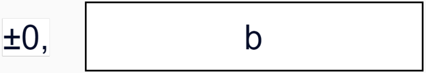
Finite precision arithmetic effects

Alberto Carini

Department of Engineering and Architecture, University of Trieste, Trieste, Italy

Finite precision arithmetic

- We have studied various structures for the implementation of digital filters. These structures are equivalent to each other when we operate with infinite precision arithmetic, but they exhibit different properties with finite precision arithmetic.
- DSP systems are typically implemented using floating-point or fixed-point arithmetic.
- In **floating-point arithmetic**, the number x is represented using an exponent γ and a mantissa m, such that $0.5 < |m| < 1$ and $x = m \cdot 2^{\gamma}$.
- In **fixed-point arithmetic**, the number x is assumed to belong to a certain interval and is represented with a fixed number of bits, with the decimal point occupying a fixed position within these bits.
- For example, values between -1 and $+1$ are represented using 1 sign bit and b bits after the decimal point.



±0, b

The smaller the value of $|x|$, the fewer significant bits are available in the representation of x (thus resulting in a larger relative error in the representation).

Finite precision arithmetic

- When using floating-point arithmetic with a high number of binary digits, we can reach conditions similar to infinite precision arithmetic.
- However, in cases where finite precision arithmetic is employed (common in hardware or software DSP implementations), it's essential to account for the effects of finite precision.
- In particular, we need to consider:
  1. The effects of quantization of filter coefficients,
  2. the A/D conversion noise,
  3. the uncorrelated noise due to rounding or truncation during multiplications,
  4. the overflow in additions,
  5. the limit cycles.

Filter coefficients quantization

- Filter design is typically conducted using infinite precision arithmetic or arithmetic with very high precision.
- However, in practical implementation, we must quantize the filter coefficients.
- As a result, the implemented filter no longer matches the transfer function we designed but rather approximates it to varying degrees, depending on the precision of the arithmetic used and the sensitivity of the realization to coefficient variations.
- From this perspective, the direct-form structure of IIR filters exhibits poor behavior, as coefficient quantization can easily lead to system instability.
- Conversely, lattice filters demonstrate better robustness to coefficient quantization.
- In situations where quantization involves a very limited number of bits, it becomes necessary to adopt structures highly robust to quantization.
- Moreover, it is essential to appropriately select the quantized parameters to meet the filter specifications; in this case, the quantized coefficients are directly designed.

Filter coefficients quantization

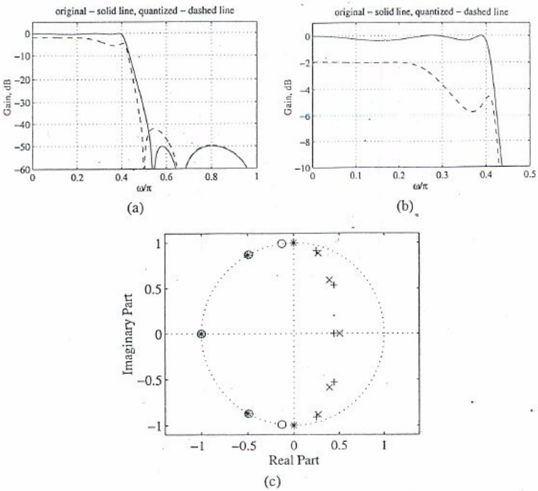


Figure 9.6: Coefficient quantization effects on a fifth-order IIR elliptic lowpass filter implemented in direct form: (a) fullband gain responses with unquantized (shown with solid line) and quantized coefficients (shown with dashed line), (b) passband details, and (c) pole-zero movements: Pole and zero locations of the filter with quantized coefficients denoted by "x" and "c", respectively, and pole and zero locations of the filter with unquantized coefficients denoted by "+" and "*", respectively.

5-bit quantization

Filter coefficients quantization

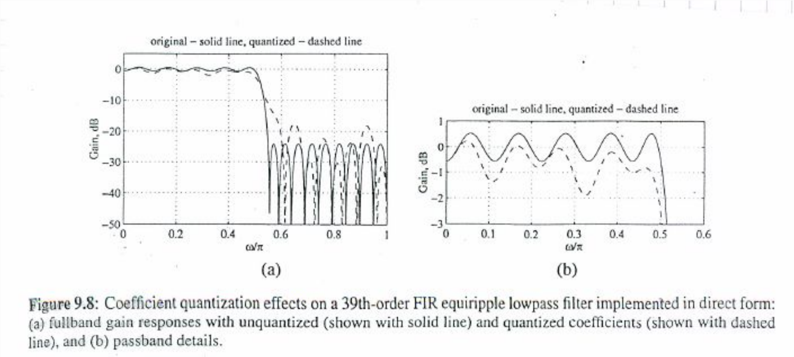
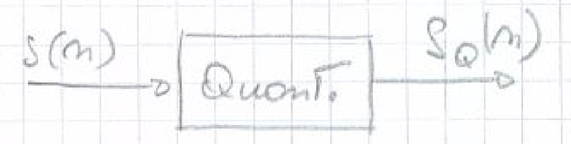


Figure 9.8: Coefficient quantization effects on a 39th-order FIR equiripple lowpass filter implemented in direct form: (a) fullband gain responses with unquantized (shown with solid line) and quantized coefficients (shown with dashed line), and (b) passband details.

