Analysis of Finite Wordlength Effects

- Ideally, the system parameters along with the signal variables have infinite precision taking any value between $-\infty$ and ∞
- In practice, they can take only discrete values within a specified range since the registers of the digital machine where they are stored are of finite length
- The discretization process results in nonlinear difference equations characterizing the discrete-time systems

1

Analysis of Finite Wordlength Effects

- These nonlinear equations, in principle, are almost impossible to analyze and deal with exactly
- However, if the quantization amounts are small compared to the values of signal variables and filter parameters, a simpler approximate theory based on a statistical model can be applied

2

Analysis of Finite Wordlength Effects

- Using the statistical model, it is possible to derive the effects of discretization and develop results that can be verified experimentally
- Sources of errors -

  (1) Filter coefficient quantization

  (2) A/D conversion

  (3) Quantization of arithmetic operations

  (4) Limit cycles

3

Analysis of Finite Wordlength Effects

- Consider the first-order IIR digital filter

$$y[n] = \alpha\, y[n-1] + x[n]$$

where $y[n]$ is the output signal and $x[n]$ is the input signal

- When implemented on a digital machine, the filter coefficient α can assume only certain discrete values $\hat{\alpha}$ approximating the original design value α

4

Analysis of Finite Wordlength Effects

- The desired transfer function is

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

- The actual transfer function implemented is

$$\hat{H}(z) = \frac{z}{z-\hat{\alpha}}$$

which may be much different from the desired transfer function $H(z)$

5

Analysis of Finite Wordlength Effects

- Thus, the actual frequency response may be quite different from the desired frequency response
- Coefficient quantization problem is similar to the sensitivity problem encountered in analog filter implementation

6

Analysis of Finite Wordlength Effects

- A/D Conversion Error - generated by the filter input quantization process
- If the input sequence $x[n]$ has been obtained by sampling an analog signal $x_a(t)$, then the actual input to the digital filter is

$$\hat{x}[n] = x[n] + e[n]$$

where $e[n]$ is the **A/D conversion error**

7

Analysis of Finite Wordlength Effects

- Arithmetic Quantization Error - For the first-order digital filter, the desired output of the multiplier is

$$v[n] = \alpha y[n-1]$$

- Due to product quantization, the actual output of the multiplier of the implemented filter is

$$\hat{v}[n] = \alpha y[n-1] + e_\alpha[n] = v[n] + e_\alpha[n]$$

where $e_\alpha[n]$ is the **product roundoff error**

8

Analysis of Finite Wordlength Effects

- Limit Cycles - The nonlinearity of the arithmetic quantization process may manifest in the form of oscillations at the filter output, usually in the absence of input or, sometimes, in the presence of constant input signals or sinusoidal input signals

9

Quantization Process and Errors

- Two basic types of binary representations of data: (1) Fixed-point, and (2) Floating-point formats
- Various problems can arise in the digital implementation of the arithmetic operations involving the binary data
- Caused by the finite wordlength limitations of the registers storing the data and the results of arithmetic operations

10

Quantization Process and Errors

- For example in fixed-point arithmetic, product of two b-bit numbers is $2b$ bits long, which has to be quantized to b bits to fit the prescribed wordlength of the registers
- In fixed-point arithmetic, addition operation can result in a sum exeeding the register wordlength, causing an overflow
- In floating-point arithmetic, there is no overflow, but results of both addition and multiplication may have to be quantized

11

Quantization Process and Errors

- In both fixed-point and floating-point formats, a negative number can be represented in one of three different forms
- Analysis of various quantization effects on the performance of a digital filter depends on
  (1) Data format (fixed-point or floating-point),
  (2) Type of representation of negative numbers,
  (3) Type of quantization, and
  (4) Digital filter structure implementing the transfer function

12

Quantization Process and Errors

- Since the number of all possible combinations of the type of arithmetic, type of quantization method, and digital filter structure is very large, quantization effects in some selected practical cases are discussed
- Analysis presented can be extended easily to other cases

13

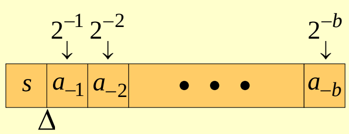
Quantization Process and Errors

- In DSP applications, it is a common practice to represent the data either as a fixed-point fraction or as a floating-point binary number with the mantissa as a binary fraction
- Assume the available wordlength is $(b+1)$ bits with the **most significant bit** (MSB) representing the sign
- Consider the data to be a $(b+1)$-bit fixed-point fraction

14

Quantization Process and Errors

- Representation of a general $(b+1)$-bit fixed-point fraction is shown below



- Smallest positive number that can be represented in this format will have a **least significant bit** (LSB) of 1 with remaining bits being all 0's

15

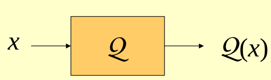
Quantization Process and Errors

- Decimal equivalent of smallest positive number is $\delta = 2^{-b}$
- Numbers represented with $(b+1)$ bits are thus quantized in steps of 2^{-b}, called **quantization step**
- An original data x represented as a $(\beta+1)$-bit fraction is converted into a $(b+1)$-bit fraction $Q(x)$ either by **truncation** or **rounding**

