Finite Word Length Effects 5

Learning Objectives

After completing this chapter, you should be able to:

- provide an understanding of the errors that arise in practical DSP systems due to quantization and use of finite word length arithmetic.
- study the effects of errors on signal quality.
- develop the techniques to combat the errors.
- enhance the skill in the design of DSP systems.

5.1 Introduction

When digital systems are implemented either in hardware or in software, the filter coefficients are stored in binary registers. These registers can accommodate only a finite number of bits and hence, the filter coefficients have to be truncated or rounded-off in order to fit into these registers. Truncation or rounding of the data results in degradation of system performance. Also, in digital processing systems, a continuous-time input signal is sampled and quantized in order to get the digital signal. The process of quantization introduces an error in the signal which is called round off noise. This makes the system non-linear, and leads to limit cycle behaviour.

In general the effects due to finite precision representation of numbers in a digital system are commonly referred to as finite word length effects. Some of the finite word length effects in digital filters are:

1. Errors due to quantization of input data by A/D converter.
2. Errors due to quantization of filter coefficients.
3. Errors due to rounding the product in multiplication.
4. Errors due to overflow in addition.
5. Limit cycles.

The effects of these errors introduced by signal processing depend on a number of factors which include the type of arithmetic used the quality of the input signal and the DSP algorithm implemented. These are discussed in this chapter.

Finite Word Length Effects 5

Learning Objectives

After completing this chapter, you should be able to:

- provide an understanding of the errors that arise in practical DSP systems due to quantization and use of finite word length arithmetic.
- study the effects of errors on signal quality.
- develop the techniques to combat the errors.
- enhance the skill in the design of DSP systems.

5.1 Introduction

When digital systems are implemented either in hardware or in software, the filter coefficients are stored in binary registers. These registers can accommodate only a finite number of bits and hence, the filter coefficients have to be truncated or rounded-off in order to fit into these registers. Truncation or rounding of the data results in degradation of system performance. Also, in digital processing systems, a continuous-time input signal is sampled and quantized in order to get the digital signal. The process of quantization introduces an error in the signal which is called round off noise. This makes the system non-linear, and leads to limit cycle behaviour.

In general the effects due to finite precision representation of numbers in a digital system are commonly referred to as finite word length effects. Some of the finite word length effects in digital filters are:

1. Errors due to quantization of input data by A/D converter.
2. Errors due to quantization of filter coefficients.
3. Errors due to rounding the product in multiplication.
4. Errors due to overflow in addition.
5. Limit cycles.

The effects of these errors introduced by signal processing depend on a number of factors which include the type of arithmetic used the quality of the input signal and the DSP algorithm implemented. These are discussed in this chapter.

5.2 Representation of Numbers in Digital System

The basic operations involved in digital signal processing are multiplications, additions and delays. They are often carried out using either fixed-point or floating-point arithmetic. Block floating-point arithmetic combines the benefits of the above two operations. Fixed-point arithmetic is the most widely used arithmetic in DSP because it is very fast and less expensive when implemented. However, it is limited in the range of numbers that can be represented. Further it is susceptible to problems of overflow which may occur when the result of an addition exceeds the permissible number range. To prevent this the operands are scaled. However, this degrades the performance of DSP systems which reduces the signal to noise ratio.

Floating-point arithmetic is preferred where the magnitude of the variables or system coefficients vary widely and eliminates overflow problem. Further, floating-point processing simplifies programming. However, floating point arithmetic is more expensive and often slower. While fixed-point digital signal processors with large word lengths are extensively used in DSP techniques where wide dynamic range and high precision are required the floating point processing provides a simpler and more natural way of achieving these requirements. The applications of floating-point arithmetic include real time parameter equalization of digital audio signals, spectrum analysis in radar and sonar, seismology, biomedicine *etc.* The above two arithmetic operations are discussed below.

5.2.1 Fixed Point Representation

In fixed point representation, the bits allotted for integer part and fraction part are fixed, and so the position of binary point is also fixed.

5.2.1.1 *Positive Binary Fraction Number*

In fixed point representation, there is only one unique way of representing positive binary number as given by the following equation:

$$\text{Positive binary fraction number, } N_p = \sum_{i=0}^{B} d_i 2^{-i} \tag{5.1}$$

where d_i = ith digit of the number and B = number of fractional digits.

5.2.1.2 *Negative Binary Fraction Number*

In fixed point representation, there are three different formats for representing negative binary numbers. They are:

1. Sign magnitude format
2. One's complement format
3. Two's complement format

1. Sign Magnitude Format

Except the sign bit all other digits of the negative of a given number are same as that of its positive representation. The sign bit is 0 for positive number and 1 for negative number.

$$\text{Positive binary fraction number, } N_p = (0 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i} \tag{5.2}$$

$$\text{Negative binary fraction number, } N_n = (1 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i} \tag{5.3}$$

For example,

$$+0.125_{10} \longrightarrow 0.001_2$$
$$-0.125_{10} \longrightarrow 1.001_2$$

2. One's Complement Format

- In one's complement format, the positive number is same as that of sign-magnitude format.
- The negative of the given number is obtained by bit by bit complement of its positive representation

$$\text{Complement of } d_i = \bar{d}_i = (1 - d_i) \tag{5.4}$$

Negative binary fraction number in one's complement is,

$$N_{1c} = (1 \times 2^0) + \sum_{i=1}^{B} (1 - d_i) 2^{-i} \tag{5.5}$$

For example,

$$+0.125_{10} \longrightarrow 0.001_2$$
$$-0.125_{10} \longrightarrow 1.110_2$$

3. Two's Complement Format

- In two's complement format, the positive number is same as that of the given magnitude format.
- The negative of the given number is obtained by taking one's complement of its positive representation and then adding one to the least significant bit.

Negative binary fraction number in two's complement is,

$$N_{2c} = (1 \times 2^0) + \sum_{i=1}^{B} (1 - d_i) 2^{-i} + (1 \times 2^{-B}) \tag{5.6}$$

For example,

$$+0.125_{10} \longrightarrow 0.001_2$$

$$-0.125_{10} \longrightarrow 1.111_2$$

Disadvantage of Fixed Point Representation

It is impossible to represent too large and too small numbers by fixed point representation. Therefore, the range of numbers that can be represented in fixed point method for a given binary word size is less compared in floating point representation.

Example 5.3

Represent the following number in fixed point representation.

$$(a)\ +0.375_{10} \qquad (b)\ -0.75_{10}$$

Solution

$(a)\ +(0.375)_{10}$

In fixed point representation

$$+0.375_{10} = (0.011)_2$$

$(b)\ -(0.75)_{10}$

In fixed point representation there are three different formats for representing negative number:

(*i*) Sign-magnitude: $-(0.75)_{10} = 1.110$.
(*ii*) One's complement: $-(0.75)_{10} = 1.001$.
(*iii*) Two's complement: $-(0.75)_{10} = 1.010$.

Example 5.4

Represent the following numbers in sign magnitude form.

$$(a)\ +8.25_{10} \qquad (b)\ -8.25_{10}$$

Solution

$$+(8.25)_{10} = 01000.010$$
$$\uparrow$$
$$\text{sign bit}$$
$$-(8.25)_{10} = 11000.010$$
$$\uparrow$$
$$\text{sign bit}$$

Example 5.5

Represent the following numbers in one complement form.

$$(a)\ -0.375_{10} \qquad (b)\ -0.0625_{10}$$

Solution

(a) $-0.375_{10} = (1.1001111)_2$

$$0.375_{10} = (0.0110000)_2$$

Complementing each bit we get $-0.375_{10} = (1.1001111)_2$.

(b) $-0.0625_{10} = (1.1110111)_2$

$$0.0625_{10} = (0.0001000)_2$$

Complementing each bit we get $-0.0625_{10} = (1.1110111)_2$.

Example 5.6

Represent the following numbers in two's complement form.

(a) -0.125_{10} (b) -0.25_{10}

Solution

(a) -0.125_{10} where $+0.125_{10} = 0.001_2$

One's complement $\Rightarrow 1.110_2$

Two's complement $\Rightarrow 1.111_2$

(b) -0.25_{10} where $+0.25_{10} = 0.010_2$

One's complement $\Rightarrow 1.101_2$

Two's complement $\Rightarrow 1.110_2$

5.2 Floating Point Representation

In floating point representation, the binary point can be shifted to desired position so that bits in the integer part and fraction part of a number can be varied. In general, the floating point number can be represented as

$$N_f = M \times 2^E \tag{5.7}$$

where M = mantissa and E = exponent.

Mantissa

- It will be in binary fraction format and in the range of $0.5 \leq M \leq 1$.
- If mantissa is characterized by 5 bits in which the left most one bit is used for representing sign and other 4 bits are used to represent a binary fraction number.

Exponent

- It is either a positive or negative integer.
- If it is characterized by 3 bits out of which the left most one bit is used to represent sign and the other two bits are used to represent a positive or negative binary integer number. The representation of floating point number is shown in Figure 5.1.

The range of number that can be represented by floating point format is from $\pm(2^{-4} \times 2^{-3})$ to $\pm((2 - 2^{-4}) \times 2^{-3})$.

Here 4 in 2^{-4} represents the 4 bits allotted for fractional binary number in mantissa and the 3 in 2^{-3} or 2^{+3} represents the maximum size of integer.

The IEEE-754 standard for 32 bit single precision floating point number is given by

$$N_f = (-1)^S \times 2^{E-127} \times M \tag{5.8}$$

where

S = 1-bit field for sign of number.
E = ε-bit field for exponent.
M = 23-bit field for Mantissa.

IEEE 754 format for 32 bit floating point number is shown in Figure 5.2.

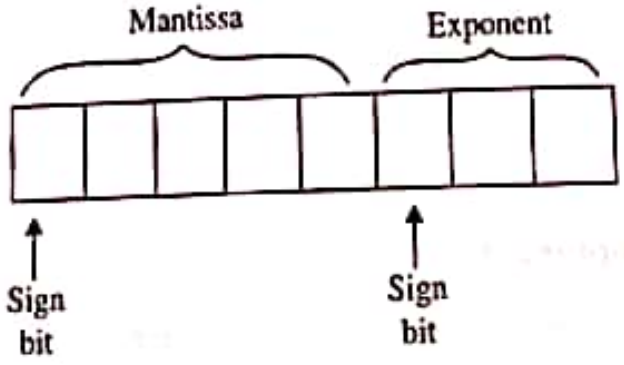


Figure 5.1 Floating point number representation.