5-bit quantization

A/D conversion noise

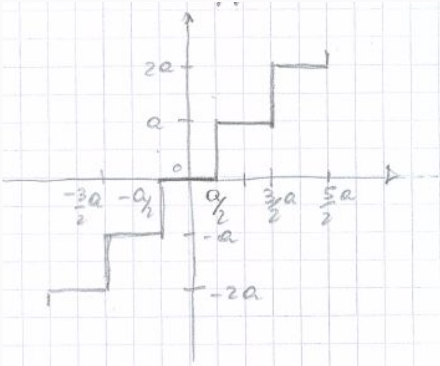
- The process of analog-to-digital conversion inevitably introduces an error due to signal quantization.



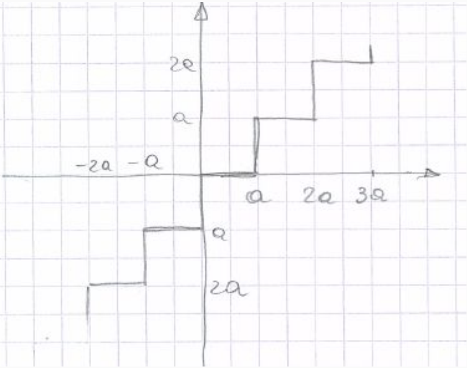
- $e(n) = s_Q(n) - s(n)$.
- The error introduced by analog-to-digital conversion, called **granular noise**, depends on the number of quantization levels and the type of quantization performed.
- The signal-to-quantization-noise ratio is influenced by the number of quantization levels.
- The A/D conversion noise is analyzed using a statistical approach.

A/D conversion noise

- The input-output relationship of the quantizer is illustrated by the following diagrams:



Rounding

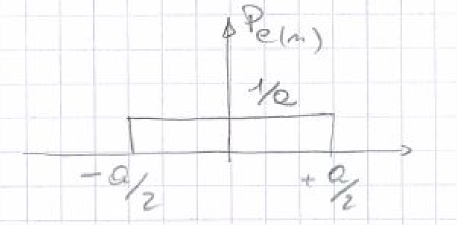


Truncation

- In case of rounding: $-\frac{1}{2}a \leq e(n) < \frac{1}{2}a$.
- In case of truncation: $0 \leq e(n) < a$.
- The expressions assume there is **no saturation** of the A/D converter, i.e., no clipping of the input signal; otherwise, the error can be much larger than a. We must as much as possible avoid saturations.

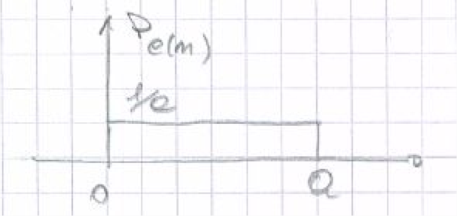
A/D conversion noise

- The quantization errors can be interpreted as generated by a random process akin to white stationary noise, with a constant probability density within the interval $[-\frac{a}{2}, +\frac{a}{2}]$ or $[0, a]$.
- It is assumed that this noise is uncorrelated with the signal.
- For rounding, we have:



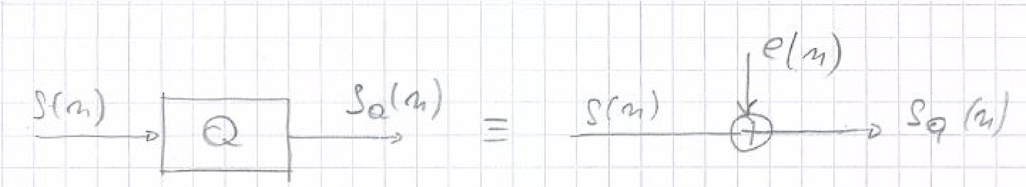
$$m_e = E[e(n)] = 0.$$
$$\sigma_e^2 = E[(e(n) - m_e)^2] = \int_{-\frac{a}{2}}^{+\frac{a}{2}} x^2 \frac{1}{a} \mathrm{d}x = \frac{a^2}{12}.$$

