIR systems

Two's complement

Two's complement is a widely used **fixed-point** representation, whereby fractions are represented by integers.
Any real number x can be represented with infinite precision in two's complement:

$$x = X_m \left(-b_0 + \sum_{i=1}^{\infty} b_i 2^{-i} \right) \qquad \text{(infinite precision)}$$

X_m is a scaling factor, b_0 is the **sign bit**, and the other bits $b_i, i = 1, \ldots, \infty$ are the **magnitude bits**.

Assuming **finite precision** of $(B+1)$ bits:

$$x \approx x_B = X_m \left(-b_0 + \sum_{i=1}^{B} b_i 2^{-i} \right) \qquad (B+1 \text{ bits precision})$$

$B+1$ is referred to as the **word length**.

Two's complement

Examples: $(1+5)$-bits and $X_m = 1$.

b_0	b_1	b_2	b_3	b_4	b_5
0	1	0	0	1	0

$= 0.5625$

b_0	b_1	b_2	b_3	b_4	b_5
0	1	0	0	1	1

$= 0.5938$

If $X_m = 2^B$. We have integer representation:

$$2^B \left(-b_0 + \sum_{i=1}^{B} b_i 2^{-i} \right) = -b_0 2^B + \sum_{i=1}^{B} b_i 2^{B-i}$$

$$0.5625 \times 2^5 = 18$$
$$0.5938 \times 2^5 = 19$$

Q notation

The **Q notation** is a convenient way of keeping track of the **binary point** ($\Diamond$).
Suppose the word length is 16 bits, then we can represent integers from $-32768 \leq A \leq 32767$.
If we want to represent a fraction a such that $-1 \leq a \leq 1$:

$$a = A \times 2^{-15} \Longleftrightarrow A = a \times 2^{15}$$

Example: $a = 0.75 \Longleftrightarrow A = 24576_{10}$ in Q15

$$0_{\Diamond} \underbrace{110000000000000}_{\text{15 bits}}$$

Q15 means 15 bits are used to represent the fraction, and only one bit is used to represent the integer (sign bit).

Another example:
Now assume that $-4 \leq a \leq 4$.
We can represent a by a Q13 integer A such that $-32768 \leq A \leq 32767$ (16 bits)

$$a = A \times 2^{-13} \iff A = a \times 2^{13}$$

Suppose $a = 3.5 \iff A = 28672_{10}$ Q13

$$011_{\diamondsuit} \underbrace{1000000000000}_{\text{13 bits}}$$

In this case, 13 bits are used to represent the fraction (0.5), whereas 3 bits are used to represent the integer part (3)

Q0 means integers

$$0111000000000000_{\diamondsuit} = 28672 \text{ Q0}$$

Roundoff error

The difference between the actual number x and its representation x_B is known as **roundoff error** (or noise):

$$e = x - x_B$$

Similarly to quantization error, the roundoff error is bounded in an interval of length Δ:

$$\Delta = \frac{\text{range}}{\text{number of steps}} = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}$$

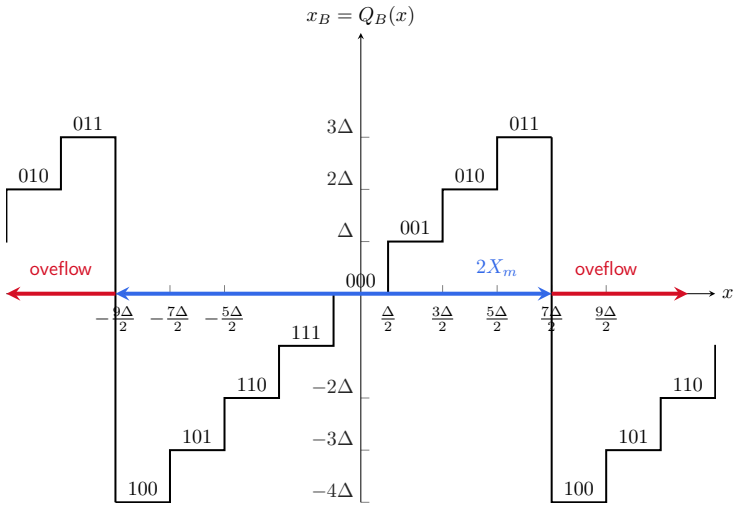
For instance, for $X_m = 1$ and Q15, we would have