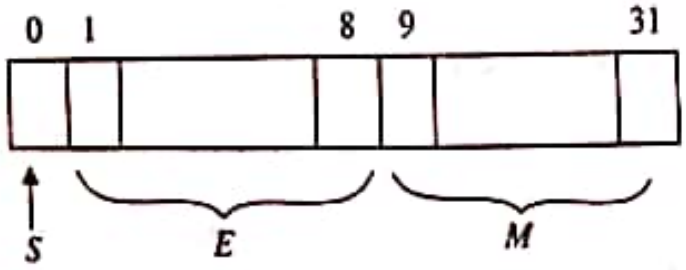


Figure 5.2 IEEE 754 format for 32 bit floating point number.

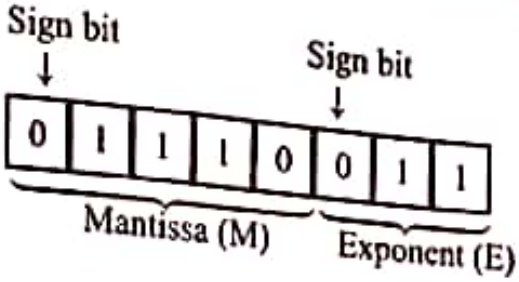
Example 5.7

Represent the following numbers in floating point representation with five bits for mantissa and three bits for exponent.

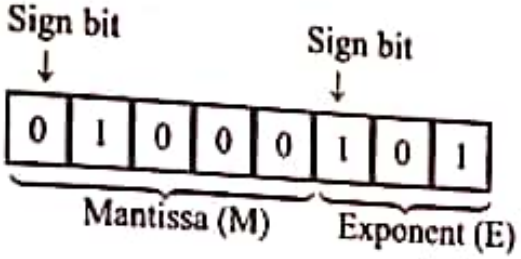
(*a*) $+7_{10}$ (*b*) $+0.25_{10}$ (*c*) -7_{10} (*d*) -0.25_{10}

Solution

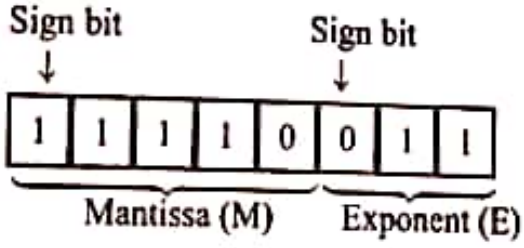
(*a*) $+7_{10} = +111_2 = 0.1110 \times 2^{+3} = 0.1110 \times 2^{+11_2}$



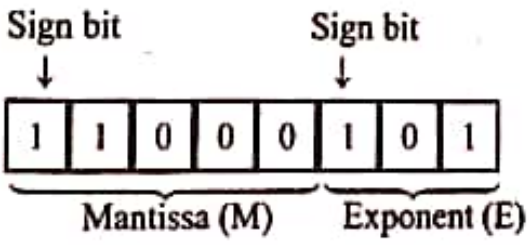
(*b*) $+0.25_{10} = +0.01_2 = 0.0100 \times 2^0 = 0.1000 \times 2^{-1} = 0.1000 \times 2^{-01_2}$



(*c*) $-7_{10} = -111_2 = 1.1110 \times 2^{+3} = 1.1110 \times 2^{+11_2}$



(*d*) $-0.25_{10} = -0.01_2 = 1.0100 \times 2^0 = 1.1000 \times 2^{-1} = 1.1000 \times 2^{-01_2}$



Example 5.12

Multiply $+0.125_{10}$ and $+5_{10}$ in floating point format.

Solution

$$+0.125_{10} = 0.100000 \times 2^{-2_{10}}$$
$$+5_{10} = 0.101000 \times 2^{+3_{10}}$$
$$+0.125_{10} \times +5_{10} \Rightarrow (0.100000 \times 0.101000) \times 2^{-2+3}$$
$$\Rightarrow 0.010100 \times 2^{1}$$

$$0.010100 \times 2^{1} \xrightarrow{\text{normalized}} 0.10100 \times 2^{0} = 0.625_{10}$$

5.3 Methods of Quantization

The process of converting a discrete-time continuous amplitude signal into a digital signal by expressing each sample value as a finite number of digits is called **quantization**. The error introduced in representing the continuous valued signal by a finite set of discrete level is called "**quantization error or quantization noise**". The quantization error is a sequence which is defined as the difference between the quantized value and the actual sample value. The actual values of the samples of $x(n)$ cannot be processed by DSP or a digital computer since it is very difficult to store and manipulate all the samples. To eliminate the excess digits that occur due to quantization either discard them (truncation) or discard them by rounding the resulting number (rounding).

Thus, there are two methods of quantization employed in digital system. They are: (1) Truncation; and (2) Rounding. They are discussed below.

5.3.1 Truncation

Truncation is the process of reducing the size of binary numbers by discarding all bits less significant than the least significant bit that is retained. In the truncation of

binary numbers to b bits, all the less significant bits beyond b^{th} bit are discarded. The quantization steps are marked on y-axis and the range of unquantized numbers are marked on x-axis.

1. Any positive unquantized number in the range $0 \leq N \leq (1 \times 2^{-b})$ will be assigned the quantization step (0×2^{-b}).
2. Any positive unquantized number in the range $(1 \times 2^{-b}) \leq N \leq (2 \times 2^{-b})$ will be assigned the quantization step (1×2^{-b}) and so on.

Example 5.13

Perform the quantization of 0.875_{10} to 2 bit by truncation.

Solution

$$0.875_{10} \xrightarrow[\text{to binary}]{\text{convert}} 0.1110_2 \xrightarrow{\text{truncate to 2 bits}} 0.11_2 \xrightarrow[\text{to decimal}]{\text{convert}} 0.75_{10}$$

11.1 Fixed Point Number System

In fixed point number system, the effect of truncation on positive numbers are same in all the three representations. The error due to truncation of negative number depends on the type of representation of the number. Let N = unquantized fixed point binary numbers and N_t = fixed point binary number quantized by truncation. The quantization error due to truncation is

$$\text{Truncation error, } e_t = N_t - N \tag{5.9}$$

The range of errors in truncation of fixed point numbers in different types of representation are shown in Figure 5.3 and tabulated in Table 5.1.

The truncation of a positive number results in a number that is smaller than the unquantized number hence truncation error is always negative.

For the truncation of negative numbers represented in sign magnitude and one's complement format, the error is always positive because truncation basically reduces the magnitude of the numbers.

Table 5.1 Range of error in truncation of fixed point numbers

Numbers and its representation	Range of error when truncated to b bits
Positive numbers	$0 \geq e > -2^{-b}$
Sign magnitude negative number	$0 \leq e < -2^{-b}$
One's complement negative number	$0 \leq e < -2^{-b}$
Two's complement negative number	$0 \geq e > -2^{-b}$

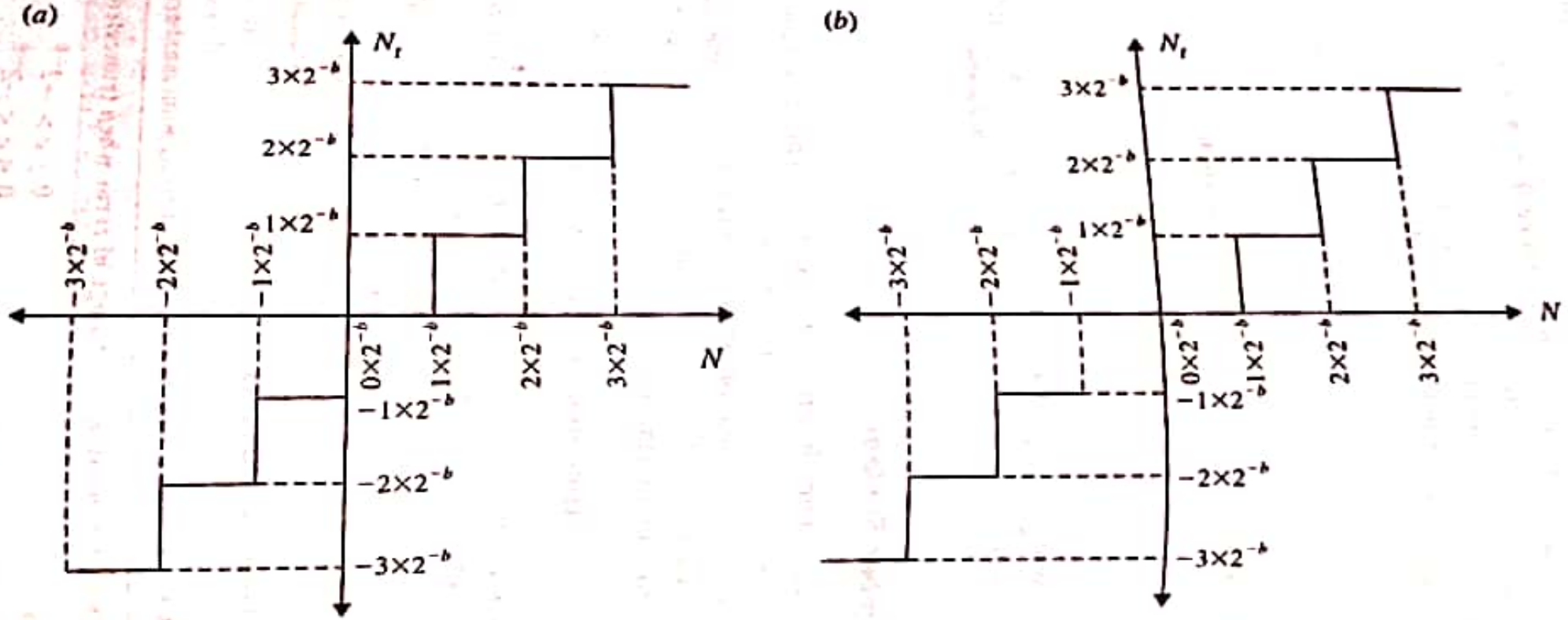


Figure 5.3 Truncation characteristics of (*a*) 2's complement, (*b*) sign magnitude and 1's complement.

In the two's complement representation the negative of a number is obtained by subtracting the corresponding positive number from 2. Therefore, the effect of truncation on a negative number is to increase the magnitude of the negative number and so the truncation error is always negative.

5.3.1.2 Floating Point Number System

In floating point representation, the mantissa of the number alone is truncated. The truncation error in a floating point number is proportional to the number being quantized.