- For truncation, we have:



$$m_e = E[e(n)] = \frac{a}{2}.$$
$$\sigma_e^2 = E[(e(n) - m_e)^2] = \frac{a^2}{12}.$$

A/D conversion noise



- To evaluate the impact of additive quantization noise on the input signal $s(n)$, we can compute the **Signal-to-Quantization-Noise Ratio (SNR)** in decibels (dB), which is defined as:

$$\boxed{\mathrm{SNR}_{\mathrm{A/D}} = 10 \log_{10} \left(\frac{\sigma_s^2}{\sigma_e^2} \right) \mathrm{dB}.}$$

Here, σ_s^2 represents the input signal variance (i.e., the power of the input signal), and σ_e^2 denotes the variance of the quantization noise (i.e., the power of the quantization noise).

A/D conversion noise

- Consider a quantization scheme with 2^{b+1} levels between $-M$ and $+M$.
- In this case:

$$a = \frac{2M}{2^{b+1}} = \frac{M}{2^b} \qquad \Longrightarrow \qquad \sigma_e^2 = \frac{M^2}{12 \cdot 2^{2b}}.$$

- This leads to the Signal-to-Quantization-Noise Ratio as:

$$\mathrm{SNR}_{\mathrm{A/D}} = 10 \log_{10}\left(\frac{12 \cdot 2^{2b} \cdot \sigma_s^2}{M^2}\right) = 6.02\, b + 10.79 + 10 \log_{10}\left(\frac{\sigma_s^2}{M^2}\right) \mathrm{dB}.$$

- This equation can help us determine the minimum number of bits b required to achieve a specific signal-to-noise ratio.
- Notably, we observe that the **SNR increases by 6 dB for each additional unit of** b.

A/D conversion noise

- Is it always possible to prevent saturation of the analog-to-digital converter?
- In most cases, our signals are random signals with a certain distribution, and there exists a non-zero probability of exceeding a predefined range.
- Suppose our input signal follows a **Gaussian distribution** with a zero mean and standard deviation σ_s. The likelihood of a particular analog sample remaining within the range $[-\sigma_s K, +\sigma_s K]$ is given by:

$$2\Phi(K) - 1 = \sqrt{\frac{2}{\pi}} \int_0^K e^{-y^2/2} \mathrm{d}y, \tag{1}$$

  where $\Phi(.)$ is the cumulative probability of a zero mean unit variance Gaussian distribution.
- For $K = 2$, the probability of an analog sample remaining within the range $[-\mathbf{2}\sigma_\mathbf{s}, +\mathbf{2}\sigma_\mathbf{s}]$ is $\mathbf{0.9544}$. On average, out of every $10,000$ input samples, approximately 456 samples fall outside this range.
- For $K = 3$, the probability of an analog sample remaining within the range $[-\mathbf{3}\sigma_\mathbf{s}, +\mathbf{3}\sigma_\mathbf{s}]$ is $\mathbf{0.9973}$. On average, out of every $10,000$ input samples, approximately 37 samples fall outside this range.
- For $K = 4$, the probability of an analog sample remaining within the range $[-\mathbf{4}\sigma_\mathbf{s}, +\mathbf{4}\sigma_\mathbf{s}]$ is $\mathbf{0.999936}$. On average, out of every $1,000,000$ input samples, approximately 64 samples fall outside this range. This range is generally considered more than sufficient to prevent clipping during conversion.

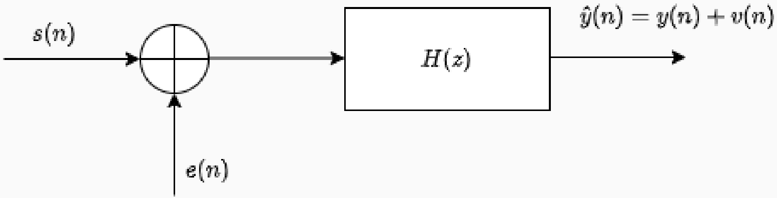
A/D conversion noise

- To prevent saturation of the analog-to-digital converter, scaling the input signal is a common approach.
- Let's explore its impact. Assuming we scale the input signal by a factor A, the variance of the scaled input, $As(n)$, becomes $A^2\sigma_x^2$.
- Consequently, the expression for signal-to-quantization noise ratio (SNR) transforms to:

$$\mathrm{SNR}_{\mathrm{A/D}} = 10\log_{10}\left(\frac{12\cdot 2^{2b}\cdot A^2\cdot\sigma_s^2}{M^2}\right) = 6.02\,b + 10.79 + 10\log_{10}\left(\frac{\sigma_s^2}{M^2}\right) + 20\log_{10}(A).$$

- When $A > 1$, the SNR increases, but so does the probability of overflow.
  Conversely, $A < 1$ reduces the probability of overflow but also lowers the SNR.
- To attain the highest possible SNR without distorting the signal, it's essential to align the analog sample range as closely as possible with the full-scale range of the A/D converter.