16

Quantization Process and Errors

- The quantization process for truncation or rounding can be modeled as shown below



17

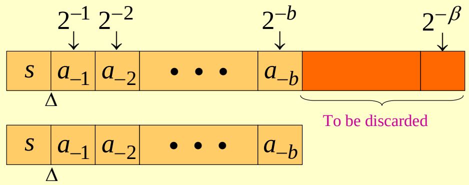
Quantization Process and Errors

- Since representation of a positive binary fraction is the same independent of format being used to represent the negative binary fraction, effect of quantization of a positive fraction remains unchanged
- The effect of quantization on negative fractions is different for the three different representations

18

Quantization of Fixed-Point Numbers

- **Truncation** of a (β+1)-bit fixed-point number to (b+1) bits is achieved by simply discarding the least significant $(\beta - b)$ bits as shown below



Quantization of Fixed-Point Numbers

- Range of **truncation error** $\varepsilon_t = Q(x) - x$ (assuming β >> b):
- Positive number and two's complement negative number

$$-\delta < \varepsilon_t \leq 0$$

- Sign-magnitude negative number and ones'-complement negative number

$$0 \leq \varepsilon_t < \delta$$

Quantization of Fixed-Point Numbers

- Range of **rounding error** $\varepsilon_r = Q(x) - x$ (assuming β >> b):
- For all positive and negative numbers

$$-\frac{\delta}{2} < \varepsilon_r \leq \frac{\delta}{2}$$

Quantization of Floating-Point Numbers

- In floating-point format a decimal number x is represented as $x = 2^E \cdot M$ where E is the **exponent** and M is the **mantissa**
- Mantissa M is a binary fraction restricted to lie in the range

$$\tfrac{1}{2} \leq M < 1$$

- Exponent E is either a positive or a negative binary number

Quantization of Floating-Point Numbers

- The quantization of a floating-point number is carried out only on the mantissa
- Range of **relative error**:

$$\varepsilon = \frac{Q(x) - x}{x} = \frac{Q(M) - M}{M}$$

- Two's complement truncation

$$-2\delta < \varepsilon_t \leq 0, \quad x > 0$$
$$0 \leq \varepsilon_t < 2\delta, \quad x < 0$$

Quantization of Floating-Point Numbers

- Sign-magnitude and ones's complement truncation

$$-2\delta < \varepsilon_t \leq 0$$

- Rounding of all numbers

$$-\delta < \varepsilon_r \leq \delta$$

- Note: We consider in this course fixed-point implementation case

Analysis of Coefficient Quantization Effects

- The transfer function $\hat{H}(z)$ of the digital filter implemented with quantized coefficients is different from the desired transfer function $H(z)$
- Main effect of coefficient quantization is to move the poles and zeros to different locations from the original desired locations

Analysis of Coefficient Quantization Effects

- The actual frequency response $\hat{H}(e^{j\omega})$ is thus different from the desired frequency response $H(e^{j\omega})$
- In some cases, the poles may move outside the unit circle causing the implemented digital filter to become unstable even though the original transfer function $H(z)$ is stable

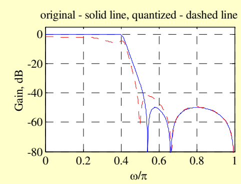
Analysis of Coefficient Quantization Effects

- Effect of coefficient quantization can be easily carried out using MATLAB
- To this end, the M-files `a2dT` (for truncation) and `a2dR` (for rounding) can be used

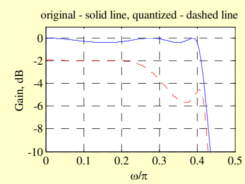
Coefficient Quantization Effects On a Direct Form IIR Filter

- Gain responses of a 5-th order elliptic lowpass filter with unquantized and quantized coefficients

Fullband Gain Response

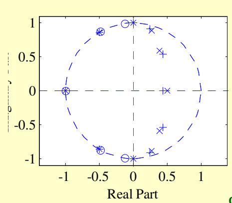


Passband Details



Coefficient Quantization Effects On a Direct Form IIR Filter

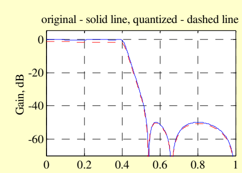
- Pole and zero locations of the filter with quantized coefficients (denoted by "x" and "o") and those of the filter with unquantized coefficients (denoted by "+" and "*")



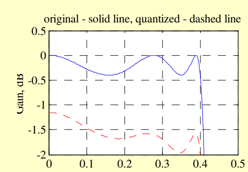
Coefficient Quantization Effects On a Cascade Form IIR Filter

- Gain responses of a 5-th order elliptic lowpass filter implemented in a cascade form with unquantized and quantized coefficients

Fullband Gain Response

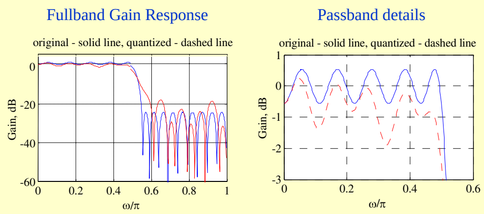


Passband Details



Coefficient Quantization Effects On A Direct Form FIR Filter

- Gain responses of a 39-th order equiripple lowpass FIR filter with unquantized and quantized coefficients