Let N_f = unquantized floating point binary number and N_{tf} = truncated floating point binary number. Now

$$N_{tf} = N_f + N_f \varepsilon_t \tag{5.10}$$

where ε_t is the relative error due to truncation of floating point number. Relative error due to truncation is,

$$\varepsilon_t = \frac{N_{tf} - N_f}{N_f} \tag{5.11}$$

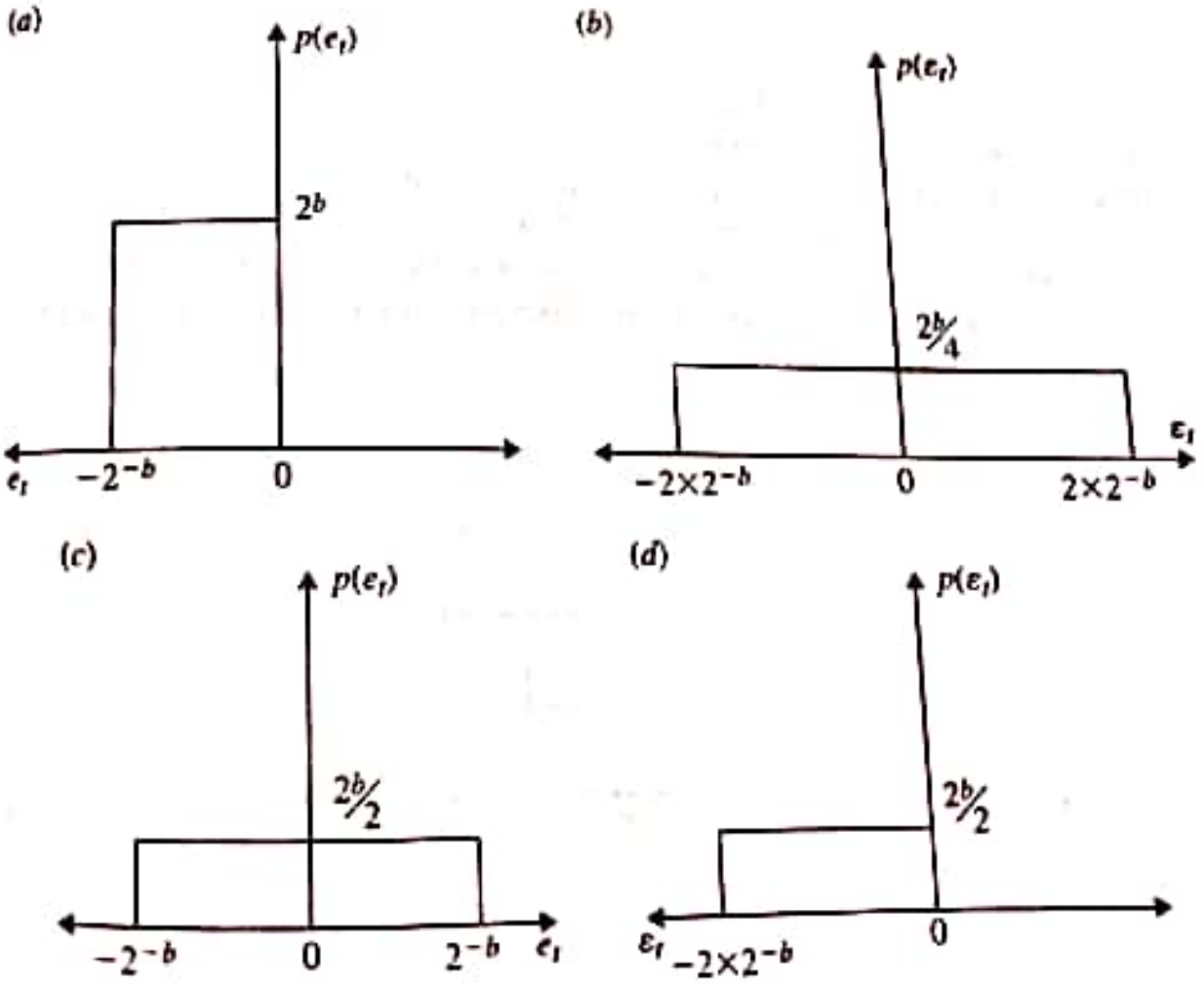


Figure 5.4 Quantization noise probability density function for truncations. (*a*) Fixed point-two complement, (*b*) Floating point when mantissa in two's complement, (*c*) Fixed point-one's complement or sign magnitude and (*d*) Floating point when mantissa is one's complement or in sign magnitude.

Table 5.2 Range of error in truncation of floating point number

Type of representation for mantissa	Range of error when mantissa is truncated to b bits
Two's complement positive mantissa	$0 \geq e_t > -2^{-b} \times 2$
Two's complement negative mantissa	$0 \leq e_t < 2^{-b} \times 2$
One's complement positive and negative mantissa	$0 \leq e_t < -2 \times 2^{-b}$
Sign magnitude positive and negative mantissa	$0 \leq e_t < -2 \times 2^{-b}$

In truncation of binary number, the range of error is known but the probability of obtaining an error within the range is not known. Hence, it is assumed that the errors occur uniformly throughout the interval. The range of error in truncation of floating point number is shown in Table 5.2 and the corresponding quantization noise probability density function is shown in Figure 5.4.

5.3.2 Rounding

Rounding is the process of reducing the size of a binary number to finite word size of b bits such that the rounded b bit number is closest to the original unquantized number. The rounding process consists of truncation and addition. In rounding of a number to b bits, first the unquantized number is truncated to b bits by retaining the most significant b bits. Then a zero or one is added to LSB of the truncated number depending on the bit that is next to the least significant bit that is retained. If the bit next to the least significant bit that is retained is zero, then zero is added to the least significant bit of the truncated number. If the bit next to the least significant bit that is retained is one, then one is added to the least significant bit of the truncated number. The input-output characteristics of the quantizer used for rounding is shown in Figure 5.5.

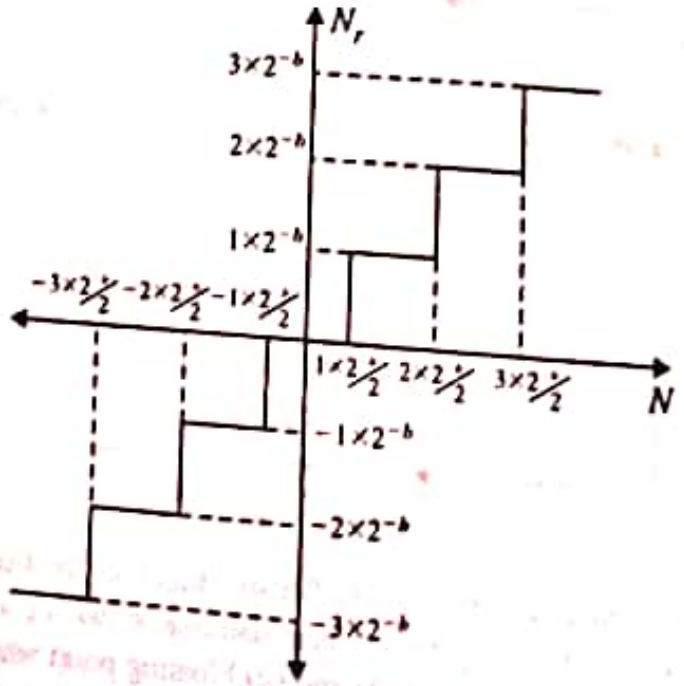


Figure 5.5 Input-output characteristics of the quantizer used for rounding.

The quantization steps are marked on y-axis and the range of unquantized numbers are marked on x-axis.

1. Any positive unquantized number in the range $1 \times \frac{2^{-b}}{2} \le N < 2 \times \frac{2^{-b}}{2}$ will be assigned the quantization step 1×2^{-b}.
2. Any positive unquantized number in the range $2 \times \frac{2^{-b}}{2} \le N < 3 \times \frac{2^{-b}}{2}$ will be assigned the quantization step 2×2^{-b} and so on.

Example 5.14

Perform the quantization of 0.0625_{10} to 3 bit by rounding.

Solution

$$0.0625_{10} \xrightarrow[\text{to binary}]{\text{convert}} 0.0001_2 \xrightarrow{\text{rounded to 3 bits}} 0.001_2 \xrightarrow[\text{to decimal}]{\text{convert}} 0.125_{10}$$

Fixed Point Number

Let N = unquantized fixed point binary number and N_r = fixed point binary number quantized by rounding. The quantization error in fixed point number due to rounding is defined as

$$\text{Rounding error, } e_r = N_r - N \tag{5.12}$$

The range of error due to rounding for all the three formats of fixed point representation is same. In fixed point representation the range of error made by rounding a number to b bits is

$$\frac{-2^{-b}}{2} \le e_r \le \frac{2^{-b}}{2} \tag{5.13}$$

Floating Point Number

Let N_f = unquantized floating point binary number and N_{rf} = rounded floating point binary number. Now

$$N_{rf} = N_f + N_f \varepsilon_r \tag{5.14}$$

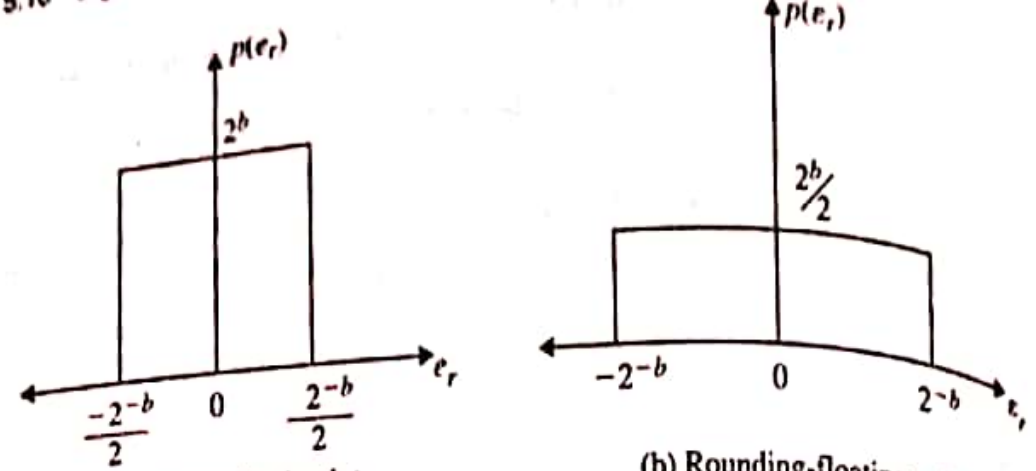
where ε_r is the relative error due to rounding of a floating point number.

$$\therefore \text{Relative error due to rounding, } \varepsilon_r = \frac{N_{rf} - N_f}{N_f} \tag{5.15}$$

The range of error by rounding a number in floating point representation to b bits is,

$$-2^{-b} \le \varepsilon_r \le 2^{-b}. \tag{5.16}$$

The probability density function for rounding fixed point and floating numbers are shown in Figure 5.6.