A/D conversion noise



- The quantized signal is commonly processed using a linear time-invariant discrete-time system $H(z)$.
- Modelling the quantized signal as the sum of the unquantized signal $s(n)$ and a quantization error $e(n)$, assuming **linearity** along with **uncorrelation between** $s(n)$ **and** $e(n)$, the **output** comprises **two components:**

  $y(n)$ associated with the unquantized input $s(n)$, and $v(n)$ associated with the quantization error $e(n)$.
- While $e(n)$ can be assumed to exhibit characteristics of white noise, $v(n)$ does not.

A/D conversion noise

- Specifically,

$$v(n) = \sum_{m=-\infty}^{+\infty} h(m)e(n-m),$$

  where the quantization noise is convolved with the impulse response of the filter. The mean of the output noise $v(n)$ is

$$m_v = m_e H(e^{j0})$$

  and the variance σ_v^2 is

$$\sigma_v^2 = \frac{\sigma_e^2}{2\pi}\int_{-\pi}^{+\pi} |H(e^{j\omega})|^2 \mathrm{d}\omega = \sigma_e^2 \sum_{m=-\infty}^{+\infty} |h(m)|^2.$$

- The noise output power spectrum is given by

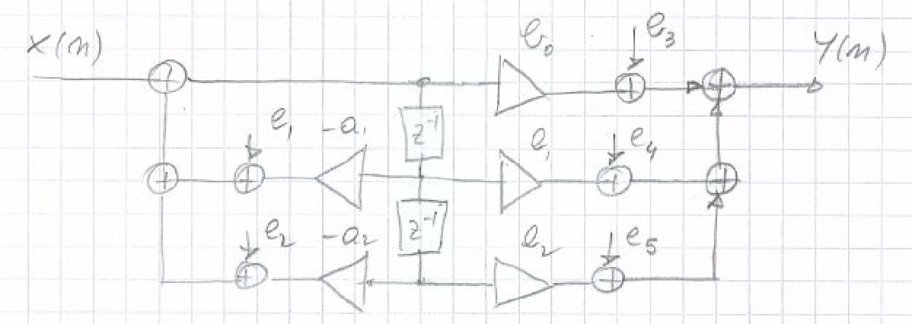
$$P_{vv}(\omega) = \sigma_e^2 |H(e^{j\omega})|^2.$$

Uncorrelated noise due to rounding or truncation in multiplications

- This noise is analyzed similarly to quantization noise.
- When two numbers with N digits are multiplied, the result typically has $2N - 1$ digits.
- To represent this result with N digits, **rounding or truncation** is necessary.
- This rounding or truncation process introduces noise, which is statistically treated similarly to quantization noise.
- The noise is assumed to originate from a white stationary noise source with a constant probability density distribution.
- It's worth noting that these various noise sources are assumed to be uncorrelated with each other and with the input signal.

Uncorrelated noise due to rounding or truncation in multiplications

- For instance, in the case of a second-order Infinite Impulse Response (IIR) filter, there exists a noise source associated with each multiplier operation.

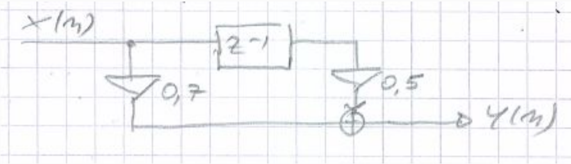


- From the selection of the quantization level, it becomes feasible to ascertain the output signal-to-noise ratio resulting from rounding or truncation.
- It's worth noting that different filter structures generally exhibit distinct noise propagation properties.

Overflow in additions

- This issue is especially significant when operating with fixed-point arithmetic.
- When dealing with numbers between 0 and 1 in fixed-point representation, the sum of two such numbers may exceed 1, leading to overflow beyond the representation limits (assuming the maximum value is 1).
- Overflow typically manifests as an impulsive noise with high amplitude at the output of the system.
- It's essential to design the system in advance to minimize the occurrence of overflows or to ensure that they have minimal probability of happening.
- To prevent overflows, it is crucial to appropriately scale all signals using constant multiplicative factors.

- For example:



- If the input signal satisfies $-1 \leq x(n) \leq +1$, then after processing, the output signal $y(n)$ may range from -1.2 to 1.2, posing a risk of overflow when using fixed-point arithmetic with a maximum value of 1.
- However, by multiplying the input signal $x(n)$ by $\frac{1}{1.2}$, we obtain an output signal that is always bounded between -1 and 1.
- This scaled output signal is identical to the original signal but scaled by a factor of $\frac{1}{1.2}$.
- This scaling ensures that the output remains within the representable range without the risk of overflow.

