NTNU – Trondheim
Norwegian University of
Science and Technology

TTT4120 Digital Signal Processing
Fall 2016

Finite-precision effects

Prof. Stefan Werner
stefan.werner@ntnu.no
Office B329

Lecture in course book*

- Proakis, Manolakis Digital Signal Processing, 4th Ed.
  - 9.4 Representation of Numbers
  - 9.6 Round-Off Effects in Digital Filters

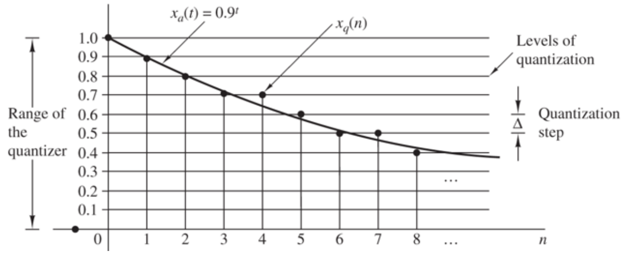
A compressed overview of topics treated in the lecture, see "Filter implementation" on ItsLearning

*Level of detail is defined by lectures and problem sets

Contents and learning outcomes

- Representations of numbers
- Limit cycles and scaling
- Statistical characterization

Introduction



- Until now, coefficients and operations of filter designs and implementations expressed using infinite-precision numbers
- In practice, finite-word-length is required in any digitalization
- Especially low-power and small-area components in wireless communications

Number representation…

- Consider the representation of numbers for digital computations
- Limited (usually fixed) number of digits to represent a number
- Fixed decimal point representation
  - Fixed amount of digits and fixed decimal point placement

$$13.234, 01.345, 00.999, \ldots$$

- Floating (decimal) point representation
  - Decimal number represented by a mantissa and an exponent

$$2.0 \cdot 10^2, 4.9 \cdot 10^8, \ldots$$

5

Number representation

- Finite precision errors not a problem in floating-point arithmetic
- Finite word length causes problems in fixed-point arithemetic
- Fixed-point implementation used only when
  - speed,
  - power
  - size,
  - and cost

  are important.

6

Finite-precision effects

- Overflow
- Quantization of filter coefficients
- Signal quantization
  - A/D conversion
  - Round-off noise
  - Limit cycles

Fixed-point representation

- Generalization of the familiar decimal representation of a number
  - String of digits with a decimal point

$$X = (b_{-A}, \dots, b_{-1}, b_0, b_1, \dots, b_B)_r$$

$$= \sum_{i=-A}^{B} b_i r^{-i}, 0 \leq b_i \leq (r-1)$$

where b_i represents the digit and r is the base (radix)

- Focus on binary representation: Generalization of the familiar decimal representation of a number $b_i \in \{0,1\}$, and $r = 2$

$$(101.01)_2 = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 5.25$$

- Most significant bit (MSB) b_{-A}, least significant bit (LSB) b_B

Fixed-point representation…

- Fraction format, $|X| < 1 \Rightarrow (A = 0, B = n - 1)$, and

$$X = (b_0, b_1, \dots, b_{n-1})_2$$

can represent unsigned integers from 0 to $1 - 2^{-n}$
- Format for positive fractions: $X = 0.b_1 b_2 \dots b_B = \sum_{i=1}^{B} b_i 2^{-i}$

$$0.011 = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$$

- MSB b_0 set to zero to represent the positive sign
- Negative fraction: $X = -0.b_1 b_2 \dots b_B = -\sum_{i=1}^{B} b_i 2^{-i}$
  – Three different ways to represent negative fractions

Fixed-point representation…

- Signed-magnitude (SM) format
  – MSB is set to 1 to represent negative sign

$$X_{SM} = 1.b_1 b_2 \dots b_B = 1 \times 2^0 + \sum_{i=1}^{B} b_i 2^{-i}, \ X \leq 1$$

$$1.011 = -\frac{3}{8}$$

- Symmetry: as many positive as negative values
- Disadvantages
  – Two ways of expressing ‘zero’: ‘plus zero’ and ‘minus zero’
  – Addition and subtractions are more complicated