$$\Delta = \frac{1}{2^{15}} = 0.0000305176$$

Overflow vs clipping

There are two possibilities when a number exceeds the representation range

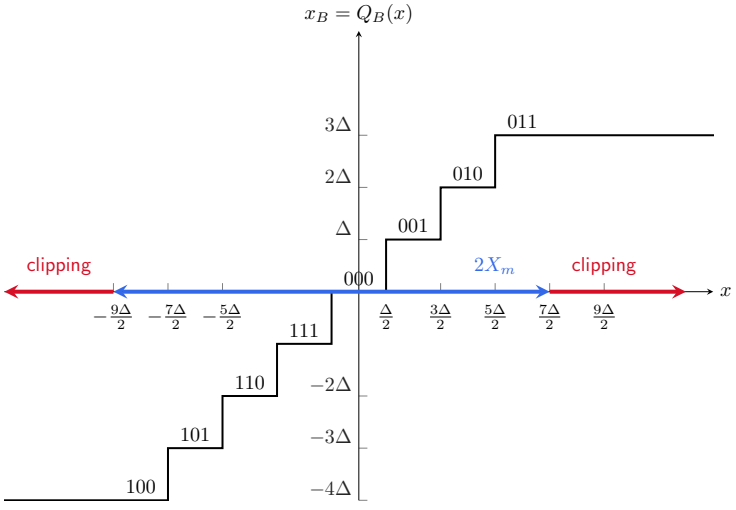
1. **Overflow** wraps around.



Overflow vs clipping

There are two possibilities when a number exceeds the representation range

2. **Clipping** holds the output at the highest level.



Overflow vs clipping

Additional comments:

- There's an inherent trade-off between quantization error and overflow/clipping.
- We'd like to make the signal small in order to make overflow/clipping rare, but making the signal small leads to excessive quantization error, much like losing bits of resolution in quantization
- If the signal is too large overflow/clipping will be frequent, resulting in frequent arithmetic errors.

Arithmetic operations in two's complement

Sign change

Complement all bits and add 1 to the least significant bit

$$-6 = -(0110) = 1001 + 1 = 1010$$

Accumulation or addition

Regular addition in binary.

When adding 3 or more two's complement numbers, the intermediate sum can overflow, but the final sum will be correct if it does not exceed the word length of the numbers.

$$6 + 4 + (-6) = \underbrace{10}_{\text{overflow}} - (6) = 4$$

$$0110 + 0100 + (1010) = \underbrace{1010}_{\text{overflow}} + 1010 = 0100$$

The final sum is correct, despite the overflow when calculating $6 + 4$.

Arithmetic operations in two's complement

Addition in different scales

When adding scaled binary numbers, you must line up the binary points. This can be done by shifting one or the other of the numbers either left or right (multiply by power of 2).

Example:

$$\begin{aligned} 00_{\diamond}01000000000000000000000000000000 &= 0.25 \quad \text{Q30} \\ +0001_{\diamond}00100000000000000000000000000000 &= 1.125 \quad \text{Q28} \end{aligned}$$

$$\begin{aligned} 0000_{\diamond}01000000000000000000000000000000 &= 0.25 \quad \text{Q30} \\ +0001_{\diamond}00100000000000000000000000000000 &= 1.125 \quad \text{Q28} \\ \hline 0001_{\diamond}01100000000000000000000000000000 &= 1.375 \quad \text{Q28} \end{aligned}$$

Arithmetic operations in two's complement

Multiplication

$$\underbrace{Y}_{2B+1\text{bits}} = \underbrace{A}_{B+1\text{bits}} \times \underbrace{X}_{B+1\text{bits}}$$

Generally the hardware produces Y with 2B+2 bits (two sign bits)
Examples:

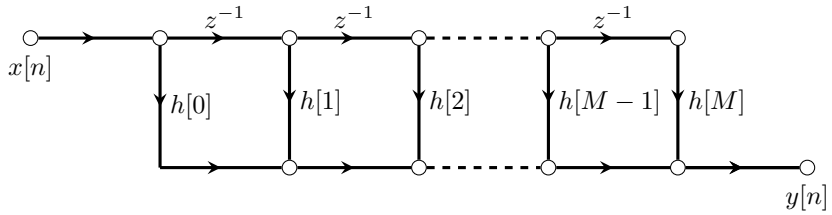
$$\begin{array}{r} 0_{\diamond}110000000000000 = 0.75 \quad \text{Q15} \\ \times 0_{\diamond}100000000000000 = 0.5 \quad \text{Q15} \\ \hline 00_{\diamond}011000000000000000000000000000 = 0.375 \quad \text{Q30} \end{array}$$

$$\begin{array}{r} 01_{\diamond}10000000000000 = 1.5 \quad \text{Q14} \\ \times 00_{\diamond}11000000000000 = 0.5 \quad \text{Q14} \\ \hline 0001_{\diamond}001000000000000000000000000 = 1.125 \quad \text{Q28} \end{array}$$

Because of the extra bits, **multiplications do not overflow**.