Overflow in additions

- While completely avoiding the overflow problem by appropriately scaling the input signal is indeed possible, it often comes at the cost of significantly penalizing the signal-to-quantization-noise ratio, as it reduces σ_s^2.
- In general, we can minimize the probability of overflow at internal nodes by inserting scaling multipliers at selected points in the digital filter structure to appropriately scale the internal signal levels.
- In many cases, most of these scaling multipliers can be absorbed by the existing multipliers in the structure.

Dynamic range scaling

- Let's consider a digital filter structure, and for a given node r, let the signal to be scaled be denoted as $u_r(n)$.
- We assume that all fixed-point numbers are represented as binary fractions and that the input sequence $x(n)$ is bounded by unity, i.e., $|x(n)| \leq 1$ for all n.
- The objective of scaling is to ensure that
$$\boxed{|u_r(n)| \leq 1}$$
for all nodes r and time instances n.
- Let's define the **scaling transfer function** $F_r(z)$ as the transfer function from the input to the r-th node.
- Its inverse z-transform $f_r(n)$ represents the impulse response from the filter input to the r-th node.
- $u_r(n)$ can be expressed as the convolution of $f_r(n)$ and $x(n)$:

$$u_r(n) = \sum_{k=-\infty}^{+\infty} f_r(k)x(n-k)$$

Dynamic range scaling: an absolute bound

- Then,

$$|u_r(n)| = \left| \sum_{k=-\infty}^{+\infty} f_r(k)x(n-k) \right| \leq \sum_{k=-\infty}^{+\infty} |f_r(n)|.$$

- Thus, we have $|u_r(n)| \leq 1$ if

$$\boxed{\sum_{k=-\infty}^{+\infty} |f_r(n)| \leq 1.}$$

- It can be proven that the above condition is both **necessary and sufficient** to guarantee no overflow.
- If it is not satisfied by the unscaled realization, we can scale the input signal with a multiplier K, where

$$\boxed{K = \frac{1}{\max_r \sum_{k=-\infty}^{+\infty} |f_r(n)|}.}$$

- This scaling rule is based on a **worst-case bound** and significantly reduces the output SNR.

Dynamic range scaling

- More practical and easy-to-use scaling rules can be derived in the frequency domain if some information about the input signals is known a priori.
- In what follows, we will assume the input $x(n)$ to be a deterministic signal with Fourier transform $X(e^{j\omega})$, and we will derive the bounds in terms of $\mathcal{L}_p$-norms.
- The $\mathcal{L}_p$-**norm** of $X(e^{j\omega})$ is defined as

$$\boxed{||X||_p = \left(\frac{1}{2\pi}\int_{-\pi}^{+\pi} |X(e^{j\omega})|^p \mathrm{d}\omega\right)^{1/p}}$$

  In most cases, the values of p used are 1, 2, and ∞.
- For $p = 2$, $||X||_2$ represents the **root-mean-square (rms)** value of $X(e^{j\omega})$.
- For $p = 1$, $||X||_1$ indicates the **mean absolute value** of $X(e^{j\omega})$.
- In the case of a continuous $X(e^{j\omega})$, $\lim_{p\to\infty} ||X||_p$ exists and represents the **peak absolute value**:

$$||X||_\infty = \max_{-\pi\leq\omega\leq+\pi} |X(e^{j\omega})|.$$

$\mathcal{L}_\infty$-bound

- Since

$$u_r(n) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} F_r(e^{j\omega})X(e^{j\omega})e^{j\omega n}\mathrm{d}\omega,$$

we have

$$\begin{aligned}|u_r(n)| &\leq \frac{1}{2\pi}\int_{-\pi}^{+\pi} |F_r(e^{j\omega})||X(e^{j\omega})|\mathrm{d}\omega \\ &\leq ||F_r||_\infty \frac{1}{2\pi}\int_{-\pi}^{+\pi} |X(e^{j\omega})|\mathrm{d}\omega \\ &\leq ||F_r||_\infty \cdot ||X||_1.\end{aligned}$$

- If $||X||_1 \leq 1$, then $|u_r(n)| \leq 1$ if

$$||F_r||_\infty \leq 1.$$

- In general, this scaling rule is rarely used since, in practice, $||X||_1 \leq 1$ does not typically hold for most encountered input signals.