Estimation of Pole-Zero Displacements

- Consider an N-th degree polynomial $B(z)$ with simple roots:

$$B(z) = \sum_{i=0}^{N} b_i z^i = \prod_{k=1}^{N} (z - z_k)$$

with $b_N = 1$

- Roots z_k of $B(z)$ are given by

$$z_k = r_k e^{j\theta_k}$$

Estimation of Pole-Zero Displacements

- Effect of coefficient quantization is to change the polynomial coefficient b_i to $b_i + \Delta b_i$
- Thus, the polynomial $B(z)$ after coefficient quantization becomes

$$\hat{B}(z) = \sum_{i=0}^{N} (b_i + \Delta b_i) z^i$$
$$= B(z) + \sum_{i=0}^{N} (\Delta b_i) z^i = \prod_{k=1}^{N} (z - \hat{z}_k)$$

Estimation of Pole-Zero Displacements

- $\hat{z}_k$ denotes the roots of $\hat{B}(z)$ and are the new locations to which roots z_k of $B(z)$ have moved
- For small changes in the coefficient values, $\hat{z}_k$ will be close to z_k and can be expressed as

$$\hat{z}_k = z_k + \Delta z_k = (r_k + \Delta r_k) e^{j(\theta_k + \Delta\theta_k)}$$

Estimation of Pole-Zero Displacements

- If Δb_i is assumed to be very small, we can express

$$\hat{z}_k = (r_k + \Delta r_k) e^{j\theta_k} e^{j\Delta\theta_k} \cong (r_k + \Delta r_k)(1 + j\Delta\theta_k) e^{j\theta_k}$$
$$\cong r_k e^{j\theta_k} + (\Delta r_k + j r_k \Delta\theta_k) e^{j\theta_k}$$

neglecting higher order terms

- Then

$$\Delta z_k = \hat{z}_k - z_k \cong (\Delta r_k + j r_k \Delta\theta_k) e^{j\theta_k}$$

Estimation of Pole-Zero Displacements

- Now we can express $1/B(z)$ by partial-fraction expansion as

$$\frac{1}{B(z)} = \sum_{k=1}^{N} \frac{\rho_k}{z - z_k}$$

where ρ_k is the residue of $1/B(z)$ at the pole $z = z_k$, i.e.,

$$\rho_k = \left.\frac{(z - z_k)}{B(z)}\right|_{z = z_k} = R_k + jX_k$$

Estimation of Pole-Zero Displacements

- If $\hat{z}_k$ is very close to z_k, then we can write

$$\frac{1}{B(\hat{z}_k)} \cong \frac{\rho_k}{\hat{z}_k - z_k}$$

or

$$\Delta z_k = \rho_k \cdot B(\hat{z}_k)$$

- But

$$\hat{B}(\hat{z}_k) = 0 = B(\hat{z}_k) + \sum_{i=0}^{N-1} (\Delta b_i)(\hat{z}_k)^i$$

37

Estimation of Pole-Zero Displacements

- Therefore

$$\Delta z_k = -\rho_k \left\{ \sum_{i=0}^{N-1} (\Delta b_i)(\hat{z}_k)^i \right\} \cong -\rho_k \left\{ \sum_{i=0}^{N-1} (\Delta b_i)(z_k)^i \right\}$$

assuming that $\hat{z}_k$ is very close to z_k

- Rewriting the above equation we get

$$(\Delta r_k + j r_k \Delta\theta_k) e^{j\theta_k} = -(R_k + jX_k) \left\{ \sum_{i=0}^{N-1} (\Delta b_i)(z_k)^i \right\}$$

38

Estimation of Pole-Zero Displacements

- Equating real and imaginary parts of the above we arrive at

$$\Delta r_k = (-R_k \mathbf{P}_k + X_k \mathbf{Q}_k) \cdot \Delta\mathbf{B} = \mathbf{S}_b^{r_k} \cdot \Delta\mathbf{B}$$

$$\Delta\theta_k = -\frac{1}{r_k}(X_k \mathbf{P}_k + R_k \mathbf{Q}_k) \cdot \Delta\mathbf{B} = \mathbf{S}_b^{\theta_k} \cdot \Delta\mathbf{B}$$

where

$$\mathbf{P}_k = [\cos\theta_k \quad r_k \quad r_k^2 \cos\theta_k \quad \cdots \quad r_k^{N-1} \cos(N-2)\theta_k]$$

$$\mathbf{Q}_k = [-\sin\theta_k \quad 0 \quad r_k^2 \sin\theta_k \quad \cdots \quad r_k^{N-1} \sin(N-2)\theta_k]$$

$$\Delta\mathbf{B} = [\Delta b_0 \quad \Delta b_1 \quad \Delta b_2 \quad \cdots \quad \Delta b_{N-1}]^T$$