(a) Rounding-fixed point (b) Rounding-floating point

Figure 5.6 Quantization noise probability density functions for rounding.

5.4 Quantization of Input Data by Analog to Digital Converter

The process of analog to digital conversion involves: (i) sampling the continuous time signal at a rate much greater than Nyguist rate and (ii) qunatising the amplitude of the sampled signal into a set of discrete amplitude levels. The input-output characteristics of a uniform quantizers is shown in Figure 5.7. This quantizer rounds the sampled signal to the nearest quantized output level. The difference between the quantized signal amplitude $x_q(n)$ and the actual signal amplitude $x(n)$ is called the quantization error $e(n)$. That is

$$e(n) = x_q(n) - x(n)$$

Since, rounding is involved in the process of quantization the range of values for the quantization error is

$$-\frac{2^{-b}}{2} \leq e(n) \leq \frac{2^{-b}}{2} \tag{5.17}$$

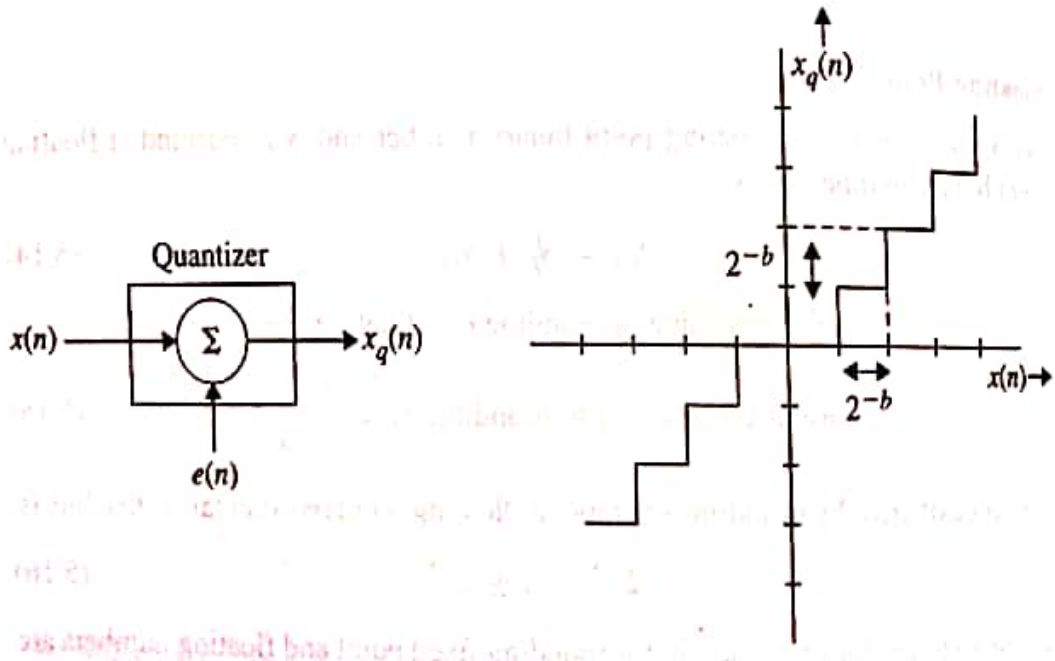


Figure 5.7 Two complement number quantization.

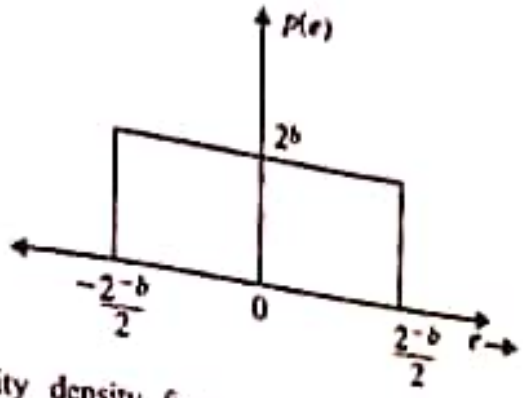


Figure 5.8 Probability density function for quantization round-off error in A/D conversion.

The quantization error is assumed to be uniformly distributed over $-\frac{2^{-b}}{2} \le e(n) \le \frac{2^{-b}}{2}$. It is also assumed that this quantization noise $e(n)$ is a stationary white noise sequence $x(n)$ which traverses several quantization levels between two successive samples.

In the process of quantization, the samples value is rounded off to the nearest quantization level. The probability density function for the quantization round off error in A/D conversion is shown in Figure 5.8.

It can be noted from the Figure 5.8, that the quantization error is uniformly distributed and the mean value of error in zero. The power of the quantization noise, which is nothing but variance (σ_e^2) is given by

$$\sigma_e^2 = E[e^2(n)] - E^2[e(n)] \tag{5.18}$$

$$= E[e^2(n)] \tag{5.19}$$

Therefore, mean value of error is zero, *i.e.*, $E[e(n)] = 0$.

Qualification step size is expressed as,

$$q = \frac{R}{2^b} \qquad \text{(for two's complement)} \tag{5.20}$$

where R = range of analog signal to be quantized. Usually the analog signal is scaled such that the magnitude of quantized signal is less or equal to one. In such case the range of analog signal to be quantized is -1 to 1 therefore $R = 2$.

$$\text{Quantization step size } q = \frac{2}{2^b} = 2.2^{-b} \tag{5.21}$$

The quantization error for rounding will be in the range of $-q/2$ to $+q/2$

$$\therefore \text{ Variance of error signal } \sigma_e^2 = \frac{1}{\frac{q}{2} - (-q/2)} \int_{-q/2}^{q/2} e^2 de \tag{5.22}$$

$$= \frac{1}{q}\left[\frac{e^3}{3}\right]_{-(q/2)}^{q/2} = \frac{1}{3q}\left[\frac{q^3}{8} + \frac{q^3}{8}\right]$$

$$= \frac{1}{3q} \cdot \frac{2q^3}{8} = \frac{q^2}{12} \tag{5.23}$$

$$\sigma_e^2 = \frac{1}{12}\left(\frac{R}{2^{-b}}\right)^2 \tag{5.24}$$

The variance of error signal is also called steady state noise power due to input quantization.

5.4.1 Output Noise Power due to the Quantization Error Signal

After converting the continuous time signal into digital signal, let us assume that this quantized signal is applied as an input to a digital system with impulse response $h(n)$.

The quantized input signal of a digital system can be represented as a sum of unquantized signal $x(n)$ and error signal $e(n)$ as shown in Figure 5.9

$$\begin{aligned} y'(n) &= x_q(n) * h(n) \\ &= [x(n) + e(n)] * h(n) \end{aligned} \tag{5.25}$$

$$y'(n) = [x(n) * h(n)] + [e(n) * h(n)] \tag{5.26}$$

$$y'(n) = y(n) + e(n) \tag{5.27}$$

where

$y(n) = x(n) * h(n)$ is the output due to input signal

$\epsilon(n) = e(n) * h(n)$ is the output due to error signal

Variance of the signal $\epsilon(n)$ is called the output noise power or steady output noise power.

Output noise power (or) steady state output noise power due to quantization errors is given by the following equation:

$$\sigma_{e0}^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n) \tag{5.28}$$

The summation of $h^2(n)$ can be evaluated using Parseval's theorem

$$\sigma_{e0}^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n) = \sigma_e^2 \frac{1}{2\pi j} \oint_c H(z)H(z^{-1})z^{-1}dz \tag{5.29}$$

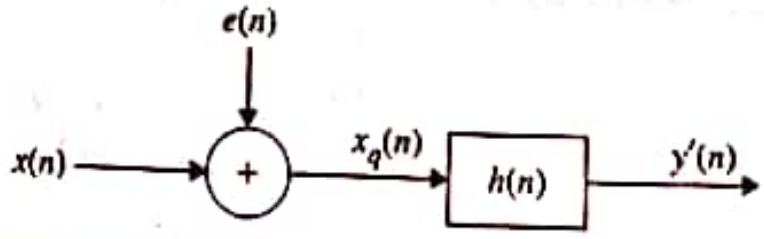


Figure 5.9 Representation of quantization noise in digital system.

The closed contour integration can be evaluated using residue theorem of z-transform

$$\sigma_{e0}^2 = \sigma_e^2 \sum_{i=1}^{N} \text{Res}[H(z)H(z^{-1})z^{-1}]|_{z=p_i} \tag{5.30}$$

where $p_1, p_2, \ldots, p_n$ are poles of $H(z)H(z^{-1})z^{-1}$. Since the closed contour integration is around the unit circle $|z| = 1$, only the residue of the poles that is inside the unit circle are considered.

Example 5.15

The output of an A/D converter is applied to a digital filter whose system function is,

$$H(z) = \frac{z(0.5)}{z - 0.5}$$

Find the output noise power from the digital filter, when the input signal is quantized to have eight bits.

Solution Given $b = 8$ (assuming sign bit is included). Let $R = 2$

$$\text{Quantization step size } q = \frac{R}{2^b} = \frac{2}{2^b} = 2^{2-b} = 2 \times 2^{-8} = 2^{-7}.$$

The input quantization noise power is obtained using equation (5.23)

$$\sigma_e^2 = \frac{q^2}{12} = \frac{(2^{-7})^2}{12} = \frac{2^{-4}}{12} = 5.086 \times 10^{-6}$$

The output noise power is given by

$$\sigma_{e0}^2 = \frac{\sigma_e^2}{2\pi j} \oint_c H(z)H(z^{-1})z^{-1}dz$$

$$= \sigma_e^2 \sum_{i=1}^{N} \text{Res}[H(z)H(z^{-1})z^{-1}]|_{z=p_i}$$

$$H(z)H(z^{-1})z^{-1} = \frac{0.5z}{z - 0.5} \cdot \frac{0.5z^{-1}}{z^{-1} - 0.5} \cdot z^{-1}$$

$$= \frac{0.25z^{-1}}{z^{-1}(z - 0.5)(1 - 0.5z)}$$

$$= \frac{0.25}{(z - 0.5)(1 - 0.5z)} \quad \text{[poles are at } z = 0.5, z = 2\text{]}$$

$\text{Res}[H(z)H(z^{-1})z^{-1}]$ due to pole $z = 0.5$ alone is to be considered.

$$\text{Res}[H(z)H(z^{-1})z^{-1}]|_{z=0.5} = \left.\frac{(z-0.5)0.25}{(z-0.5)(1-0.5z)}\right|_{z=0.5}$$

[$\because$ $z = 0.5$ pole lies insider the unit circle]

$$= \frac{1}{3}$$

Therefore, the output noise power is

$$\sigma_{e0}^2 = \sigma_e^2 \times \text{Res}[H(z)H(z^{-1})z^{-1}]|_{z=0.5}$$

$$= 5.086 \times 10^{-6} \times \frac{1}{3}$$

$$\boxed{\sigma_{e0}^2 = 1.6954 \times 10^{-6}.}$$

Example 5.17

The input to the system

$$y(n) = 0.999y(n-1) + x(n)$$

is applied to an ADC. What is the power produced by the quantization noise at the output of the filter if the input is quantized to (*a*) 8 bits and (*b*) 16 bits?