$\mathcal{L}_2$-bound

- Applying the **Schwartz inequality** to

$$u_r(n) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} F_r(e^{j\omega})X(e^{j\omega})e^{j\omega n}\mathrm{d}\omega,$$

  we obtain that

$$|u_r(n)|^2 \leq \left(\frac{1}{2\pi}\int_{-\pi}^{+\pi} |F_r(e^{j\omega})|^2\mathrm{d}\omega,\right)\cdot\left(\frac{1}{2\pi}\int_{-\pi}^{+\pi} |X(e^{j\omega})|^2\mathrm{d}\omega,\right),$$

  i.e.,

$$|u_r(n)|^2 \leq ||F_r||_2 \cdot ||X||_2.$$

- If the input signal has finite energy bounded by unity, i.e., $||X||_2 \leq 1$, then preventing adder overflow can be achieved by scaling the filter such that the $\mathcal{L}_2$-norm of the transfer functions from the input to all adder outputs are bounded by unity:

$$||F_r||_2 \leq 1 \quad \forall r.$$

A more general scaling rule

- According to the **Holder's inequality**,

  $$|u_r(n)|^2 \leq ||F_r||_p \cdot ||X||_q,$$

  for all $p, q \geq 1$ satisfying $(1/p) + (1/q) = 1$.
- The $\mathcal{L}_\infty$-bound is obtained for $p = \infty$ and $q = 1$.
  The $\mathcal{L}_2$-bound is obtained for $p = q = 2$.
- Another useful bound is the $\mathcal{L}_1$-bound obtained for $p = 1$ and $q = \infty$.
- Provided $||X||_q < 1$, then $|u_r(n)| \leq 1$ if

  $$||F_r||_p \leq 1.$$

- In many structures, all scaling multipliers can be absorbed into the existing feedforward multipliers without increasing the total number of multipliers.
- Let's consider, for example, the commonly used cascade form IIR digital filter structure.

- Let us consider first the unscaled IIR filter structure.

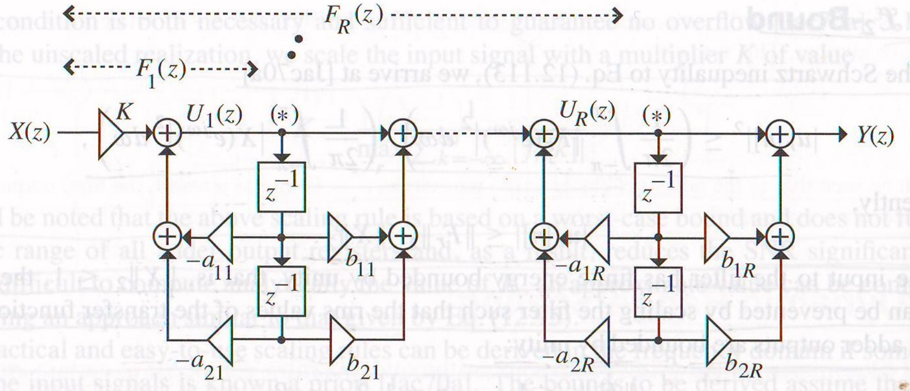


Figure 12.27: An unscaled cascade realization of second-order IIR sections.

Scaling the Cascade form IIR filter structure

$$H(z) = K \prod_{i=1}^{R} H_i(z)$$

where

$$H_i(z) = \frac{B_i(z)}{A_i(z)} = \frac{1 + b_{1i} z^{-1} + b_{2i} z^{-2}}{1 + a_{1i} z^{-1} + a_{2i} z^{-2}}.$$

- The nodes to be scaled are those marked with (*) in the figure and correspond to the inputs of the multipliers in each second-order section.
- The scaling transfer functions are defined as:

$$F_r(z) = \frac{K}{A_r(z)} \prod_{l=1}^{r-1} H_l(z).$$

Scaling the Cascade form IIR filter structure

- The scaled transfer function is shown in the following figure.

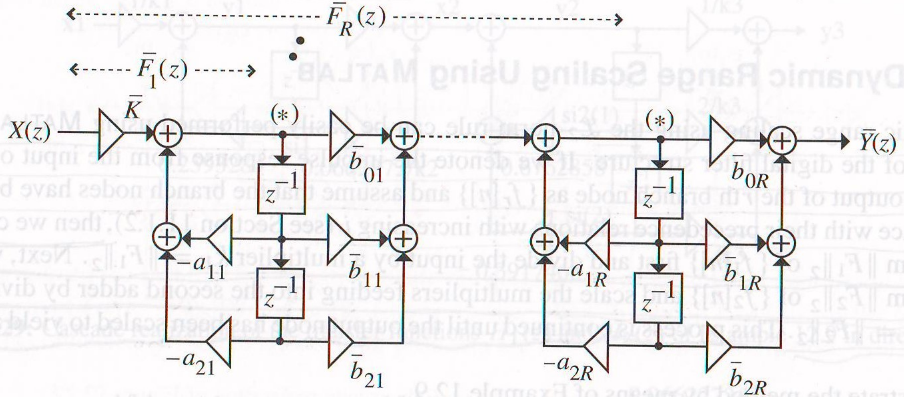


Figure 12.28: The scaled cascade realization.