39

Estimation of Pole-Zero Displacements

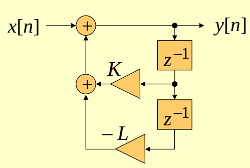
- The sensitivity vectors $\mathbf{S}_b^{r_k}$ and $\mathbf{S}_b^{\theta_k}$ depend only on $B(z)$ and are independent of $\Delta\mathbf{B}$
- Once these vectors have been calculated, pole-zero displacements for any sets of $\Delta\mathbf{B}$ can be calculated using the equations given
- Elements of $\Delta\mathbf{B}$ are multiplier coefficient changes only for the direct form realizations

40

Estimation of Pole-Zero Displacements

- Example - Consider the direct form II realization of

$$H(z) = \frac{z^2}{z^2 - Kz + L} = \frac{z^2}{B(z)}$$



41

Estimation of Pole-Zero Displacements

- Here $B(z) = z^2 - Kz + L = (z - z_1)(z - z_2)$

where $z_1 = re^{j\theta}$, $z_2 = re^{-j\theta}$

- We compute

$$\rho_1 = \left.\frac{z - z_1}{B(z)}\right|_{z=z_1} = -\frac{j}{2r\sin\theta}$$

- Therefore

$$\Delta\mathbf{B} = [\Delta L \quad -\Delta K]^T$$

$$\mathbf{Q}_1 = [-\sin\theta \quad 0]$$

$$\mathbf{P}_1 = [\cos\theta \quad r]$$

42

Estimation of Pole-Zero Displacements

- Substituting these values we get

$$\Delta r = X_1 \mathbf{Q}_1 \Delta \mathbf{B} = \frac{1}{2r}\Delta L$$

$$\Delta\theta = -\frac{1}{r}(X_1 \mathbf{P}_1 \Delta \mathbf{B}) = \frac{\Delta L}{2r^2\tan\theta} - \frac{\Delta K}{2r\sin\theta}$$

- It can be seen that the 2nd-order direct form IIR structure is highly sensitive to coefficient quantizations for transfer functions with poles close to θ = 0 or π

43

Estimation of Pole-Zero Displacements

- Consider an arbitrary digital filter structure with R multipliers given by

$$\alpha_k,\ k = 1, 2, \ldots, R$$

- The multiplier coefficients α_k are multilinear functions of the coefficients b_i of the polynomial $B(z)$

44

Estimation of Pole-Zero Displacements

- Thus, when α_k changes into $\alpha_k + \Delta\alpha_k$ due to coefficient quantization, the change Δb_i in the polynomial coefficient b_i can be expressed as

$$\Delta b_i = \sum_{k=1}^{R} \frac{\partial b_i}{\partial \alpha_k}\Delta\alpha_k, \quad i = 1, 2, \ldots, N-1$$

45

Estimation of Pole-Zero Displacements

- In matrix form we have $\Delta\mathbf{B} = \mathbf{C}\cdot\Delta\boldsymbol{\alpha}$

where

$$\mathbf{C} = \begin{bmatrix} \frac{\partial b_0}{\partial \alpha_1} & \frac{\partial b_0}{\partial \alpha_2} & \cdots & \frac{\partial b_0}{\partial \alpha_R} \\ \frac{\partial b_1}{\partial \alpha_1} & \frac{\partial b_1}{\partial \alpha_2} & \cdots & \frac{\partial b_1}{\partial \alpha_R} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial b_{N-1}}{\partial \alpha_1} & \frac{\partial b_{N-1}}{\partial \alpha_2} & \cdots & \frac{\partial b_{N-1}}{\partial \alpha_R} \end{bmatrix}$$

$$\Delta\boldsymbol{\alpha} = [\Delta\alpha_1 \quad \Delta\alpha_2 \quad \Delta\alpha_3 \quad \cdots \quad \Delta\alpha_R]^{\mathrm{T}}$$

46

Estimation of Pole-Zero Displacements

- Here the root displacements are given by

$$\Delta r_k = \mathbf{S}_b^{r_k}\cdot\mathbf{C}\cdot\Delta\boldsymbol{\alpha}$$

$$\Delta\theta_k = \mathbf{S}_b^{\theta_k}\cdot\mathbf{C}\cdot\Delta\boldsymbol{\alpha}$$

where the sensitivity vectors $\mathbf{S}_b^{r_k}$ and $\mathbf{S}_b^{\theta_k}$ are as given earlier

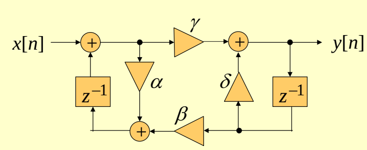
- Note: The matrix **C** depends on the structure but has to be computed only once

47

Estimation of Pole-Zero Displacements

- Example - Consider the coupled-form structure with a transfer function given by

$$H(z) = \frac{\gamma z^2}{z^2 - (\alpha+\delta)z + (\alpha\delta - \beta\gamma)}$$



48

Estimation of Pole-Zero Displacements

- If $\alpha = \delta = r\cos\theta$ and $\beta = -\gamma = r\sin\theta$, then the transfer function becomes

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

- Comparing the denominator of the above with that of the transfer function of the direct form structure we get

$$K = \alpha + \delta = 2\alpha$$
$$L = \alpha\delta - \beta\gamma = \alpha^2 + \beta^2$$