Quantization in systems implementation

Example of FIR system (same applies to IIR):



Practical issues:

- Coefficients $\{h[n]\}$ must be quantized to fit the representation.
- Multiplications and additions are realized with finite precision.
- Multiplications do not overflow.
- Additions may overflow. Must scale input signal or coefficients to prevent overflow.
- Decreasing the representation leads to **roundoff errors** e.g., going from $2B+1$ bits to $B+1$ bits after multiplication. Roundoff noise will be treated with the linear noise model, similarly to quantization.

Coefficient quantization

Given a system

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 - a_1 z^{-1} - \ldots - a_N z^{-N}}.$$

The coefficients $\{a_1, \ldots, a_M, b_0, \ldots b_N\}$ obtained in design must be quantized to $(B+1)$ bits for implementation:

$$\hat{b}_k = Q_B\{b_k\} = b_k + \Delta b_k, \qquad k = 0, \ldots, M$$
$$\hat{a}_k = Q_B\{a_k\} = a_k + \Delta a_k, \qquad k = 1, \ldots, N$$

Quantizing to (B+1) bits:

$$\hat{b}_k = Q_B\{b_k\} = \begin{cases} 2^{-B}\text{round}(b_k \times 2^B), & \textbf{rounding} \\ 2^{-B}\lfloor b_k \times 2^B \rfloor, & \textbf{truncating} \end{cases}$$

In Matlab:

$$\hat{b}_k = Q_B\{b_k\} = \begin{cases} \texttt{2\^{}(-B)*round(b*2\^{}B)}, & \textbf{rounding} \\ \texttt{2\^{}(-B)*floor(b*2\^{}B)}, & \textbf{truncating} \end{cases}$$

Coefficient quantization

Question: what happens to poles, zeros, and the frequency response after coefficient quantization?

For FIR systems the main concerns are

- Preserving linear phase (if linear phase)
- Error in magnitude and phase responses

For IIR systems the main concerns are

- Stability
- Error in magnitude and phase responses

Example: coefficient quantization of linear-phase FIR filter

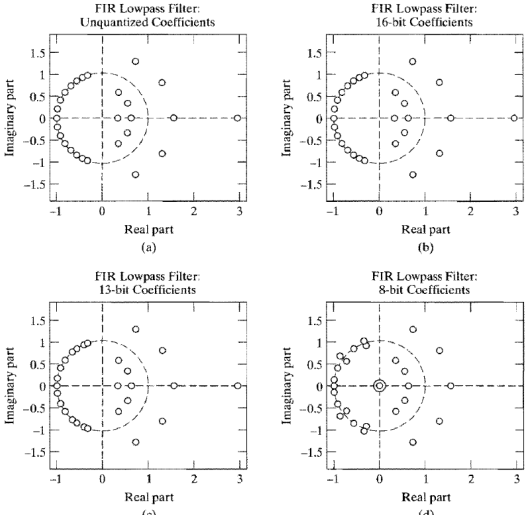
TABLE 6.5 UNQUANTIZED AND QUANTIZED COEFFICIENTS FOR AN OPTIMUM FIR LOWPASS FILTER ($M = 27$)

Coefficient	Unquantized	16 bits	14 bits	13 bits	8 bits
$h[0] = h[27]$	1.359657×10^{-3}	45×2^{-15}	11×2^{-13}	6×2^{-12}	0×2^{-7}
$h[1] = h[26]$	-1.616993×10^{-3}	-53×2^{-15}	-13×2^{-13}	-7×2^{-12}	0×2^{-7}
$h[2] = h[25]$	-7.738032×10^{-3}	-254×2^{-15}	-63×2^{-13}	-32×2^{-12}	-1×2^{-7}
$h[3] = h[24]$	-2.686841×10^{-3}	-88×2^{-15}	-22×2^{-13}	-11×2^{-12}	0×2^{-7}
$h[4] = h[23]$	1.255246×10^{-2}	411×2^{-15}	103×2^{-13}	51×2^{-12}	2×2^{-7}
$h[5] = h[22]$	6.591530×10^{-3}	216×2^{-15}	54×2^{-13}	27×2^{-12}	1×2^{-7}
$h[6] = h[21]$	-2.217952×10^{-2}	-727×2^{-15}	-182×2^{-13}	-91×2^{-12}	-3×2^{-7}
$h[7] = h[20]$	-1.524663×10^{-2}	-500×2^{-15}	-125×2^{-13}	-62×2^{-12}	-2×2^{-7}
$h[8] = h[19]$	3.720668×10^{-2}	1219×2^{-15}	305×2^{-13}	152×2^{-12}	5×2^{-7}
$h[9] = h[18]$	3.233332×10^{-2}	1059×2^{-15}	265×2^{-13}	132×2^{-12}	4×2^{-7}
$h[10] = h[17]$	-6.537057×10^{-2}	-2142×2^{-15}	-536×2^{-13}	-268×2^{-12}	-8×2^{-7}
$h[11] = h[16]$	-7.528754×10^{-2}	-2467×2^{-15}	-617×2^{-13}	-308×2^{-12}	-10×2^{-7}
$h[12] = h[15]$	1.560970×10^{-1}	5115×2^{-15}	1279×2^{-13}	639×2^{-12}	20×2^{-7}
$h[13] = h[14]$	4.394094×10^{-1}	14399×2^{-15}	3600×2^{-13}	1800×2^{-12}	56×2^{-7}