- The scaling process has introduced a new multiplier $\bar{b}_{0l}$ in each second-order section.

Scaling the Cascade form IIR filter structure

- Let us denote

$$||F_r||_p = \alpha_r \qquad \forall r$$

$$||H||_p = \alpha_{R+1}$$

and let us choose the scaling constant as

$$\bar{K} = \beta_0 K; \qquad \bar{b}_{lr} = \beta_r b_{lr} \qquad l = 0, 1, 2; \qquad r = 1, 2, \dots, R.$$

- It can be easily verified that

$$\bar{F}_r = \left(\prod_{i=0}^{r-1} \beta_i \right) F_r(z)$$

$$\bar{H} = \left(\prod_{i=0}^{R} \beta_i \right) H(z)$$

Scaling the Cascade form IIR filter structure

- With the scaling we want

$$||\bar{F}_r||_p = \left(\prod_{i=0}^{r-1} \beta_i\right) ||F_r||_p = \alpha_r \left(\prod_{i=0}^{r-1} \beta_i\right) = 1, \qquad r = 1, 2, \dots, R$$

$$||\bar{H}||_p = \left(\prod_{i=0}^{R} \beta_i\right) ||H||_p = \alpha_{R+1} \left(\prod_{i=0}^{R} \beta_i\right) = 1.$$

- Solving these equations we arrive at

$$\beta_0 = \frac{1}{\alpha_1},$$
$$\beta_r = \frac{\alpha_r}{\alpha_{r+1}}, \qquad r = 1, 2, \dots, R.$$

Scaling the Cascade form IIR filter structure

- There are many possible cascade realizations of a higher-order IIR transfer function achieved through various pole-zero pairings and orderings.
- In fact, for a cascade of R second-order sections, there are $(R!)^2$ different possible realizations.
- Each of these realizations will have different scaling and output noise power.
- What is the optimal pole-zero pairing and ordering of the sections in a cascade realization?
- There exists a simple heuristic set of rules.

Scaling the Cascade form IIR filter structure

- For the pole-zero pairing:
- First, pair the complex pole pair closest to the unit circle with the nearest complex zero pair.
- Next, pair the complex pole pair closest to the previous set of poles with its nearest complex zero pair.
- Repeat this process until all poles and zeros have been paired.

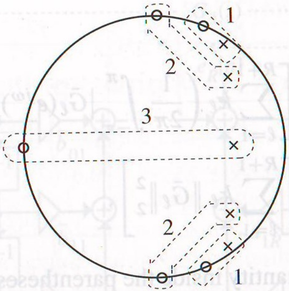


Figure 12.32: Illustration of the pole-zero pairing and ordering rules.

- The above procedure will reduce the peak gain of the section characterized by the paired poles and zeros.

Scaling the Cascade form IIR filter structure

- As for the ordering of the sections, it depends on the $\mathcal{L}_p$-norm used for scaling.
- If the $\mathcal{L}_2$-norm is used, the ordering of the paired sections does not significantly influence the output noise power.
- If an $\mathcal{L}_\infty$-scaling is employed, the section with poles closest to the unit circle and exhibiting the most pronounced magnitude response should be placed closest to the output end.
  The next section should be the one with the next most pronounced magnitude response, and so on.
  The first section should be the one with the least pronounced magnitude response. The same rule applies in the general case.

Limit cycles

- In recursive systems, the nonlinearities associated with finite precision arithmetic (such as rounding and overflows) can induce **periodic oscillations** at the system output **or** yield **a constant output**, even when the input is zero or remains constant.
- These phenomena are known as **limit cycles** and are directly caused by errors arising from rounding in multiplications or overflows in additions.
- The problem is especially notable in IIR filters with poles near the unit circle.

Limit cycles: example

- Consider the system,

$$y(n) = 0.95y(n-1) + x(n),$$

$$H(z) = \frac{1}{1 - 0.95z^{-1}}$$

with $x(n) = 0$ for all $n \geq 0$, $y(-1) = 13$.