49

Estimation of Pole-Zero Displacements

- Taking the partials of both sides of the last two equations we get

$$\begin{bmatrix} \Delta L \\ \Delta K \end{bmatrix} = \begin{bmatrix} 2r\cos\theta & 2r\sin\theta \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix}$$

- Finally, substituting the results of the previous example we arrive at

$$\begin{bmatrix} \Delta r \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2r} & 0 \\ \frac{1}{2r^2\tan\theta} & -\frac{1}{2r\sin\theta} \end{bmatrix} \begin{bmatrix} 2r\cos\theta & 2r\sin\theta \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix}$$

50

Estimation of Pole-Zero Displacements

- or,

$$\begin{bmatrix} \Delta r \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{r}\sin\theta & \frac{1}{r}\cos\theta \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix}$$

- As can be seen from the above, the coupled-form structure is less sensitive to multiplier coefficient quantization than the direct form structure

51

A/D Conversion Noise Analysis

- A/D converters used for digital processing of analog signals in general employ two's-complement fixed-point representation to represent the digital equivalent of the input analog signal
- For the processing of bipolar analog signals, the A/D converter generates a bipolar output represented as a fixed-point signed binary fraction

52

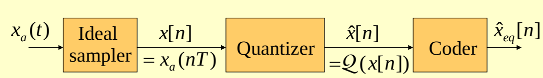
Quantization Noise Model

- The digital sample generated by the A/D converter is the binary representation of the quantized version of that produced by an ideal sampler with infinite precision
- If the output word is of length (b+1) bits including the sign bit, the total number of discrete levels available for the representation of the digital equivalent is 2^{b+1}

53

Quantization Noise Model

- The dynamic range of the output register depends on the binary number representation selected for the A/D converter
- The model of a practical A/D conversion system is as shown below



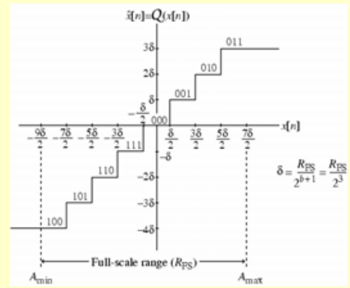
54

Quantization Noise Model

- The quantization process employed by the quantizer can be either rounding or truncation
- Assuming rounding is used, the input-output characteristic of a 3-bit A/D converter with the output in two's-complement form is as shown next

Quantization Noise Model

- Input-output characteristic



Quantization Noise Model

- The binary equivalent $\hat{x}_{eq}[n]$ of the quantized input analog sample $\hat{x}[n]$ for a two's-complement binary representation, is a binary fraction in the range
$$-1 \le \hat{x}_{eq}[n] < 1$$
- It is related to the quantized sample $\hat{x}[n]$ through
$$\hat{x}_{eq}[n] = \frac{2\hat{x}[n]}{R_{FS}}$$
where R_{FS} denotes the **full-scale range** of the A/D converter

Quantization Noise Model

- Assume the input signal has been scaled to be in the range of ± 1 by dividing its amplitude by $R_{FS}/2$, as is usually the case
- The decimal equivalent of $\hat{x}_{eq}[n]$ is then equal to $\hat{x}[n]$
- For a (b+1)-bit bipolar A/D converter, the total number of quantization levels is 2^{b+1}
- The full-scale range is $R_{FS} = 2^{b+1}\delta$
where δ is the quantization step size

Quantization Noise Model

- For the 3-bit bipolar A/D converter, total number of levels is $2^3 = 8$
- The full-scale range is $R_{FS} = 8\delta$ with a maximum value of $A_{max} = 7\delta/2$ and a minimum value of $A_{min} = -9\delta/2$
- If the input analog sample $x_a(nT)$ is within the full-scale range
$$-\frac{9\delta}{2} < x_a(nT) \le \frac{7\delta}{2}$$
it is quantized to one of the 8 discrete levels shown earlier

Quantization Noise Model

- In general, for a (b+1)-bit bipolar A/D converter employing two's-complement representation, the full-scale range is given by
$$-(2^{b+1}+1)\frac{\delta}{2} < x_a(nT) \le (2^{b+1}-1)\frac{\delta}{2}$$
- Denote the difference between the quantized value $Q(x[n]) = \hat{x}[n]$ and the input sample $x[n]$ as the quantization error:
$$e[n] = Q(x[n]) - x[n] = \hat{x}[n] - x[n]$$

Quantization Noise Model

- It follows from the input-output characteristic of the 3-bit bipolar A/D converter given earlier that $e[n]$ is in the range

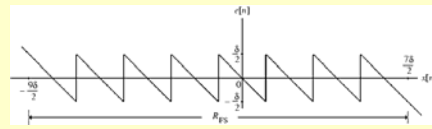
$$-\frac{\delta}{2} < e[n] \leq \frac{\delta}{2}$$

assuming that a sample exactly halfway between two levels is rounded up to the nearest higher level and assuming that the analog input is within the A/D converter full-scale range

61

Quantization Noise Model

- In this case, the quantization error $e[n]$, called the granular noise, is bounded in magnitude according to $-\frac{\delta}{2} < e[n] \leq \frac{\delta}{2}$
- A plot of the $e[n]$ of the 3-bit A/D converter as a function of the input sample $x[n]$ is shown below