Fixed-point representation…

- One's-complement format
  - Negative numbers represented as

$$X_{1C} = 1.\bar{b}_1\bar{b}_2 \ldots \bar{b}_B = 1 \times 2^0 + \sum_{i=1}^{B}(1 - b_i)2^{-i}\, X \leq 1$$

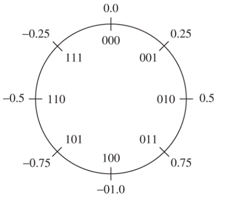
$$X_{1C} = 1.100 = -\frac{3}{8}$$

Fixed-point representation…

- Two's-complement format
  - Most commonly used
  - Negative numbers represented as

$$X_{2C} = 1.\bar{b}_1\bar{b}_2 \ldots \bar{b}_B + 0\,0 \cdots 01$$
$$= X_{1C} + 2^{-B}$$
$$\frac{3}{8} = 0.011 \;\Rightarrow X_{2C} = X_{1C} + 0.001 = 1.101$$

- Range for $(B+1)$-bit numbers is from -1 to $1 - 2^{-B}$



(2 + 1)-bit numbers

Fixed-point representation…

- Summary advantages of two's-complement format
  - Provides for all $2^B + 1$ distinct representations for a B-bit fractional representation. Only one representation for zero.
  - Complement of a complement is the number itself

$$\bar{X} = X_{2C} \Rightarrow \bar{X}_{2C} = X$$

  - Unifies subtraction and addition operations (subtractions are essentially additions)
  - In a sum of more than two numbers, the internal overflow do not affect the final result as long as the result is within the range

13

Floating-point representation

- Floating-point represented by a mantissa and an exponent

$$X = M \cdot 2^E$$

- Mantissa and exponent require a sign bit for representing positive and negative numbers
- Floating-point form can cover a larger dynamic range than finite-precision for same number of bits by varying the resolution across the range
  - For the same range floating point, provides finer resolution for small numbers but coarser resolution for the larger numbers
  - Fixed-point provides a uniform resolution throughout the range

$$X_1 = 5 = 0.101 \cdot 2^{0.11} = 0.101 \cdot 2^3 = (101)_2 = 5$$

$$X_2 = \frac{3}{8} = 0.110 \cdot 2^{1.01} = 0.110 \cdot 2^{-1} = (0.011)_2 = \frac{3}{8}$$

14

Fixed-point implementation

- The way additions and multiplications are carried out using fixed-point numbers depends on the format used for negative fraction
  - Two's-complement addition

$$\frac{4}{8}-\frac{3}{8}=\frac{4}{8}+\left(-\frac{3}{8}\right)=(0.100)_2+(1.101)_2=(0.001)_2=\frac{1}{8}$$

  - Carry-bit does not propagate beyond MSB

$$\frac{6}{8}+\frac{3}{8}=(0.110)_2+(0.011)_2=(1.001)_2=-\frac{7}{8}$$

15

Fixed-point implementation…

- The limited dynamic range can lead to large errors
  - In previous example the error equals the total dynamic range

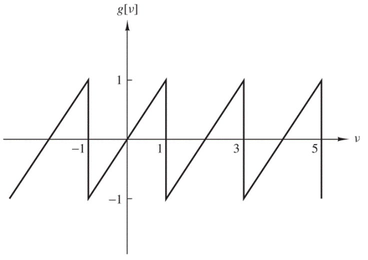


Figure 9.6.4 Characteristic functional relationship for two's-complement addition of two or more numbers.

  - Problem prevented by scaling or saturation

16

Fixed-point implementation…

- The way additions and multiplications are carried out using fixed-point numbers depends on the format used for negative fraction
  - Two's-complement multiplication

$$\frac{3}{8} \cdot \frac{3}{8} = (0.011)_2 \cdot (0.011)_2 = (0.001001)_2 = \frac{9}{64}$$

  - Will be rounded to $(0.001)_2 = \frac{1}{8}$
  - Rounding error $E_r = \frac{9}{64} - \frac{8}{64} = \frac{1}{64}$