- Let us assume to round $y(n)$ to an integer value:

n	$y(n)$ exact	$y(n)$ rounded
-1	13	13
0	12.35	$12 \longleftarrow (12.35)$
1	11.73	$11 \longleftarrow (11.4)$
2	11.14	$10 \longleftarrow (10.45)$
3	10.75	$10 \longleftarrow (9.5)$
4	10.05	$10 \longleftarrow (9.5)$
5	9.5	$10 \longleftarrow (9.5)$

- Another example is the first-order IIR filter depicted in the following figure, which includes a quantizer after the multiplication

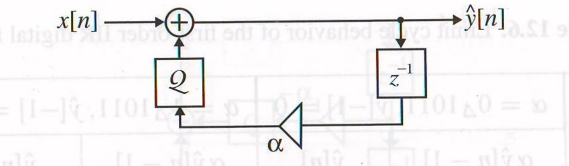


Figure 12.48: A first-order IIR filter with a quantizer.

- The digital filter is assumed to be implemented using 6-bit fractional arithmetic with a quantization step of $\delta = 2^{-5}$.
- The input signal is set as $x(0) = 0.4$, and $x(n) = 0$ for $n > 0$, with the initial condition $y(-1) = 0$.

Limit cycles: example

- The following figure shows the limit cycles that can be observed for $\alpha = 0.6$, resulting in a constant non-zero output, and for $\alpha = -0.6$, leading to an oscillatory output.

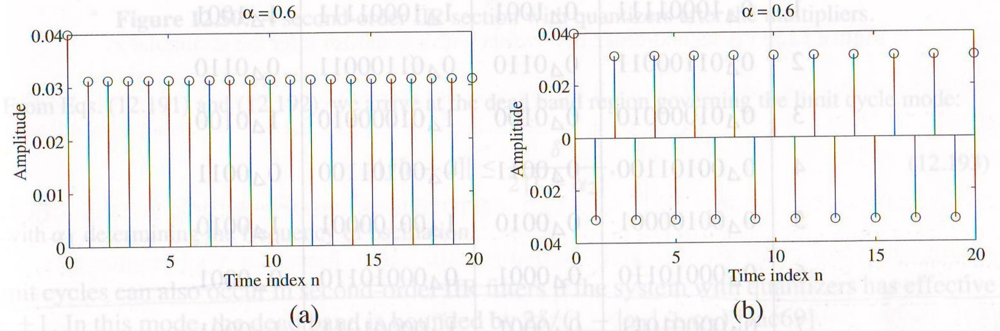


Figure 12.49: Illustration of limit cycles in a first-order IIR digital filter: (a) $\alpha = 0.6$ and (b) $\alpha = -0.6$.

Limit cycles: example

- The following figure shows the limit cycles that can be observed for $\alpha = 0.6$, resulting in a constant non-zero output, and for $\alpha = -0.6$, leading to an oscillatory output.

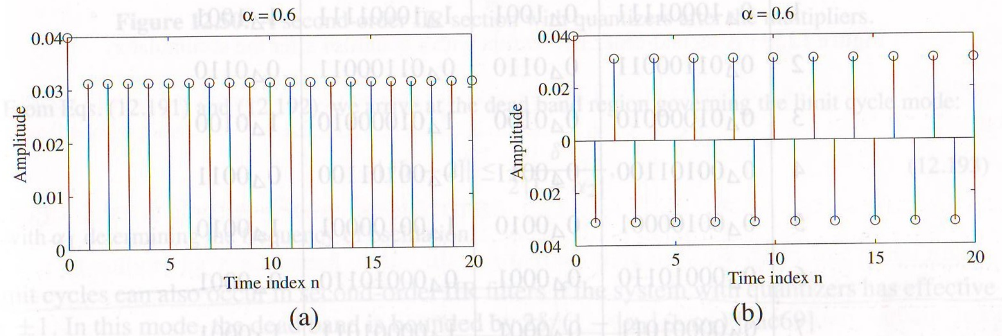


Figure 12.49: Illustration of limit cycles in a first-order IIR digital filter: (a) $\alpha = 0.6$ and (b) $\alpha = -0.6$.

Limit cycles

- The output signal can become entirely independent of the input in the presence of limit cycles.
- There are conditions under which these limit cycles can be avoided, such as utilizing truncation towards 0 instead of rounding, or adding a small random noise to the input.
- Generally, increasing the precision of the arithmetic improves the behavior of the recursive system, thereby reducing the amplitude of the limit cycles.

Bibliography

- For more information study:

  S. K. Mitra, "Digital Signal Processing: a computer based approach," 4th edition, McGraw-Hill, 2011
  Chapter 12.1, pp. 664-665
  Chapter 12.2, pp. 665-667
  Chapter 12.4.1, pp. 668-672
  Chapter 12.5, pp. 681-687
  Chapter 12.7, pp. 695-699 and 702-705
  Chapter 12.11, pp. 719-721

Unless otherwise specified, all images have either been originally produced or have been taken from S. K. Mitra, "Digital Signal Processing: a computer based approach."