Question: will the linear phase property of this FIR filter be preserved after coefficient quantization?

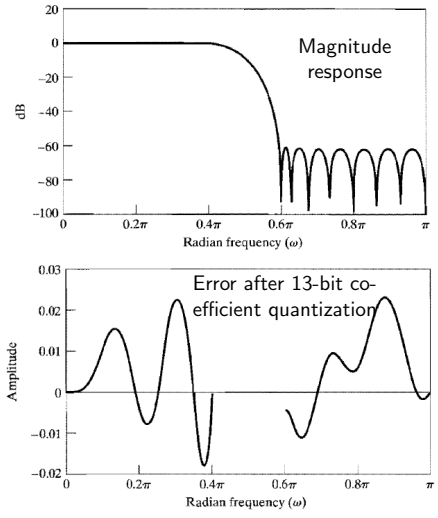
Example: coefficient quantization of linear-phase FIR filter

Zeros move around significantly, but they remain in **conjugate** and **conjugate reciprocal** pairs ($\{c, c^*, 1/c, 1/c^*\}$) since symmetry of impulse response is preserved after quantization



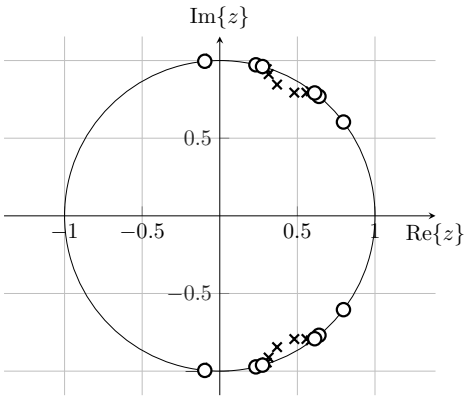
Example: coefficient quantization of linear-phase FIR filter

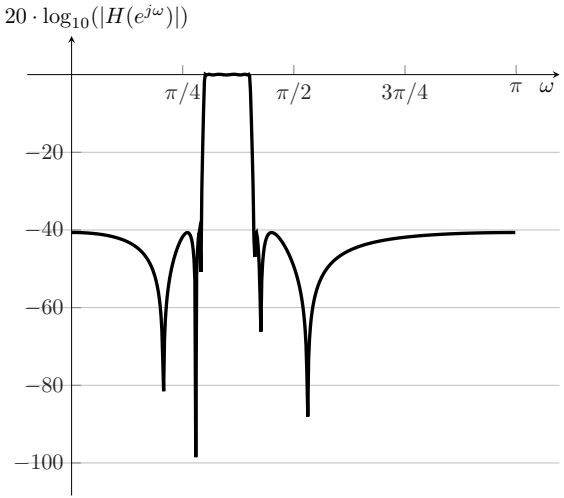
Error in the magnitude response for 13-bit quantization. Error for transition band is not shown.



Example: coefficient quantization in IIR filter

Bandpass 12th-order elliptic filter





Example: coefficient quantization in IIR filter

Quantization with 16 bits makes the system **unstable**. Some of the poles fall outside the unit circle.

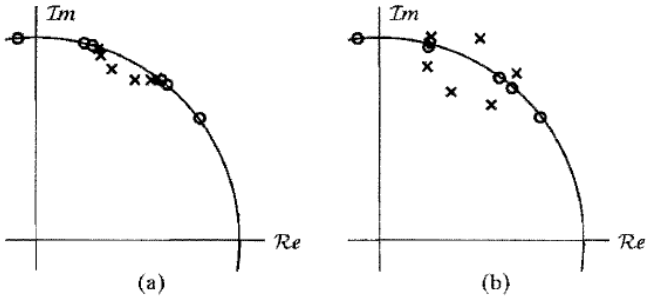


Figure 6.48 IIR coefficient quantization example. (a) Poles and zeros of $H(z)$ for unquantized coefficients. (b) Poles and zeros for 16-bit quantization of the direct form coefficients.