62

Quantization Noise Model

- When the input analog sample is outside the full-scale range of the A/D converter, the magnitude of error $e[n]$ increases linearly with an increase in the magnitude of the input
- In such a situation, the error $e[n]$ is called the **saturation noise** or the **overload noise** as the A/D converter output is "clipped" to the maximum value $1-2^{-b}$ if the analog input is positive or to the minimum value -1 if the analog input is negative

63

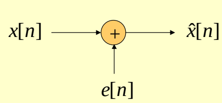
Quantization Noise Model

- A clipping of the A/D converter output causes signal distortion with highly undesirable effects and must be avoided by scaling down the input analog signal $x_a(nT)$ to ensure that it remains within the A/D converter full-scale range
- We therefore assume that input analog samples are within the A/D converter full-scale range and thus, there is no saturation error

64

Quantization Noise Model

- Now, the input-output characteristic of an A/D converter is nonlinear, and the analog input signal is not known a priori in most cases
- It is thus reasonable to assume for analysis purposes that the error $e[n]$ is a random signal with a statistical model as shown below



65

Quantization Noise Model

- For simplified analysis, the following assumptions are made:

  (1) The error sequence $\{e[n]\}$ is a sample sequence of a wide-sense stationary (WSS) white noise process, with each sample $e[n]$ being uniformly distributed over the range of the quantization error

  (2) The error sequence is uncorrelated with its corresponding input sequence $\{x[n]\}$

  (3) The input sequence is a sample sequence of a stationary random process

66

Quantization Noise Model

- These assumptions hold in most practical situations for input signals whose samples are large and change in amplitude very rapidly in time relative to the quantization step in a somewhat random fashion
- These assumptions have also been verified experimentally and by computer simulations

67

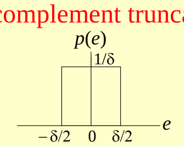
Quantization Noise Model

- The statistical model makes the analysis of A/D conversion noise more tractable and results derived have been found to be useful for most applications
- If ones'-complement or sign-magnitude truncation is employed, the quantization error is correlated to the input signal as the sign of each error sample $e[n]$ is exactly opposite to the sign of the corresponding input sample $x[n]$

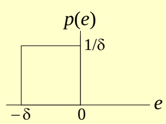
68

Quantization Noise Model

- As a result, practical A/D converters use either rounding or two's-complement truncation
- Quantization error probability density functions $p(e)$ for rounding and two's-complement truncation are as shown below



Rounding



Two's-complement truncation

69

Quantization Noise Model

- Mean and variance of the error sample $e[n]$:
- Rounding -

$$m_e = \frac{(\delta/2) - (\delta/2)}{2} = 0$$

$$\sigma_e^2 = \frac{((\delta/2) - (-\delta/2))^2}{12} = \frac{\delta^2}{12}$$

- Two's-complement truncation -

$$m_e = \frac{0-\delta}{2} = -\frac{\delta}{2}$$

$$\sigma_e^2 = \frac{(0-\delta)^2}{12} = \frac{\delta^2}{12}$$

70

Signal-to-Quantization Noise Ratio

- The effect of the additive quantization noise $e[n]$ on the input signal $x[n]$ is given by the **signal-to-quantization noise ratio** given by

$$SNR_{A/D} = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) dB$$

where σ_x^2 is the input signal variance representing the **signal power** and σ_e^2 is the noise variance representing the **quantization noise power**

71

Signal-to-Quantization Noise Ratio

- For rounding, $e[n]$ is uniformly distributed in the range $(-\delta/2, \delta/2)$
- For two's-complement truncation, $e[n]$ is uniformly distributed in the range $(-\delta, 0)$
- For a bipolar $(b+1)$-bit A/D converter

$$\delta = 2^{-(b+1)} R_{FS}$$

- Hence

$$\sigma_e^2 = \frac{2^{-2b}(R_{FS})^2}{48}$$

72

Signal-to-Quantization Noise Ratio

- Therefore $SNR_{A/D} = 10\log_{10}\left(\frac{48\sigma_x^2}{2^{-2b}(R_{FS})^2}\right)$

$$= 6.02b + 16.81 - 20\log\left(\frac{R_{FS}}{\sigma_x}\right) \; dB$$

- This expression can be used to determine the minimum wordlength of an A/D converter needed to meet a specified $SNR_{A/D}$
- Note: $SNR_{A/D}$ increases by 6 dB for each bit added to the wordlength

73

Signal-to-Quantization Noise Ratio

- For a given wordlength, the actual SNR depends on σ_x, the rms value of the input signal amplitude and the full-scale range R_{FS} of the A/D converter
- Example - Determine the SNR in the digital equivalent of an analog sample $x[n]$ with a zero-mean Gaussian distribution using a $(b+1)$-bit A/D converter having $R_{FS} = K\sigma_x$

74

Signal-to-Quantization Noise Ratio

- Here

$$SNR_{A/D} = 6.02b + 16.81 - 20\log_{10}\left(\frac{R_{FS}}{\sigma_x}\right)$$
$$= 6.02b + 16.81 - 20\log_{10}(K)$$

