filter.

Major factors that influence the choice of a specific realization:

Computational Complexity, memory requirements, and finite word length effects.

Computational Complexity:

It refers to the number of arithmetic operations (multiplication, division & addition) required to compute an output value y(n) for the system.

Note: factors considered during present: number of times a fetch from memory is performed, no. of times a comparision between two numbers is performed per output sample).

Memory Requirement:

It refers to the number of memory locations required to store the system parameters, past input, past output & any intermediate computed values.

Finite word length effect (or) finite precision effect:

* The parameters of the system must be represented with finite precision of the computer (or) hardware.
* The computation that are performed in the process of computing an output from the system must be rounded off (or) truncated to fit within the limited precision constraint of hardware.

is whether we are going to perform fixed point (or) floating point arithmetic.

Representation of Numbers:

* We consider representation of numbers for digital computation.
* The main characteristic of digital arithmetic is the limited number of digits used to represent numbers.
* This constraint leads to finite numerical precision in computation, which leads to round off errors & non-linear effects in the performance of digital filters.

Fixed Point Representation of Numbers:

* In the decimal representation of number, the digits to the left of the decimal point represent the integer part of the number, and the digits to the right of the decimal point represent the fractional part of the number.

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

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

where b_i – digits.
r – radix (or) base
A – no. of integer digits.
B – no. of fractional digits.

eg. $(123.45)_{10} = 1\times10^2 + 2\times10^1 + 3\times10^0 + 4\times10^{-1} + 5\times10^{-2}$

$(101.01)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

* Binary representation:

⇒ $r = 2$, digits $\{b_i\}$ are called binary digits (or) bits. & take values $\{0, 1\}$.

⇒ * binary digit b_A is (MSB) most significant bit of the number.

⇒ binary digit b_B is (LSB) Least significant bit.

Unsigned integer format:

* consider n-bit integer format

⇒ $A = n-1$, $B = 0$.

⇒ Magnitude range: 0 to $2^n - 1$.

Fraction format:

⇒ $A = 0$, $B = n-1$.

⇒ Range of numbers: 0 to $1 - 2^{-n}$.

Representation of signed Binary fraction:

There are three ways to represent negative numbers. The format for positive fraction is the same in all three formats.

$$X = 0.b_1 b_2 b_3 \cdots b_B = \sum_{i=1}^{B} b_i 2^{-i} \quad X \geq 0$$

MSB, $b_0 = 0$ to represent positive sign.

⇒ Negative fraction:

$$X = -0.b_1 b_2 \cdots b_B = -\sum_{i=1}^{B} b_i 2^{-i}$$

(a) Sign magnitude format:

⇒ MSB is set to 1 to represent negative sign.

$$X_{SM} = 1.b_1 b_2 \cdots b_B, \quad \text{for} \quad X \leq 0$$

(b) One's Complement format:

$$X_{1C} = 1.\bar{b}_1 \bar{b}_2 \cdots \bar{b}_B, \quad X \leq 0$$

$\bar{b}_i = 1 - b_i$, which is one's complement of b_i

(c) Two's Complement format:

* The two's complement is formed by complementing the positive number and adding one LSB.

$$X_{2C} = 1.\bar{b}_1 \bar{b}_2 \bar{b}_3 \cdots \bar{b}_B + 00\ldots01, \quad X < 0$$

where + represents modulo-2 addition that ignores any carry generated from the sign bit.

eg: $-3/8$

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

Representing $3/8$ in digital format = 0011

Two's complement = 1's complement of 0011 = 1100

adding 0001 0001

1101

∴ 2's complement of $-3/8$ = 1101.

⇒ With two's-complement arithmetic, range of (B+1) bit number is from -1 to $1 - 2^{-B}$.

Binary Floating Point Representation of Numbers:

⇒ In Fixed point representation, to cover a range of numbers $(x_{max} - x_{min})$

Resolution, $\boxed{\Delta = \frac{x_{max} - x_{min}}{m-1}}$

⇒ $m = 2^b$, specifies no. of levels

⇒ b = no of bits.

Note:

⇒ In fixed point representation, resolution is fixed.

Floating Point Representation:

⇒ The Mantissa M is the fractional part of the number & falls in the range $\frac{1}{2} \leq M \leq 1$.