In general, tightly clustered roots are very sensitive to coefficient quantization.
How to mitigate this problem?

- Higher resolution
- Cascade or parallel forms are less sensitive to coefficient quantization than direct forms.

Summary on coefficient quantization

FIR

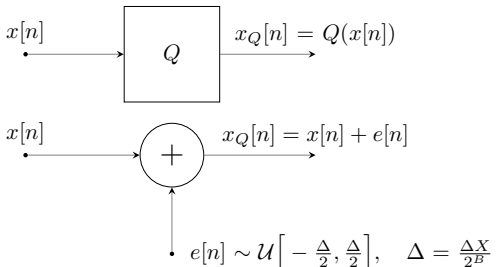
- FIR systems will remain stable after coefficients quantization since all poles are at $z = 0$.
- We can preserve the linear phase property by keeping the impulse response symmetric: $h[n] = \pm h[M - n]$. This is typically assured in the implementation by using the special structure for linear phase FIR systems (lecture 7)
- Magnitude response is affected, but difference is typically not significant.

IIR

- IIR systems can become unstable after coefficients quantization
- These issues are mitigated by increasing word length, or using cascade or parallel forms.

Roundoff noise and the linear noise model

As for quantization, roundoff is modeled as an additive **white noise uniformly distributed** that is independent of the input signal.



Assuming $X_m = 1$ (typical), we have the following values:

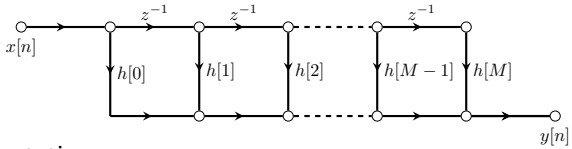
$$\sigma_B^2 = \frac{\Delta^2}{12} = \frac{2^{-2B}}{12} \qquad \text{(average power)}$$

$$\phi_{ee}[m] = \sigma_B^2 \delta[m] \qquad \text{(autocorrelation function)}$$

$$\Phi_{ee}(e^{j\omega}) = \sigma_B^2 \qquad \text{(PSD)}$$

Roundoff noise in FIR systems

Direct form of FIR filter



Two forms of implementation:

1. Quantization immediately after each multiplication:

$$H(z) = \sum_{k=0}^{M} Q_B\{h[k]z^{-k}\}$$

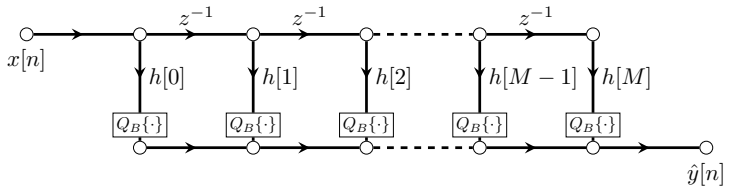
   The multiplication produces $2B+1$ bits, but the B LSBs are discarded.

2. Quantization immediately after accumulation:

$$H(z) = Q_B\left\{\sum_{k=0}^{M} h[k]z^{-k}\right\}$$

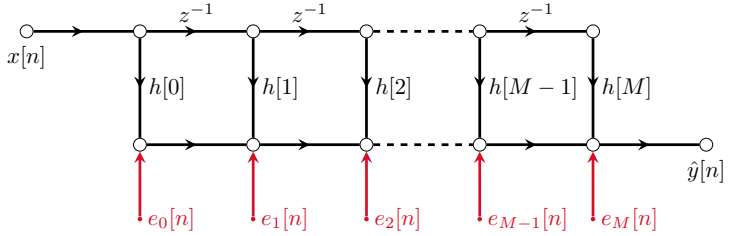
   Requires $(2B+1)$-bit accumulators (adders). Multiply and accumulate (MAC) instruction in many architectures.