(Anna University, May, 2007)

Solution Given

$$y(n) = 0.999y(n-1) + x(n)$$

Taking z-transform on the both sides we get

$$Y(z) = 0.999z^{-1}Y(z) + X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 0.999z^{-1}}$$

The quantization noise power at the output of the digital filter is

$$\sigma_{e_0}^2 = \sigma_e^2 \sum_{i=1}^{N} \text{Res}[H(z)H(z^{-1})z^{-1}]|_{z=p_i}$$

$$H(z)H(z^{-1})z^{-1} = \left(\frac{1}{1 - 0.999z^{-1}}\right)\left(\frac{1}{1 - 0.999z}\right)z^{-1}$$

$$= \frac{z^{-1}}{z^{-1}(z - 0.999)(1 - 0.999z)}$$

$$= \frac{z^{-1}}{(z - 0.999)(1 - 0.999z)}$$

Here $H(z)$ has only one pole at $p = 0.999$ which lies inside the unit circle. Therefore

$$\sum_{i=1}^{N} \text{Res}[H(z)H(z^{-1})z^{-1}]\Big|_{z=p_i} = \text{Res}[H(z)H(z^{-1})z^{-1}]\Big|_{z=0.999}$$

$$= (z - 0.999)\frac{1}{[z - 0.999][1 - 0.999z]}\Big|_{z=0.999}$$

$$= \frac{1}{(1 - 0.999z)}\Big|_{z=0.999}$$

$$= \frac{1}{(1 - 0.999z)^2}$$

$$= 500.25$$

Therefore

$$\sigma_{e_0}^2 = \sigma_e^2(500.25)$$

$$= \frac{q^2}{12}(500.25) = \frac{\left(\frac{R}{2^b}\right)^2}{12}(500.25)$$

Let $R = 2V$

(*a*) Given $b = 8$ bits (including sign bit)

$$\sigma_{e_0}^2 = \frac{\left(\frac{2}{2^8}\right)^2}{12}[500.25]$$

$$= \frac{2^{-14}}{12}(500.25) = 2.544 \times 10^{-3}$$

(*b*) Given $b = 16$ bits

$$\sigma_{e_0}^2 = \frac{\left(\frac{2}{2^{16}}\right)^2}{12}[500.25]$$

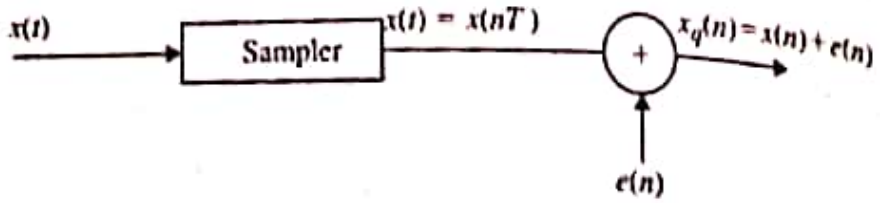
$$= \frac{2^{-30}}{12}(500.25) = 3.882 \times 10^{-8}$$

Example 5.18

Consider $(b + 1)$-bits (including sign bit) bipolar A/D converter. Obtain an expression for signal to quantization noise ratio. State the assumption made.

(Anna University, May 2007)

Solution The quantization noise model of A/D converter is,



The A/D converter output is the sum of the input signal $x(n)$ and the error signal $e(n)$. If the rounding is used for quantization then the quantization error is

$$e(n) = x_q(n) - x(n) \text{ is bounded by } \frac{-q}{2} \le e(n) \le \frac{q}{2}$$

In most cases, we can assume that the A/D conversion error $e(n)$ has the following properties:

(i) The error sequence $e(n)$ is a sample sequence of a stationary random process.
(ii) The error signal is uncorrected with $x(n)$ and other signal in the system.
(iii) The error is a white noise process with uniform amplitude probability distribution over the range of quantization error.

The variance of $e(n)$ is given by

$$\sigma_e^2 = E[e^2(n)] - E^2[e(n)]$$

$$\sigma_e^2 = E[e^2(n)]$$

$$= \frac{1}{q}\int_{-q/2}^{q/2} e^2(n)de = \frac{1}{q}\left[\frac{e^3}{3}\right]_{-q/2}^{q/2}$$

$$= \frac{1}{3q} \times \left[\frac{q^3}{8} + \frac{q^3}{8}\right] = \frac{q^2}{12}$$

$$\sigma_e^2 = \frac{\left(\frac{2}{2^{b+1}}\right)^2}{12}$$

$$\boxed{\sigma_e^2 = \frac{2^{-2b}}{12}}$$

where b is number of bits (excluding sign bit). σ_e^2 is also known as the steady state noise power due to input quantization.

If the input signal is $x(n)$ and its variance is σ_x^2, then the ratio of signal power to noise power which is known as signal to noise ratio for rounding is

$$\boxed{\text{SNR} = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{\frac{2^{-2b}}{12}} = 12(2^{2b}\sigma_x^2)}$$

Signal to noise ratio in dB is expressed as

$$\text{SNR (dB)} = 10\log_{10}\frac{\sigma_x^2}{\sigma_e^2} = 10\log_{10}(12\,2^{2b}\sigma_x^2)$$

$$\boxed{\text{SNR (dB)} = 6.02b + 10.79 + 10\log_{10}\sigma_x^2}$$

SNR increases approximately by 6 dB for each bit added to register length.

5.5 Quantization of Filter Coefficients

In the design of a digital filter the coefficients are evaluated with infinite precision. But they are limited by the word length of the register used to store the coefficients. Usually the filter coefficients are quantized to the word size of the register used to store them either by truncation or by rounding.

The location of poles and zeros of the digital filters directly depends on the value of filter coefficients. The quantization of the filter coefficients will modify the value of poles and zeros and so the location of the poles and zeros will be shifted from the desired location. This will create deviation in the frequency response of the system. Hence we obtain a filter having a frequency response that is different from the frequency response of the filter with unquantized coefficients. The sensitivity of the filter frequency response characteristics to quantization of the filter coefficients is minimized by realizing the filter having a large number of poles and zeros as an interconnection of second-order section. **Therefore, the coefficient quantization has less effect in cascade realization when compared to other realizations.**

Example 5.19

For the second order IIR filter, the system function is,

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

Study the effect of shift in pole location with 3 bit coefficient representation in direct and cascade forms.

Solution

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.45z^{-1})} = \frac{z^2}{(z-0.5)(z-0.45)}$$

Original poles of $H(z) \Longrightarrow p_1 = 0.5$ and $p_2 = 0.45$.

Case (i) Direct Form

$$H(z) = \frac{1}{(1 - 0.5z^{-1})(1 - 0.45z^{-1})} = \frac{1}{(1 - 0.95z^{-1} + 0.225z^{-1})}$$

Quantization of coefficient by truncation

$$.95_{10} \xrightarrow{\text{Convert to binary}} .1111_2 \xrightarrow{\text{Truncate to 3 bits}} .111_2 \xrightarrow{\text{Convert to decimal}} .815_{10}$$

$$.225_{10} \xrightarrow{\text{Convert to binary}} .0011_2 \xrightarrow{\text{Truncate to 3 bits}} .001_2 \xrightarrow{\text{Convert to decimal}} .125_{10}$$

$$H(z) = \frac{1}{1 - 0.875z^{-1} + 0.125z^{-2}}$$

$$\boxed{H(z) = \frac{1}{(1 - 0.695z^{-1})(1 - 0.179z^{-1})}}$$

The poles are at $p_1 = 0.695$ and $p_2 = 0.179$.

Case (ii) Cascade Form

$$H(z) = \frac{1}{(1 - 0.5z^{-1})(1 - 0.45z^{-1})}$$

Quantization by truncation

$$.5_{10} \xrightarrow{\text{Convert to binary}} .1000_2 \xrightarrow{\text{Truncate to 3 bits}} .100_2 \xrightarrow{\text{Convert to decimal}} .5_{10}$$

$$.45_{10} \xrightarrow{\text{Convert to binary}} .0111_2 \xrightarrow{\text{Truncate to 3 bits}} .011_2 \xrightarrow{\text{Convert to decimal}} .375_{10}$$

$$\boxed{H(z) = \frac{1}{1 - 0.5z^{-1}} \times \frac{1}{1 - 0.375z^{-1}}}$$

The poles are $p_1 = 0.5$ and $p_2 = 0.375$.

Conclusion:

- From direct form, we can see that the quantized poles deviate very much from the original poles.
- From cascade form, we can see that one pole is exactly the same while the other pole is very close to the original pole.

5.6 Product Quantization Error

In fixed point arithmetic the product of two b bit numbers results in number of $2b$ bits length. If the word length of the register used to store the result is b bit, then

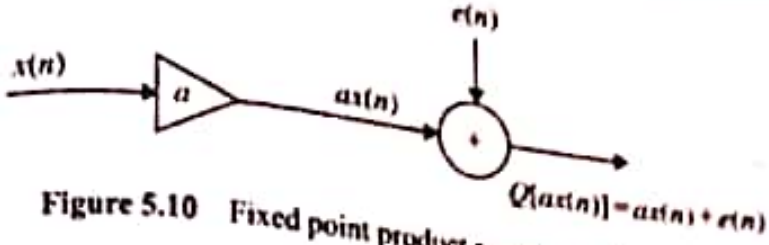


Figure 5.10 Fixed point product round off noise model

it is necessary to quantize the product to b bits, which produce an error known as **product quantization error or product round off noise**. In realization structures of digital system, multipliers are used to multiply the signal by constants.

The model for fixed point round off noise following a multiplication is shown in Figure 5.10.

The multiplication is modelled as an infinite precision multipliers followed by an adder where round off noise is added to the product so that over all result equals some quantization level. The roundoff noise sample is a zero mean random variable with a variance $(2^{-2b}/3)$, where b is the number of bits used to represent the variables.