- Computed values of the SNR for various values of K are as given below:

	$b=7$	$b=9$	$b=11$	$b=13$	$b=15$
$K=4$	46.91	58.95	70.99	83.04	95.08
$K=6$	43.39	55.43	67.47	79.51	91.56
$K=8$	40.89	52.93	64.97	77.01	89.05

75

Signal-to-Quantization Noise Ratio

- The probability of a particular input analog sample with a zero-mean Gaussian distribution staying within the full-scale range $2K\sigma_x$ is given by

$$2\Phi(k) - 1 = \sqrt{\frac{2}{\pi}} \int_0^K e^{-y^2/2} dy$$

76

Signal-to-Quantization Noise Ratio

- Thus, for $K = 4$, the probability of an analog sample staying within the full-scale range $8\sigma_x$ is 0.9544

On average about 456 samples out of 10,000 samples will fall outside the full-scale range and be clipped

77

Signal-to-Quantization Noise Ratio

- For $K = 6$, the probability of an analog sample staying within the full-scale range $12\sigma_x$ is 0.9974

On average about 26 samples out of 10,000 samples will fall outside the full-scale range and be clipped

- In most applications, a full-scale range of $16\sigma_x$ is more than adequate to ensure no clipping in conversion

78

Effect of Input Scaling on SNR

- Consider the scaled input $Ax[n]$
- The variance of the scaled input is $A^2\sigma_x^2$
- Then

$$SNR_{A/D} = 6.02b + 16.81 - 20\log_{10}(K) + 20\log_{10}(A)$$

- For a given b, the SNR can be increased by scaling up the input signal by making $A > 1$

79

Effect of Input Scaling on SNR

- But increasing A also increases the probability that some of the input analog samples being outside the full-scale range R_{FS} and as result, the expression for $SNR_{A/D}$ no longer holds
- Moreover, the output is clipped, causing severe distortion in the digital representation of the input analog signal

80

Effect of Input Scaling on SNR

- A scaling down of the input analog signal by choosing $A < 1$ decreases the SNR
- It is therefore necessary to ensure that the input analog sample range matches as close as possible to the full-scale range of the A/D converter to get the maximum possible SNR without any signal distortion

81

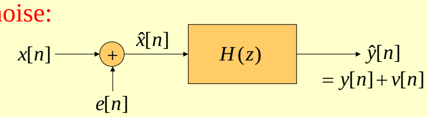
Propagation of Input Quantization Noise to Digital Filter Output

- To determine the propagation of input quantization noise to the digital filter output, we assume that the digital filter is implemented using infinite precision
- In practice, the quantization of arithmetic operations generates errors inside the digital filter structure, which also propagate to the output and appear as noise

82

Propagation of Input Quantization Noise to Digital Filter Output

- The internal noise sources are assumed to be independent of the input quantization noise and their effects can be analyzed separately and added to that due to the input noise
- Model for the analysis of input quantization noise:



83

Propagation of Input Quantization Noise to Digital Filter Output

- Because of the linearity property of the digital filter and the assumption that $x[n]$ and $e[n]$ are uncorrelated, the output $\hat{y}[n]$ of the LTI system can thus expressed as

$$\hat{y}[n] = y[n] + v[n]$$

where $y[n]$ is the output generated by the unquantized input $x[n]$ and $v[n]$ is the output generated by the error sequence $e[n]$

84

Propagation of Input Quantization Noise to Digital Filter Output

- Therefore

$$v[n] = h[n] \circledast e[n] = \sum_{m=-\infty}^{\infty} e[m]h[n-m]$$

- The mean m_v of the output noise $v[n]$ is given by

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

and its variance σ_v^2 is given by

$$\sigma_v^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

85

Propagation of Input Quantization Noise to Digital Filter Output

- The output noise power spectrum is given by

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

- The normalized output noise variance is given by

$$\sigma_{v,n}^2 = \frac{\sigma_v^2}{\sigma_e^2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

86

Propagation of Input Quantization Noise to Digital Filter Output

- Alternately,

$$\sigma_{v,n}^2 = \frac{1}{2\pi j} \oint_C H(z)H(z^{-1})z^{-1}dz$$

where C is a counterclockwise contour in the ROC of $H(z)H(z^{-1})$

- An equivalent expression for the normalized output noise variance is

$$\sigma_{v,n}^2 = \sum_{n=-\infty}^{\infty} \left| h[n] \right|^2$$

87

Algebraic Computation of Output Noise Variance

- In general, $H(z)$ is a causal stable real rational function with all poles inside the unit circle in the z-plane
- It can be expressed in a partial-fraction expansion form

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

where $H_i(z)$ is a low-order causal stable real rational transfer function

88

Algebraic Computation of Output Noise Variance

- Substituting the partial-fraction expansion of $H(z)$ in

$$\sigma_{v,n}^2 = \frac{1}{2\pi j} \oint_C H(z)H(z^{-1})z^{-1}dz$$

we arrive at

$$\sigma_{v,n}^2 = \frac{1}{2\pi j} \sum_{k=1}^{R} \sum_{\ell=1}^{R} \oint_C H_k(z)H_\ell(z^{-1})z^{-1}dz$$