1. Quantization immediately after multiplication

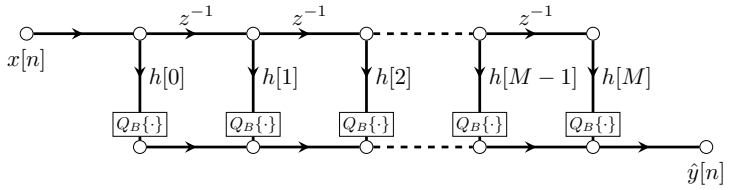


Equivalent **linear noise model**:

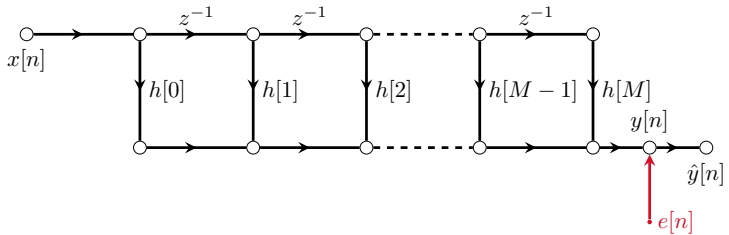
Every noise source has average power: $\sigma_B^2 = \frac{\Delta^2}{12}$



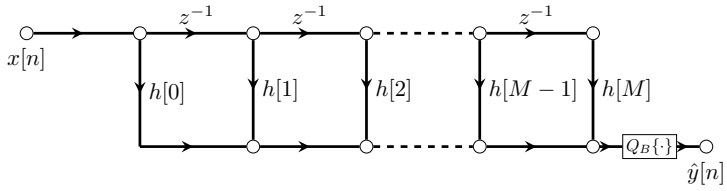
1. Quantization immediately after multiplication



Equivalent **linear noise model**:
Lump noise sources into one with average power: $\mathbb{E}(e^2[n]) = (M+1)\sigma_B^2$

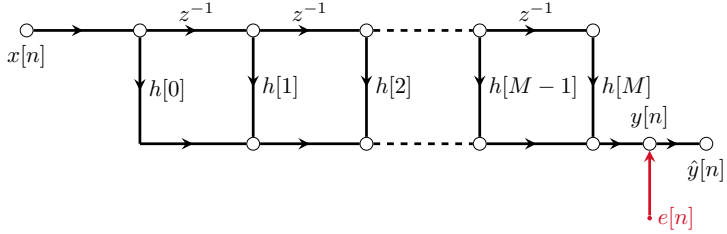


2. Quantization immediately after accumulation



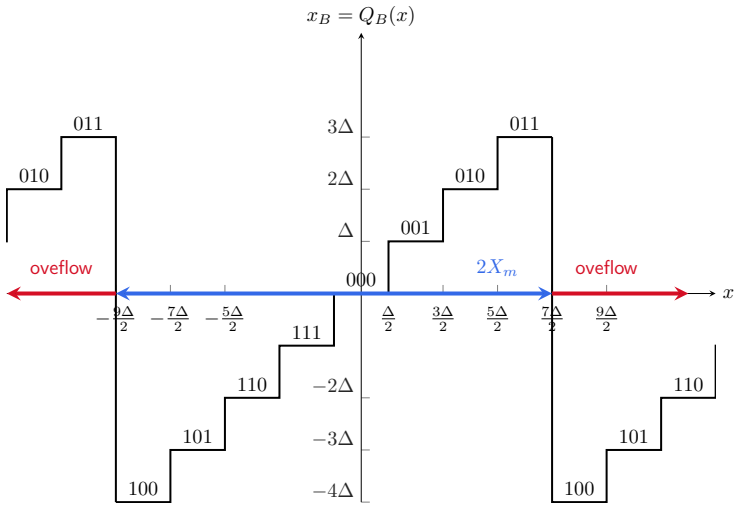
Equivalent **linear noise model**:

Output noise source has average power $\mathbb{E}(e^2[n]) = \frac{\Delta^2}{12}$. The noise power is $(M+1)$ times smaller than in quantization after multiplication



Preventing overflow in FIR systems

Larger range X_m reduces chance of overflow, but leads to higher quantization error.



Preventing overflow in FIR systems

Assuming that the input signal is bounded between $-1 \leq x[n] \leq 1$. Overflow will not happen, if $|y[n]| < 1$ for any input $x[n]$.
What are the conditions in the coefficients $h[n]$ to avoid overflow?

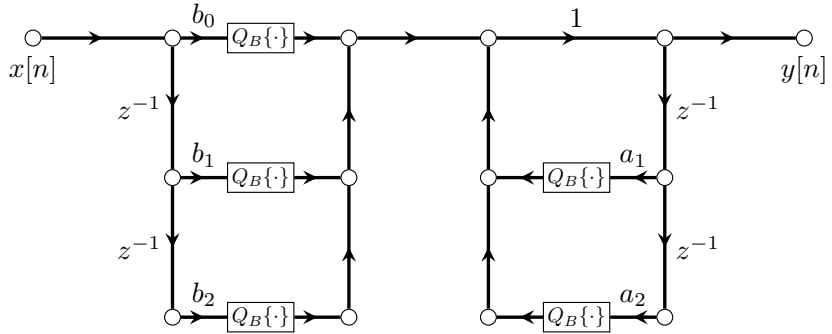
$$
\begin{aligned}
|y[n]| &= \left| \sum_{k=0}^{M} h[k]x[n-k] \right| && \text{(modulus of convolution sum)} \\
&\leq \sum_{k=0}^{M} |h[k]| \cdot |x[n-k]| && \text{(Schwarz inequality)} \\
&\leq \sum_{k=0}^{M} |h[k]| && \text{(since } -1 \leq x[n] \leq 1\text{)} \\
&< 1 \implies \text{no overflow}
\end{aligned}
$$