17

Fixed-point implementations…

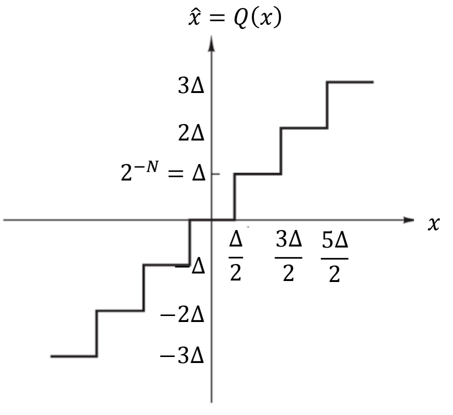
- Quantization of real-valued signal x into $N = B + 1$ bits

$$\hat{x} = Q(x) = x + \epsilon$$

- Error ϵ limited in range

$$-\frac{\Delta}{2} \leq \epsilon \leq \frac{\Delta}{2} = \frac{2^{-N}}{2}$$

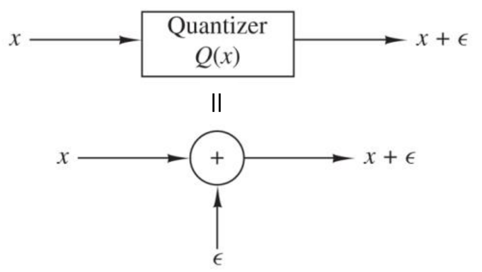
- Errors uniformly distributed



18

Fixed-point implementations…

- Linear model for analyzing quantization effects



- PDF of quantization error:

$$p_E(\epsilon) = \begin{cases} \frac{1}{\Delta}, & |\epsilon| \leq \frac{\Delta}{2} \\ 0, & \text{else} \end{cases}$$

19

Statistical characterization of errors…

- Error power (variance):

$$\sigma_\epsilon^2 = \int_{-\infty}^{\infty} \epsilon^2 p_E(\epsilon) d\epsilon = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \epsilon^2 \frac{1}{\Delta} d\epsilon$$

$$= \frac{\epsilon^3}{3\Delta}\Big|_{\epsilon=-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = \frac{1}{3\Delta}\left[\left(\frac{\Delta}{2}\right)^3 - \left(-\frac{\Delta}{2}\right)^3\right]$$

$$= \frac{\Delta^2}{12} = \frac{2^{-2N}}{12}$$

20

Fixed-point implementation…

- Fixed-point implementations lead to four possible nonlinearities
  1. Rounding due to limited resolution (number of bits)
  2. Overflow due to limited dynamic range
  3. Inaccuracy in filter specs due to use of quantized filter coefficients
  4. Limit cycles (oscillations) due to quantized filter coefficients and rounding
- We will look at Items 1 and 2

21

Effects in digital filters: scaling

- Scaling to prevent overflow
  - Signal must be scaled before addition to make sure that the sum is less than unity, i.e., ensure that $x_1[n] + x_2[n] < 1$
  - Suppose that we pass sequence $x[n]$ through filter $h[n]$

$$|y[n]| = |\sum_{m=-\infty}^{\infty} h[m]x[n-m]|$$

$$\leq \sum_{m=-\infty}^{\infty} |h[m]||x[n-m]|$$

- Suppose that $x[n]$ is upper bounded by unity, $|x[n]| < A_x$, we get

$$|y[n]| \leq A_x \sum_{m=-\infty}^{\infty} |h[m]|, \forall n$$

- If dynamic range is limited to [-1,1), how to scale $x[n]$ such that $|y[n]| < 1$?

22

Effects in digital filters: scaling

- Overflow is prevented if $x[n]$ scaled such that

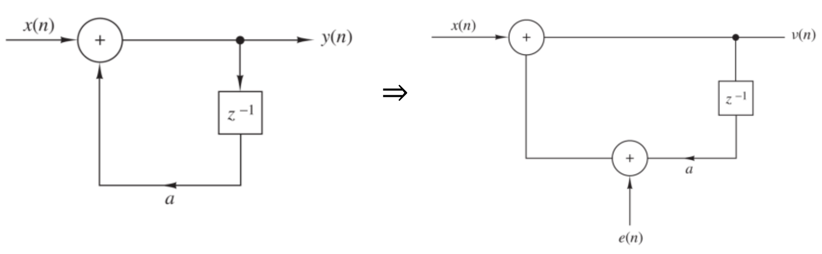
$$A_x < \frac{1}{\sum_{m=-\infty}^{\infty}|h[m]|}$$