89

Algebraic Computation of Output Noise Variance

- Since $H_k(z)$ and $H_\ell(z)$ are stable transfer functions, it can be shown that

$$\oint_C H_k(z)H_\ell(z^{-1})z^{-1}dz = \oint_C H_\ell(z)H_k(z^{-1})z^{-1}dz$$

- Thus, we can write

$$\sigma_{v,n}^2 = \frac{1}{2\pi j} \sum_{k=1}^{R} \oint_C H_k(z)H_k(z^{-1})z^{-1}dz + \frac{2}{2\pi j} \sum_{k=1}^{R-1} \sum_{\ell=k+1}^{R} \oint_C H_k(z)H_\ell(z^{-1})z^{-1}dz$$

90

Algebraic Computation of Output Noise Variance

- In most practical cases, $H(z)$ has only simple poles with $H_k(z)$ being either a 1st-order or a 2nd-order transfer function
- Typical terms in the partial-fraction expansion of $H(z)$ are:

$$A, \quad \frac{B_k}{z - a_k}, \quad \frac{C_k z + D_k}{z^2 + b_k z + d_k}$$

- Let a typical contour integral be denoted as

$$I_i = \frac{1}{2\pi j} \oint_C H_k(z) H_\ell(z^{-1}) z^{-1} dz$$

91

Table of Typical Contour Integrals

$H_k(z)$	$H_\ell(z^{-1})$: A	$H_\ell(z^{-1})$: $\frac{B_\ell}{z^{-1} - a_\ell}$	$H_\ell(z^{-1})$: $\frac{C_\ell z^{-1} + D_\ell}{z^{-2} + b_\ell z^{-1} + d_\ell}$
A	I_1	0	0
$\frac{B_k}{z - a_k}$	0	I_2	I'_4
$\frac{C_k z + D_k}{z^2 + b_k z + d_k}$	0	I_4	I_3

92

Table of Typical Contour Integrals

- where

$$I_1 = A^2$$

$$I_2 = \frac{B_k B_\ell}{1 - a_k a_\ell}$$

$$I_3 = \frac{(C_k C_\ell + D_k D_\ell)(1 - d_k d_\ell) - (C_\ell D_k - D_\ell C_k d_k) b_\ell - (C_k D_\ell - D_k C_\ell d_\ell) b_k}{(1 - d_k d_\ell)^2 + d_k b_\ell^2 + d_\ell b_k^2 - (1 + d_k d_\ell) b_k b_\ell}$$

$$I_4 = \frac{B_\ell (C_k + D_k a_\ell)}{1 + b_k a_\ell + d_k a_\ell^2}$$

$$I'_4 = \frac{B_k (C_\ell + D_\ell a_k)}{1 + b_\ell a_k + d_\ell a_k^2}$$

93

Algebraic Computation of Output Noise Variance

- Example - Consider a first-order digital filter with a transfer function

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

- A partial-fraction expansion of $H(z)$ is

$$H(z) = 1 + \frac{\alpha}{z - \alpha}$$

- The two terms in the above expansion are

$$H_1(z) = 1, \quad H_2(z) = \frac{\alpha}{z - \alpha}$$

94

Algebraic Computation of Output Noise Variance

- Therefore, the normalized output noise variance is given by

$$\sigma_{v,n}^2 = 1 + \frac{\alpha^2}{1 - \alpha^2} = \frac{1}{1 - \alpha^2}$$

- If the pole is close to the unit circle, we can write $|\alpha| = 1 - \varepsilon$, where $\varepsilon \cong 0$
- In which case

$$\sigma_{v,n}^2 = \frac{1}{1 - (1 - \varepsilon)^2} \cong \frac{1}{2\varepsilon}$$

95

Algebraic Computation of Output Noise Variance

- Thus, as the pole gets closer to the unit circle, the output noise increases rapidly to very high values approaching infinity
- For high-Q realizations, the wordlengths of the registers storing the signal variables should be of longer length to keep the round-off noise below a prescribed level

96

Computation of Output Noise Variance Using MATLAB

- In the MATLAB implementation of the algebraic method outlined earlier, the partial-fraction expansion can be carried out using the M-file `residue`
- This results in terms of the form A and $B_k/(z-a_k)$ where the residues B_k and the poles a_k are either real or complex numbers
- For variance calculation, only the terms I_1 and I_2 are then employed

97

Computation of Output Noise Variance Using MATLAB

- An alternative fairly simple method of computation is based on the output noise variance formula

$$\sigma_{v,n}^2 = \sum_{n=-\infty}^{\infty} |h[n]|^2$$

- For a causal stable digital filter, the impulse response decays rapidly to zero values
- Hence we can write

$$\sigma_{v,n}^2 = S_L \cong \sum_{n=0}^{L} |h[n]|^2$$

98

Computation of Output Noise Variance Using MATLAB

- To determine an approximate value of $\sigma_{v,n}^2$ the sum S_L is computed for $L = 1, 2, \ldots,$ and the computation is stopped when

$$S_L - S_{L-1} < \kappa$$

where κ is a specified small number, which is typically chosen as 10^{-6}

99