Roundoff noise in IIR systems

- We'll treat roundoff noise in IIR systems similarly to in FIR systems
- The inherent feedback of IIR systems will lead to roundoff noise **shaping**
- Now the difference between the different structures will become apparent
  - Direct form I & II
  - Transposed forms
  - Cascade forms
  - Parallel forms

Direct form I IIR filter

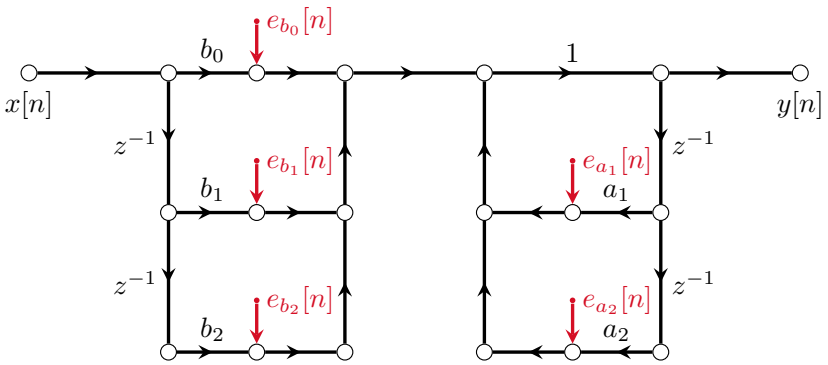
Quantization is performed after every multiplication



Direct form I IIR filter

As for FIR filters, we can replace the quantizers by noise sources with average power

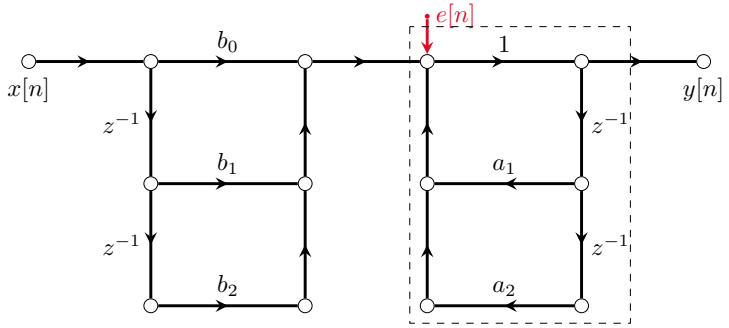
$$\sigma_B^2 = \frac{\Delta^2}{12} = \frac{2^{-2B}}{12}$$



Direct form I IIR filter

Combining all noise sources into one

$$\sigma_e^2 = (M+1+N)\sigma_B^2 = (M+1+N)\frac{2^{-2B}}{12}$$



Quantization noise is **shaped** by the filter defined in the dashed rectangle:

$$H_a(z) = \frac{1}{A(z)} = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}}$$ (shaped by the poles)

Direct form I IIR filter

Calculating the roundoff noise PSD at the output:

$$\begin{aligned}\Phi_{\tilde{e}\tilde{e}}(e^{j\omega}) &= |H_a(e^{j\omega})|^2 \Phi_{ee}(e^{j\omega}) \\ &= \frac{1}{|A(e^{j\omega})|^2}(M+N+1)\sigma_B^2\end{aligned}$$

The average power is obtained by integrating the PSD over $[-\pi, \pi]$:

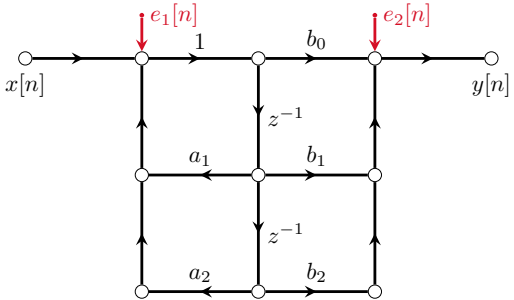
$$\begin{aligned}\sigma_{\tilde{e}}^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{\tilde{e}\tilde{e}}(e^{j\omega})d\omega \\ &= (M+N+1)\frac{\sigma_B^2}{2\pi}\int_{-\pi}^{\pi}\frac{1}{|A(e^{j\omega})|^2}d\omega\end{aligned}$$

The actual value will depend on $A(e^{j\omega})$, but the integral could be larger than 1. That is, $A(e^{j\omega})$ could enhance the roundoff noise.