⇒ The exponential factor is 2^E where Exponent E is positive (or) negative integer.

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

⇒ eg: $X_1 = 5$ has

Mantissa $M_1 = 0.101000$

Exponent $E_1 = 011$

$X_2 = 3/$

Mantissa. $M_2 = 0.110000$

Exponent $E_2 = 101$

(Left most bit in exponent represents sign bit)

⇒ Multiplying two numbers:

$X_1 = M_1 2^{E_1}$. $X_2 = M_2 2^{E_2}$

$$\boxed{X_1 X_2 = (M_1 M_2) \cdot 2^{(E_1 + E_2)}}$$

Adding two numbers:

$X_1 = M_1 \cdot 2^{E_1}$ $\quad X_2 = M_2 \cdot 2^{E_2}$

$E_1 = E_2$] then $\boxed{X_1 + X_2 = (M_1 + M_2)\, 2^{E_1}}$

Note:

floating point representation provides larger dynamic range.

Comparing fixed point & floating point Representation:

* A computer has word size 32 bit.

Fixed point Representation:

=> range : $-(2^{31}-1)$ to $+(2^{31}-1)$

resolution: 1

=> resolution: 2^{-10}.

Dynamic range: $-(2^{21}-2^{-10})$ to $(2^{21}-2^{-10})$

Floating point Representation:

=> Mantissa: 23 bits plus sign bit.

Exponent: 7 bit + sign bit.

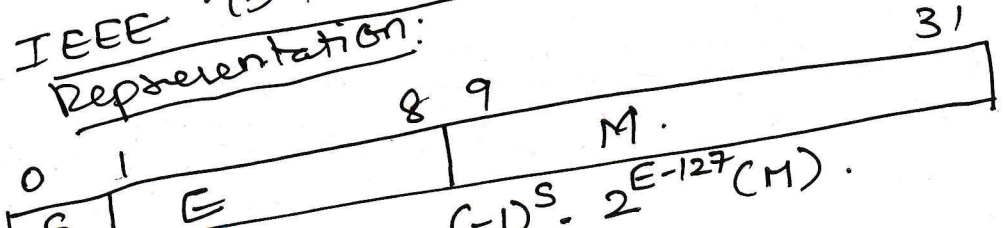
Smallest number: Sign (23 bit) 0.100...0, sign (7 bits) 1111111 $= \frac{1}{2} \times 2^{-127}$

$= 0.3 \times 10^{-38}$

Largest number: Sign (23 bits) 0.111...1, sign (7 bits) 0, 1111111 $= (1-2^{-23}) \times 2^{127}$

$\simeq 1.7 \times 10^{38}$.

=> IEEE 754 standard for floating point Representation:



Problems in finite word length effects:

① Parameter Quantization in digital filter
② Round off noise in multiplication
③ Overflow in addition
④ Limit cycles.

Errors Resulting from Rounding & Truncation:

Quantization Noise:

* The numeric equivalent of each sample $x(n)$ is expressed by a finite number of bits giving the sequence $x_q(n)$.
* The difference signal $e(n) = x_q(n) - x(n)$ is called quantization noise (or) A/D conversion noise.
* If n bits are available, the number of levels available for quantizing $x(n)$ is 2^{b+1}.
* Interval between successive levels, $q = \frac{2}{2^{b+1}} = 2^{-b}$

Truncation:

⇒ Discarding all bits less significant than LSB.

eg: Truncation from 8 bits to 4bits.

8bit: 0.00110011

4bit: 0.0011

Rounding:

Rounding of a number by bits is done by choosing a 'b' bit number close to the original number.

(or) rounded down to 3 bits as 0.110.

Error due to truncation & Rounding:

* If x_T is the truncated value of x and b bits are used after truncation.

$$0 \geq x_T - x > -2^{-b} \qquad [x_T < x]$$

* In one's complement representation & sign magnitude representation,

$$0 \leq x_T - x < 2^{-b}$$

⇒ In floating point Representation

* The Mantissa is truncated to N bits.