- Scaling reduces the signal resolution and signal power
- Reduced signal-to-noise ratio (SNR)

23

Effects in digital filters: quantization

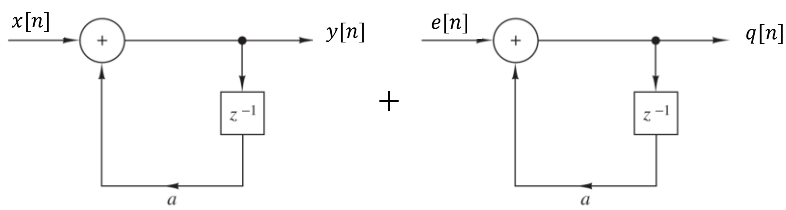
- Analysis of quantization effects in digital filters is hard
- Effects of quantizing the product of two numbers and clipping the sum of two numbers not easily modeled for large systems
- Model the quantization error as an additive noise sequence $e[n]$
- Example: Single pole filter



24

Effects in digital filters: quantization…

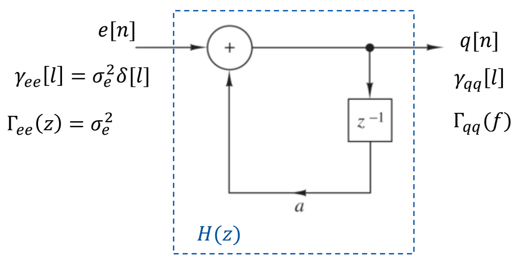
- Output can be separated into two components
  - One is due to the input seqence $x[n]$
  - Second is due to white sequence $e[n]$



- We can now calculate the ouput power due to quantization error

Effects in digital filters: quantization…

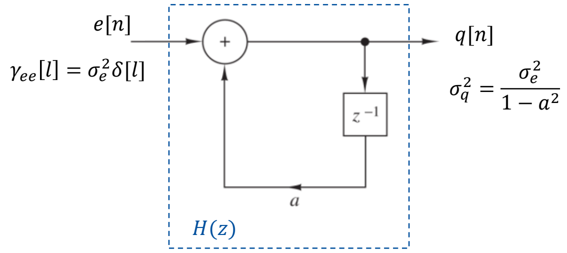
- Variance of the quantization error σ_q^2:



$$\sigma_q^2 = E\{q^2[n]\} = \gamma_{qq}[0] = \frac{\sigma_e^2}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega$$

$$= \sigma_e^2 \sum_{k=-\infty}^{\infty} h^2[k] = \frac{\sigma_e^2}{1-a^2}$$

Effects in digital filters: quantization…

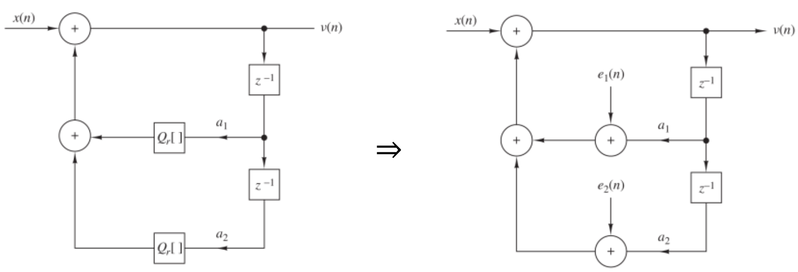


- Observations from single-pole filter
  - Noise power at the output is increased relative to the input noise

27

Effects in digital filters: quantization

- Example: Two-pole filter



- Same idea as in single-pole filter
  - Output noise power obtained by exciting system with

$$e[n] = e_1[n] + e_2[n]$$

28

Effects in digital filters: quantization…

- Digital filters are linear systems, but when quantizers are incorporated, they become nonlinear
  - Possible to have an output sequence even in the abscence of input signal
  - Limit cycles: Undersired oscillations at the output of a recursive filter as a result of quantization (rounding and overflow)
- Example: $y[n] = -\frac{1}{2}y[n-1] + x[n]; y[-1] = 0, n \geq 0$

  Determine $y[n]$ for $x[n] = \frac{7}{8}\delta[n]$, assuming 3-bit quantizer in the multiplication