In general the following assumptions are made regarding the statistical independence of the various noise sources in the digital filter.

1. Any two different samples from the same noise source are uncorrelated.
2. Any two different noise source, when considered as random processes are uncorrelated.
3. Each noise source is uncorrelated with the input sequence.

The product quantization noise model for first order and second order system are shown in Figure 5.11. The product quantization noise models for IIR using cascade is shown in Figure 5.12.

In each noise model there are a number of noise sources. The output noise variance due to each source is computed separately by considering one noise source at a time. The total output noise variance is given by sum of the output noise variance at all the noise sources. For each noise source, the noise transfer function (NTF) has to be determined by treating the noise source as input and the output being the output of the system. NTF for noise sources $e_{a11}(n)$ in Figure 5.12 $= H_1(z)$ and NTF for noise sources $e_{a12}(n)$ in Figure 5.12 $= H_2(z)$.

Let $e_k(n)$ be the error signal from k^{th} noise source, $h_k(n)$ the impulse response for k^{th} noise source and $T_k(n)$ the noise transfer function (NTF) for k^{th} noise source.

$$\text{Variance of } k^{\text{th}} \text{ noise source } \sigma_{ek}^2 = \frac{q^2}{12} = \frac{2^{-2b}}{3} \qquad [\because R = 2]$$

Output noise variance due to k^{th} noise source is,

$$\sigma_{e0k}^2 = \sigma_{ek}^2 \sum_{n=0}^{\infty} h_k^2(n)$$

Conclusion: Thus, in cascade form realization, the product noise roundoff power is less in case (*ii*) when compared to case (*i*) and also direct form realization.

5.7 Limit Cycles in Recursive System

5.7.1 Zero-Input Limit Cycles

In recursive systems, when the input is zero or some non-zero constant value, the non-linearities due to finite precision arithmetic operation may cause periodic oscillations, in the output. During periodic oscillations, the output $y(n)$ of a system will oscillate between a finite positive and negative value for increasing n or the output will become constant for increasing n. Such oscillations are called limit cycles. If the system output enters a limit cycle, it will continue to remain in limit cycle even when the input is made zero. Hence, these limit cycles are also called zero input limit cycles.

Consider the following difference equation of first order system with one pole only.

$$y(n) = ay(n-1) + x(n) \tag{5.33}$$

The system has one product $ay(n-1)$. If the product is quantized to finite word length then the response $y(n)$ will deviate from actual value. Let $y'(n)$ be the response of the system when the product is quantized.

$$y'(n) = Q\left[ay'(n-1)\right] + x(n) \tag{5.34}$$

5.7.2 Overflow Limit Cycle Oscillation

In fixed point addition of two binary numbers the overflow occurs when the sum exceeds the finite word length of the register used to store the sum. The overflow in addition may lead to oscillation in the output which are referred to as an overflow limit cycle. An overflow in addition of two or more binary numbers occurs when the sum exceeds the dynamic range of number system. Let us consider two positive numbers n_1 and n_2 which is represented in sign magnitudes.

$$n_1 = \frac{3}{8} \rightarrow 0.011$$

$$n_2 = \frac{6}{8} \rightarrow 0.110$$

$$n_1 + n_2 = 1.001 \rightarrow -\frac{1}{8} \text{ in sign magnitude}$$

In the above example, when two positive numbers are added the sum is wrongly interpreted as a negative number. The transfer characteristics of an adder is shown in Figure 5.16, where n is the input to the adder and $y(n)$ is the corresponding output. The overflow oscillations can be eliminated if saturation arithmetic is performed.

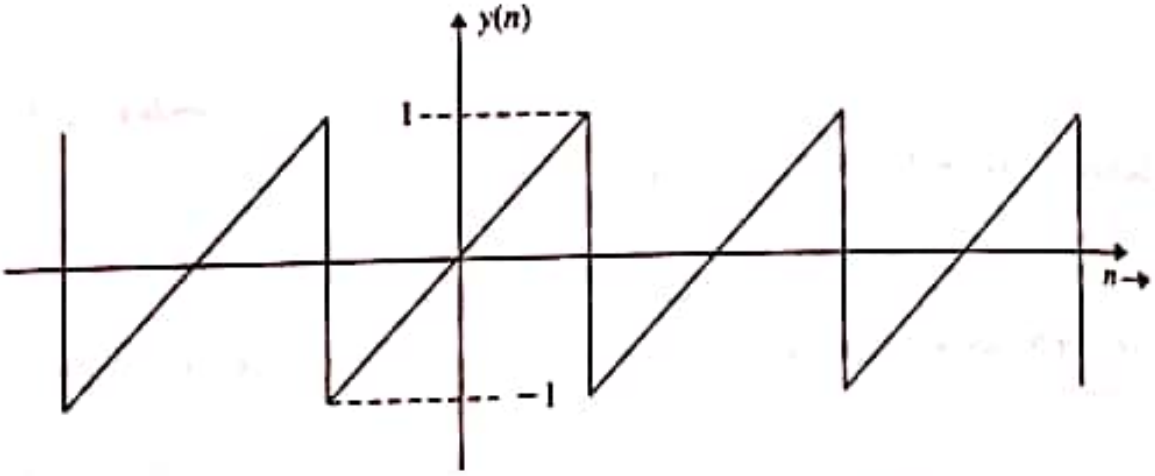


Figure 5.16 Transfer characteristics of an adder.

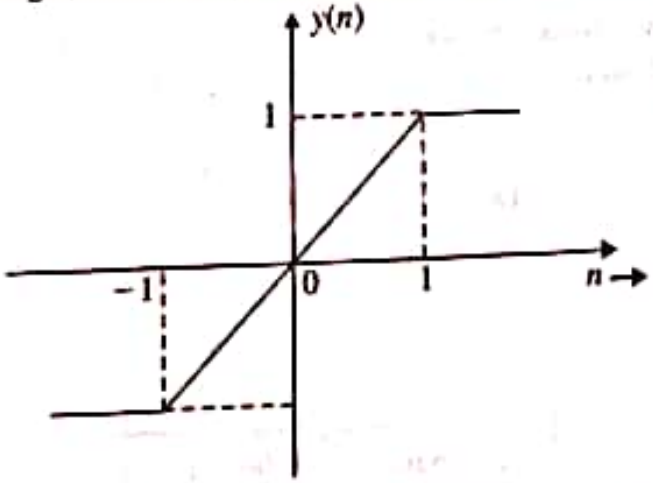


Figure 5.17 Characteristics of saturation adder.

The characteristics of saturation adder is shown in Figure 5.17. In saturation arithmetic, when an overflow is sensed, the output is set equal to maximum allowable value and when an underflow is sensed, the output (sum) is set equal to minimum allowable value. The saturation arithmetic introduces non-linearity in the adder and the signal distortion due to this non-linearity is small if the saturation occurs infrequently.

5.8 Scaling to Prevent Overflow

Saturation arithmetic eliminates limit cycle due to overflow, but it causes undesirable signal distortion due to non-linearity of the clipper. In order to limit the amount of non-linear distortion, it is important to scale the input signal to the adder such that the overflow becomes a rare event. Let

$x(n)$ = Input to the system

$h_m(n)$ = Impulse response between the input and output of node $-m$

$y_m(n)$ = Response of the system at node $-m$

To scale the input signal so that

$$\sum_{n=-\infty}^{\infty} |y_m(n)|^2 \leq S^2 \sum_{n=-\infty}^{\infty} |x(n)|^2$$

where S is the scaling factor, using Parseval's and residue theorems the expression for scaling factor is given by

$$S^2 = \frac{1}{\sum_{n=-\infty}^{\infty} |h_m(n)|^2} = \frac{1}{2\pi j \oint_c S(z)S(z^{-1})z^{-1}dz}$$

$$S^2 = \frac{1}{\sum_{i=1}^{N} \text{Res}[S(z)S(z^{-1})z^{-1}dz]|_{z=p_i}} \tag{5.37}$$

where $S(z)$ is the transfer function seen between the input to system and output of summing node $-m$. For example, consider the second order system which is shown in Figure 5.18.

The transfer function between the input to the system and output of adder A is given by

$$S(z) = \frac{W(z)}{X(z)}$$

The output signal of adder A is

$$W(z) = X(z) - a_1 z^{-1} W(z) - a_2 z^{-2} W(z)$$

$$X(z) = W(z) + a_1 z^{-1} W(z) + a_2 z^{-2} W(z)$$

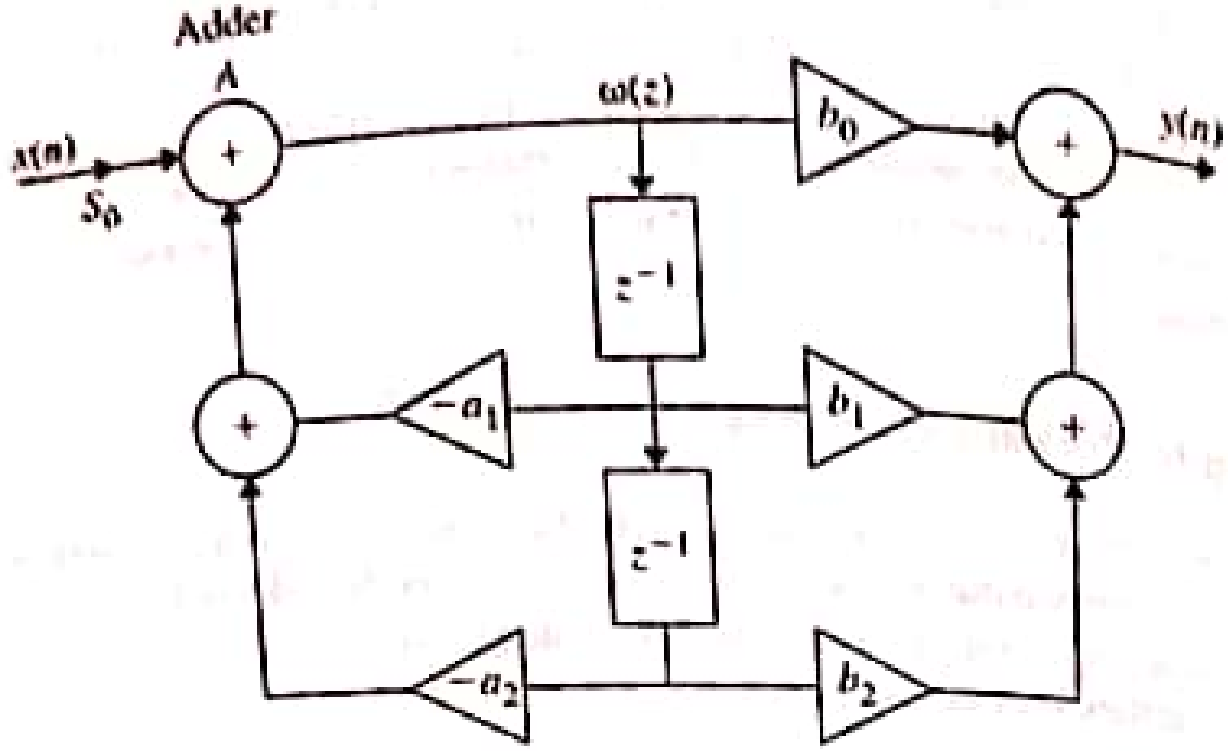


Figure 5.18 Second order system with input in scaled by S_0.