Direct form II IIR filter

Similarly to the direct form I, quantization may be performed after multiplications. The quantizers are replaced by noise sources of average power σ_B^2. Noise sources are combined whenever possible

$$\sigma_{e_1}^2 = N\sigma_B^2 \qquad \text{(average power of } e_1[n]\text{)}$$
$$\sigma_{e_2}^2 = (M+1)\sigma_B^2 \qquad \text{(average power of } e_2[n]\text{)}$$



Noise source $e_2[n]$ is already at the output, but $e_1[n]$ will be shaped by the filter $H(z)$:

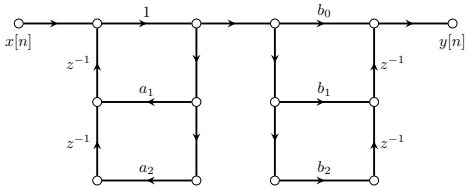
$$\begin{aligned}\Phi_{\tilde{e}\tilde{e}}(e^{j\omega}) &= |H(e^{j\omega})|^2\Phi_{e_1e_1}(e^{j\omega}) + \Phi_{e_2e_2}(e^{j\omega}) \\ &= N\sigma_B^2|H(e^{j\omega})|^2 + (M+1)\sigma_B^2\end{aligned}$$

The average quantization noise power at the output is given by

$$\begin{aligned}\sigma_{\tilde{e}}^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_{\tilde{e}\tilde{e}}(e^{j\omega})d\omega \\ &= N\sigma_B^2\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\omega})|^2d\omega + (M+1)\sigma_B^2 \\ &= N\sigma_B^2\sum_{n=-\infty}^{\infty}|h[n]|^2 + (M+1)\sigma_B^2\end{aligned}$$

What about transposed forms?

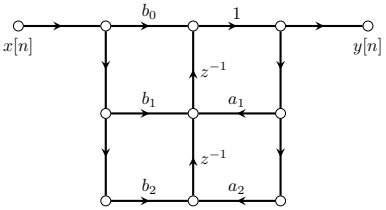
Transposed direct form I



Average power of each noise source

PSD at the output:

Transposed direct form II



Average power of each noise source

PSD at the output:

Summary roundoff noise in IIR systems

PSD of roundoff noise at the output

	I	II
Direct	$(M+N+1)\sigma_B^2 \dfrac{1}{\lvert A(e^{j\omega})\rvert^2}$	$N\sigma_B^2 \lvert H(e^{j\omega})\rvert^2 + (M+1)\sigma_B^2$
Transposed	$N\sigma_B^2 \lvert H(e^{j\omega})\rvert^2 + (M+1)\sigma_B^2$	$(M+N+1)\sigma_B^2 \dfrac{1}{\lvert A(e^{j\omega})\rvert^2}$

$M+1$ coefficients $\{b_0, b_1, \ldots, b_M\}$, assumed different from zero and one ($b_i \neq 0, b_i \neq 1$).
N coefficients $\{a_1, \ldots, a_N\}$, assumed different from zero and one ($a_i \neq 0, a_i \neq 1$).
$\sigma_B^2 = 2^{-2B}/12$ for a $(B+1)$-bit two's complement representation.

Conclusions:

- The least *noisy* implementation depends on $H(e^{j\omega})$ and on $A(e^{j\omega})$.
- Use parallel and cascade forms to group 2nd-order factors (good grouping is important) and realize each subsystem with the most convenient structure.
- In addition to grouping, in cascade forms attention also has to be paid to the ordering of the subsystems.

Summary

- Two's complement is a fixed-point representation that represents fractions as integers
- There's an inherent trade-off between roundoff noise and overflow/clipping
- FIR systems remain stable after coefficient quantization
- Linear phase FIR systems remain linear phase after coefficient quantization, since the impulse response remains symmetric
- Coefficient quantization may lead to instability in IIR systems, as poles may move outside the unit circle
- Similarly to quantization noise, roundoff noise is modeled by an additive uniformly distributed white noise that is independent of the input signal (the linear noise model).
- Roundoff noise is minimized by performing quantization only after accumulation, but this requires $(2B + 1)$-bit adders
- In FIR structures the equivalent roundoff noise at the output is white
- IIR structures lead to roundoff noise shaping
- The least noisy IIR structure depends on the system
- Cascade and parallel forms are used to mitigate total roundoff noise