29

Effects in digital filters: quantization…

- Quantized output: $\hat{y}[n] = Q\left[-\frac{1}{2}y[n-1]\right] + x[n]; \hat{y}[-1] = 0,$

  $B = 3$ bits (3 fraction bits and one sign bit)

$$\hat{y}[0] = x[0] \qquad = +\frac{7}{8}$$

$$\hat{y}[1] = Q\left[-\frac{1}{2}\left(+\frac{7}{8}\right)\right] = Q\left[-\frac{7}{16}\right] \qquad = -\frac{1}{2}$$

$$\hat{y}[2] = Q\left[-\frac{1}{2}\left(-\frac{1}{2}\right)\right] = Q\left[+\frac{1}{4}\right] \qquad = +\frac{1}{4}$$

$$\hat{y}[3] = Q\left[-\frac{1}{2}\left(+\frac{1}{4}\right)\right] = Q\left[-\frac{1}{8}\right] \qquad = -\frac{1}{8}$$

$$\hat{y}[4] = Q\left[-\frac{1}{2}\left(-\frac{1}{8}\right)\right] = Q\left[+\frac{1}{16}\right] \qquad = +\frac{1}{8}$$

$$\vdots$$

30

Fixed-point implementations…

- Quantization of filter coefficients
  - Leads to non-ideal frequency response
  - Direct-form structures are sensitive to coefficient rounding for filter orders $N > 2$
  - Use of parallel- and/or cascade structures

Effects in digital filters: quantization…

- Second-order IIR section
- Quantization of filter coefficients $2r\cos\theta$ and r^2 with 4 bits

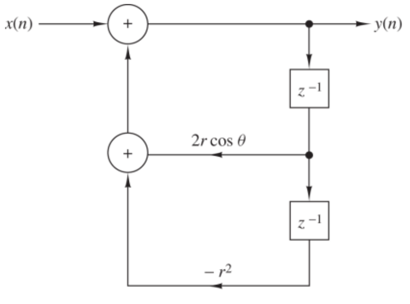


Figure 9.5.2 Realization of a two-pole IIR filter.

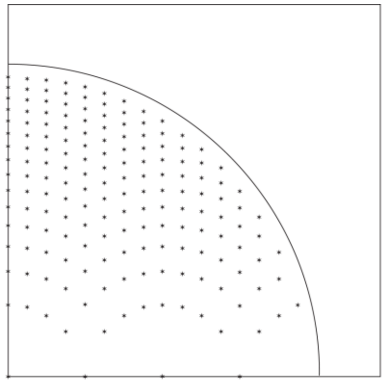


Figure 9.5.3 Possible pole positions for two-pole IIR filter realization in Fig. 9.5.2.

Effects in digital filters: quantization…

- Alternative structure for second-order IIR section (more mult)
- Quantization of filter coefficients $2r\cos\theta$ and $2r\sin\theta$ with 4 bits

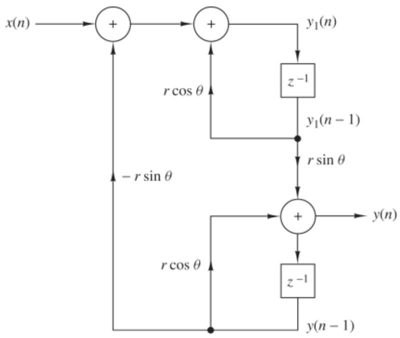


Figure 9.5.4 Coupled-form realization of a two-pole IIR filter.

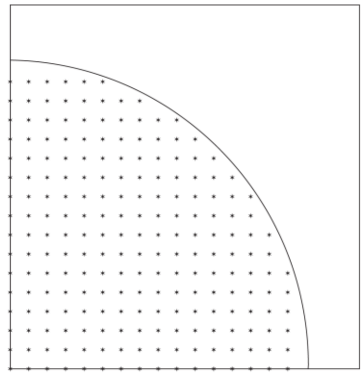


Figure 9.5.5 Possible pole positions for the coupled-form two-pole filter in Fig. 9.5.4.

33

Summary of filter structures

- All filter structures give identical output in infinite precision
- Advantages and disadvantages show up in finite precision
  – Other factors include computational complexity, and storage requirements

34

Summary

- Today we discussed:
  - Number representations
  - Rounding errors and limit cycles
- Next:
  - Multirate processing