$$\therefore S(z) = \frac{W(z)}{X(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Summary

- The performance of a DSP system is limited by the number of bits used in the implementation. The common sources of errors are due to input quantization, coefficient quantization, product round off and addition overflow.
- The ADC quantization noise is reduced by increasing the number of ADC bits or by using multirate techniques.
- When an IIR digital filter is implemented, errors arise in representing the filter coefficients. These errors can be reduced to acceptable level by using more bits. However, this increases the cost.
- Addition of two large numbers of a similar sign may produce an overflow which results in excess of permissible word length. This occurs at the output of adders and can be prevented by scaling the inputs to the adders in such a way that the outputs are kept low. However, this reduces the signal to noise ratio and increases the cost.
- In digital filter, the product of two variables requires very long bits. For recursive filters, if it is not reduced, subsequent computations will cause the number of bits to grow without limit. Truncation or rounding is used to quantize the products to

the permissible word length. When the products are quantized, roundoff errors occur and they may lead to oscillations in the output even when there is no input. Further the signal to noise ratio is also small. The roundoff noise can be reduced or minimized by passing through subsequent sections of a cascade structure where it is alternated.

- Reduction in signal to noise ratio due to roundoff error can be offset by the use of error spectral shaping scheme. However this scheme increases the number of multiplications and additions but computationally more efficient.

Short Questions and Answers

1. **What is meant by finite word length effects in digital filters?**
   The fundamental operations in digital filters are multiplication and addition. When these operations are performed in a digital system, the input data as well as the product and sum (output data) have to be represented in finite word length, which depends on the size (length) of the register used to store the data. In digital computation the input and output data (sum and product) are quantized by rounding or truncation to convert them into finite word size. This creates error (in noise) in the output or creates oscillations (limit cycles) in the output. These effects due to finite precision representation of numbers in digital system are called as finite word length effects.
2. **List some of the finite word length effects in digital filters.**
   1. Errors due to quantization of the input data.
   2. Errors due to quantization of the filter coefficients.
   3. Errors due to rounding the product in multiplications.
   4. Limit cycles due to product quantization and overflow in addition.
3. **Explain the fixed point representation of binary numbers.**
   In fixed point representation of binary numbers in a given word size, the bits allotted for integer part and fraction part of the numbers are fixed and therefore the position of binary point is fixed. The most significant bit is used to represent the sign of the number. This is shown in Figure 5.22.

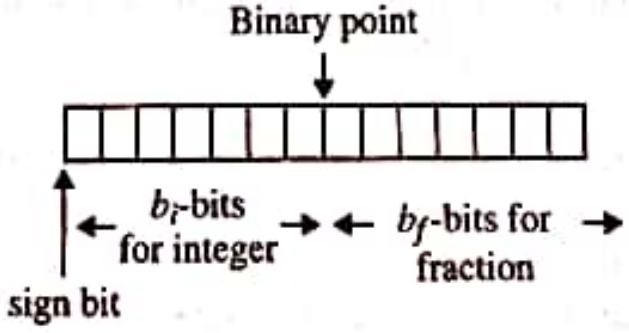


Figure 5.22

4. **What are the different formats of fixed point representation?**
In fixed point representation, there are three different formats for representing binary numbers:

1. Sign – magnitude format.
2. One's – complement format.
3. Two's – complement format.

In all the three formats, the positive number is same but they differ only in representing negative numbers.

5. **Explain the floating point representation of binary numbers.**
The floating numbers will have a mantissa part and exponent part. In a given word size the bits allotted for mantissa and exponent are fixed. The mantissa is used to represent a binary fraction number and the exponent is a positive or negative binary integer. The value of the exponent can be adjusted to move the position of the binary point in mantissa. Hence, this representation is called floating point. The floating point number is expressed as

$$\text{Floating point number, } N_f = M \times 2^E$$

where M = mantissa and E = exponent.

6. **Give the IEEE-754 standard format for 32 bit floating point numbers.**
The IEEE-754 standard for 32 bit single precision floating point number is given by floating point numbers, $N_f = (-1) \times 2^{E-127} \times M$. This is shown in Figure 5.23.

$S = 1$ bit field for sign of number.
$E = 8$ bit field for exponent.
$M = 23$ bit field for mantissa.

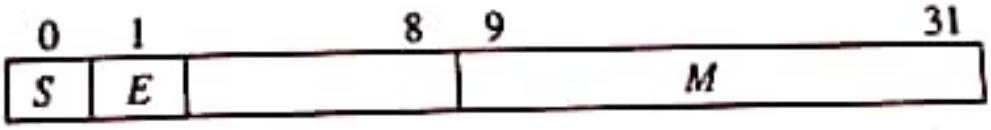


Figure 5.23

*7. **What are the types of arithmetic used in digital computers?**
The floating point arithmetic and two's complement arithmetic are the two types of arithmetic employed in digital systems.

8. **Compare the fixed point and floating point number representations.**

Fixed Point Representation	Floating Point Representation
1. In a b bit binary the range of numbers represented is less when compared to floating point representation.	In a b bit binary the range of the numbers represented is large when compared to fixed point representation.
2. The position of binary point is fixed.	The position of binary point is variable.
3. The resolution is uniform throughout the range.	The resolution is variable.

9. **Compare the fixed point and floating point arithmetic.**

Fixed Point Arithmetic	Floating Point Arithmetic
1. The accuracy of the result is less due to smaller dynamic range.	The accuracy of the result will be higher due to larger dynamic range.
2. Speed of processing is high.	Speed of processing is low.
3. Hardware implementation is cheaper.	Hardware implementation is costlier.
4. Fixed point arithmetic can be used for real time computation.	Floating point arithmetic cannot be used for real time computation.
5. Quantization error occurs only in multiplication.	Quantization error occurs in both multiplication and addition.

10. **What are the two types of quantization employed in digital system?**
The two type of quantization in digital system are truncation and rounding.

11. **What is Truncation?**
The truncation is the process of reducing the size of binary number by discarding all bits less significant than the least significant bit that is retained. (In truncation of a binary number to b bits all the less significant bits beyond b^{th} bit are discarded).

12. **Sketch the characteristics of the quantizer used for truncation.**

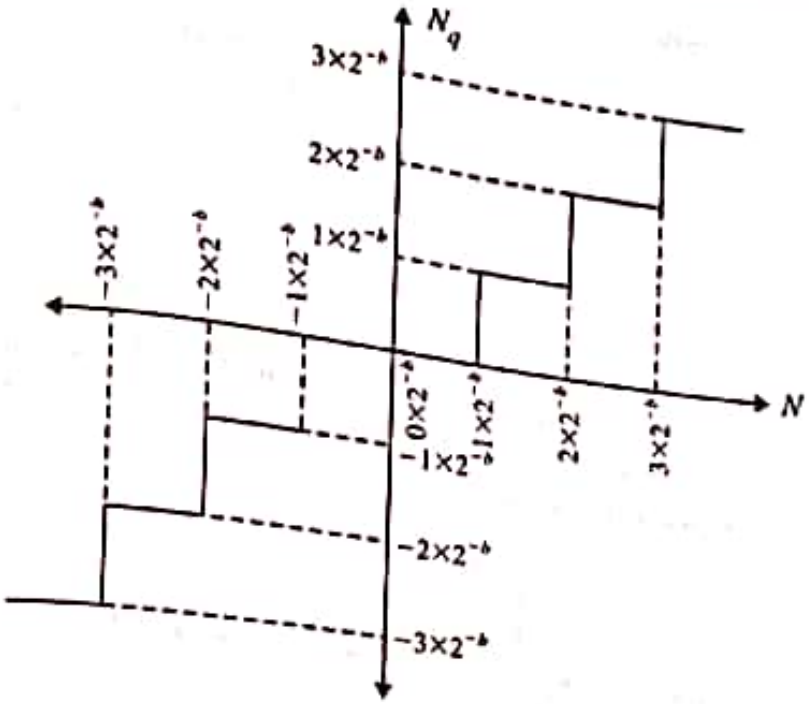
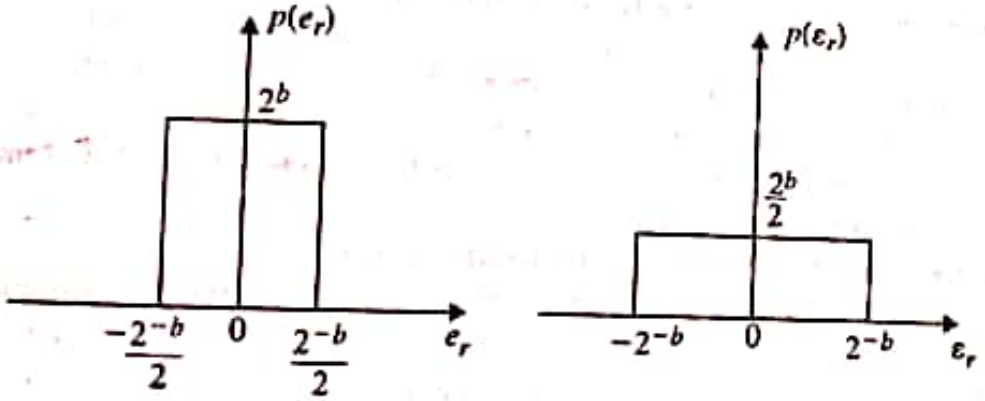


Figure 5.24

13. **What is rounding?**
Rounding is the process of reducing the size of a binary number to finite word sizes of b bits such that, the rounded b bit number is closest to the original unquantized number.

14. **Sketch the noise probability density function for rounding.**



Rounding-fixed point Rounding-floating point

Figure 5.25

15. **What are the errors generated by A/D process?**
The A/D process generates two types of errors. They are quantization error and saturation error. The quantization error is due to representation of the sampled signal by a fixed number of digital level (quantization levels). The

saturation error occurs when the analog signal exceeds the dynamic range of A/D converter.

16. **What is quantization step size?**
In digital systems, the numbers are represented in binary. With b bit binary we can generate 2^b different binary codes. Any range of analog value to be represented in binary should be divided into 2^b levels with equal increment. The 2^b levels are called quantization levels and the increment in each level is called quantization step size. If R is the range of analog signal then,

$$\text{Quantization step size, } q = \frac{R}{2^b}$$

17. **How the input quantization noise is represented in LTI system?**
The quantized input signal of a digital system can be represented as a sum of unquantized signal $x(n)$ and error signal $e(n)$ as shown in Figure 5.26 below.

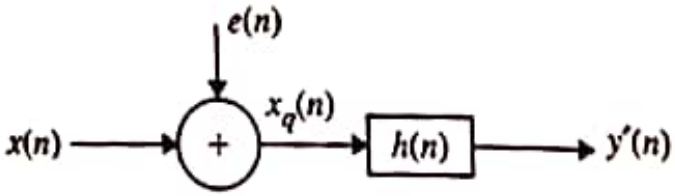


Figure 5.26

18. **What is steady state output noise due to input quantization?**
The input signal to digital system can be considered as a sum of unquantized signal and error signal due to input quantization. The response of the system can be expressed as a summation of response due to unquantized input and error signal. The response of the system due to error signal is given by convolution of error signal and impulse response. The variance of the response of the system for error signal is called steady state output noise power.

19. **How the digital filter is affected by quantization of filter coefficients?**
The quantization of filter coefficients will modify the values of poles and zeros and so the location of poles and zeros will be shifted from the desired location. This will create deviation in the frequency response of the system. Hence, the resultant filter will have a frequency response different from that of the filter with unquantized coefficients.

20. **What is meant by product quantization error?**
In digital computation, the output of the multiplier *i.e.*, the products are quantized to the finite word length in order to store them in registers and to be used in subsequent calculation. The error due to the quantization of the output of multiplier is referred to as product quantization error.

21. **Why rounding is preferred for quantizing the product?**
In digital system the product quantization is performed by rounding due to the following desirable characteristics of rounding.

(i) The rounding error is independent of the type of arithmetic.
(ii) The mean value of rounding error signal is zero.
(iii) The variance of the rounding error signal is least.

22. **Define noise transfer function (NTF).**
The noise transfer function (NTF) is defined as the transfer function from the noise source to the filter output. The NTF depends on the structure of the digital network.

23. **Draw the statistical model of the fixed point product quantization.**
The multiplier is considered as an infinite precision multiplier. Using an adder the error signal is added to the output of the multiplier so that the output of the adder is equal to the quantized product.

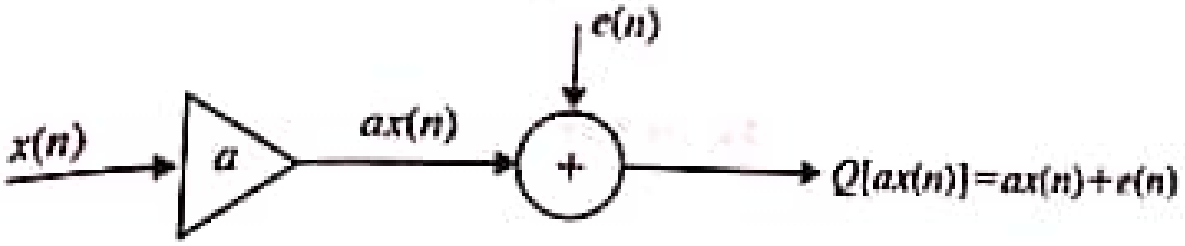


Figure 5.27

26. **What are limit cycles?**
In recursive systems when the input is zero or some non-zero constant value, the non-linearities due to finite precision arithmetic operations may cause periodic oscillations in the output. These oscillations are called limit cycles.

27. **What are two types of limit cycles?**
The two types of limit cycles are zero input limit cycle and overflow limit cycle.

28. **What is zero Input limit cycle?**
In recursive system, the product quantization may create periodic oscillations in the output. These oscillations are called limit cycles. If the system output enters a limit cycle, it will continue to remain in limit cycle even when the input is made zero. Hence, these limit cycles are also called zero input limit cycles.

29. **What is dead band?**
In a limit cycle the amplitudes of the output are confined to a range of values, which is called dead band of a filter.

30. **How the system output can be brought out of the limit cycle?**
The system output can be brought out of limit cycle by applying an input of large magnitude, which is sufficient to drive the system out of limit cycle.

31. **Draw the transfer characteristics of two's complement adder?**

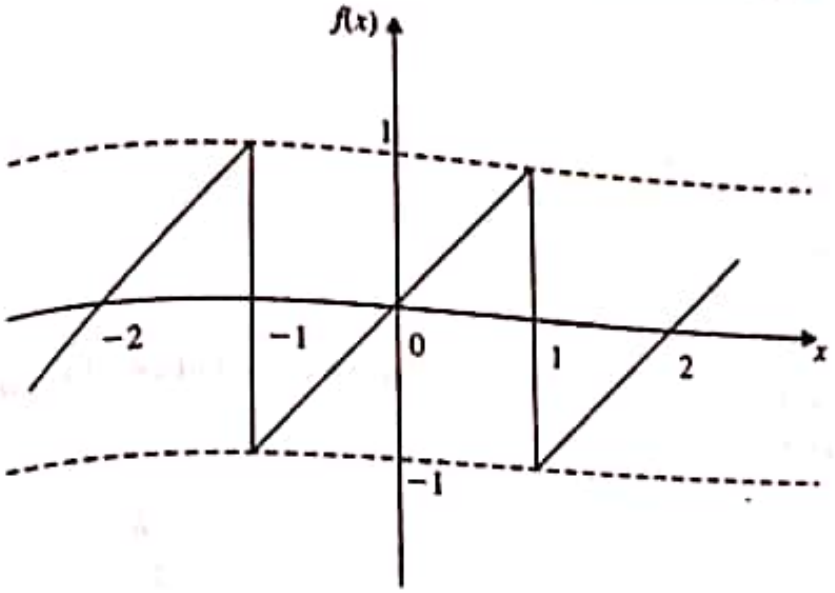


Figure 5.30

32. **What is saturation arithmetic?**
In saturation arithmetic when the result of arithmetic operations exceeds the dynamic range of number system, then the result is set to maximum or minimum possible value. If the upper limit is exceeded, then the result is set to maximum possible value. If the lower limit is exceeded, then the result is set to minimum possible value.

33. **What is overflow limit cycle?**
In fixed point addition the overflow occurs when the sum exceeds the finite word length of the register used to store the sum. The overflow in addition may lead to oscillations in the output which is called overflow limit cycle.

34. **How overflow limit cycles can be eliminated?**
The overflow limit cycles can be eliminated either by using saturation arithmetic or by scaling the input signal to the adder.

35. **What is the drawback in saturation arithmetic?**
The saturation arithmetic introduces non-linearity in the adder which leads to signal distortion.

36. **Give the rounding errors for fixed and floating print arithmetic.**

For fixed point arithmetic

$$\text{Rounding error, } e_r = N_r - N$$

where N_r is the fixed point binary number quantized by rounding and N is the unquantized fixed point binary number.

The range of error due to rounding is

$$-\frac{2^{-b}}{2} \le e_r \le \frac{2^{-b}}{2}$$

For floating point arithmetic

$$\text{Rounding error, } E_r = \frac{N_{rf} - N_f}{N_f}$$

where N_{rf} is rounded floating point binary number and N_f unquantized floating point binary number.

The range of error due to rounding is

$$-2^{-b} \le e_r \le 2^{-b}$$

37. What is the steady state noise power at the output of an LTI system due to the quantization at the input to L bits?

$$\text{Quantization step } (q) = \frac{R}{2^{-b}} = \frac{R}{2^{-L}}$$

$$\text{Steady state input noise power } \sigma_e^2 = \frac{q^2}{12} = \frac{(R/2^{-L})^2}{12}$$

$$\text{Steady state noise power at the output is } \sigma_{e0}^2 = \sigma_e^2 \sum_{i=1}^{N} \text{Re}[H(z)H(z^{-1})z^{-1}]_{z=p_i}$$

38. What are the three quantization errors due to finite word length, registers in digital filters?

1. Input quantization error.
2. Coefficient quantization error.
3. Product quantization error.

39. Write the 2's complement of the following (*a*) +7 and (*b*) −7

$$(a) + 7_{10} = (0111)_2$$

$$(b) - 7_{10} = 1\text{'s complement} \Rightarrow 1000$$

$$= 2\text{'s complement} \Rightarrow 1001$$

$$-7_{10} = (1001)_2$$

40. What is the steady state noise power due to quantization if the number of bits is b?

Steady state noise power

$$\sigma_e^2 = \frac{2^{-2b}}{12}$$

where b is the number of bits excluding sign bit.

41. Why rounding is preferred to truncation in realizing digital filter?

Rounding is preferred to truncation due to the following desirable characteristics of rounding.

1. The rounding error is independent of the type of arithmetic
2. The mean value of rounding error signal is zero
3. The variance of rounding error signal is least

42. Express the fraction $(-7/32)_{10}$ in signed magnitude and two's complement notation using 6 bits?

$$\left(\frac{-7}{32}\right)_{10} = -(0.21875)_{10}$$

$$\text{signed magnitude: } \left(\frac{-7}{32}\right)_{10} = (1.0011)_2$$

$$\text{two's complement: } \left(\frac{-7}{32}\right)_{10} = (1.11001)_2$$

43. Identify the various factors which degrade the performance of the digital filter implementation when finite word length is used.

1. Error due to quantization of the input data.
2. Error due to quantization of the filter coefficients.
3. Error due to rounding the product in multiplication.
4. Limit cycles due to product quantization and overflow in addition.

44. Express the fraction (7/8) and (−7/8) in sign magnitude and two's complement and 1's complement.

$$\text{Fraction: } \left(\frac{7}{8}\right)_{10} = (0.111)_2 \text{ in sign magnitude}$$

$$= (0.111)_2 \text{ is 1's and 2's complement}$$

$$\text{Fraction: } \left(-\frac{7}{8}\right)_{10} = (1.111)_2 \text{ in sign magnitude}$$

$$= (1.000)_2 \text{ in 1's complement}$$

$$= (1.001)_2 \text{ in 2's complement}$$

45. What are the different quantization methods?

The common methods of quantization are (1) Truncation and (2) Rounding.