Error, $e = x_T - x = 2^c (M_T - M)$

* In two's complement representation of mantissa,

$$0 \geq M_T - M > -2^{-b}$$

$$0 \geq e > -2^{-b} 2^c \longrightarrow (1)$$

⇒ relative error, $\varepsilon = \frac{x_T - x}{x} = \frac{e}{x}$.

$$(1) \Rightarrow \quad 0 \geq \varepsilon x > -2^{-b} 2^c \qquad (\because x = 2^c M)$$

$$0 \geq \varepsilon 2^c M > -2^{-b} 2^c$$

$$0 \geq \varepsilon M > -2^{-b}$$

if $M = 1/2$, the relative error is maximum

$$0 \geq \varepsilon > -2^{-b}$$

⇒ In fixed point arithmetic, the error due to rounding a number to b bits produces an error, $e = x_R - x$

$$-\frac{2^{-b}}{2} \leq x_R - x \leq \frac{2^{-b}}{2}$$

⇒ In floating point arithmetic, for rounding

$$-\frac{2^{-b}}{2} \leq M_R - M \leq \frac{2^{-b}}{2}$$

(where M_R & M are ~~truncated~~ rounded normal mantissa)

Input Quantization Error:

* Quantization error arises when a continuous signal is converted into digital value.

⇒ Quantization error is given by

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

where, $x_q(n)$ = sampled quantized value
$x(n)$ = sampled unquantized value.

Note:

a) If rounding is used to get $x_q(n)$ then

$$-\frac{q}{2} \leq e(n) \leq \frac{q}{2}$$

(If b bits are used for rounding then $q = 2^{-b}$)

for eg: unquantized signal $x(n) = (0.70)_{10} = (0.1011001111)_2$ ↓ Since '1' rounded to next value

* After rounding to 3 bits, sampled quantized value, $x_q(n) = 0.101$

$$\begin{array}{r} 0.101 \\ +\ 1 \\ \hline 0.110 \end{array}$$

$e(n) = 0.75 - 0.70$
$= 0.05.$

⇒ b) If truncation is used to get quantized sample value, the signal is represented by highest quantization level that is not greater than the signal.

$$-q \leq e(n) < 0 \qquad q = 2^{-b}$$

⇒ Quantization error mean value for rounding is '0' & two's complement truncation is $\frac{-q}{2}$.

———

Zero Input Limit Cycle oscillation

When a stable IIR digital filter is excited by a finite input sequence, that is constant, the output will ideally decay to zero. The nonlinearities due to the finite precision arithmetic operation often cause periodic oscillation to occur in the output. Such oscillations in recursive systems are called zero input limit cycle oscillation.

eg: Consider first order IIR filter with difference equation:

$$y(n) = x(n) + \alpha y(n-1)$$

$\Rightarrow \alpha = 1/2$, Data register length = 3 bit + sign bit.

Let $x(n) = \begin{cases} 0.875 & \text{for } n=0 \\ 0 & \text{otherwise} \end{cases}$

Quantization to 3 bits.

n	x(n)	y(n-1)	αy(n-1)	Q[αy(n-1)]	y(n) = x(n) + Q[α
0	0.875	0.0	0.0	0.000	7/8
1	0	7/8	7/16	0.100	1/2
2	0	1/2	1/4	0.010	1/4
3	0	1/4	1/8	0.001	1/8
4	0	1/8	1/16	0.001	1/8
5	0	1/8	1/16	0.001	1/8

$\Rightarrow$ From table we infer that for n ≥ 3 the output remains constant & gives 1/8 as steady output causing limit cycle oscillation.

Dead band:

⇒ The limit cycles occur as a result of quantization effect in multiplication.

⇒ The amplitude of the output during a limit cycle are confined to a range of values that is called dead band of filter.

Eg: consider difference equation of a IIR filter.

$y(n) = x(n) + \alpha y(n-1)$, $n > 0$.

⇒ After rounding the product term

$$y_p(n) = Q[\alpha y(n-1)] + x(n).$$

$$Q[\alpha y(n-1)] = y(n-1) \text{ for } \alpha > 0$$
$$= -y(n-1) \text{ for } \alpha < 0 \quad \rightarrow (1)$$

during limit cycle oscillation.

* By definition of rounding:

$$|Q[\alpha y(n-1)] - \alpha y(n-1)| \leq \frac{2^{-b}}{2} \rightarrow (2)$$

Sub (1) in (2)

$$|y(n-1)| - |\alpha y(n-1)| \leq \frac{2^{-b}}{2}$$

$$\therefore y(n-1) \leq \frac{\frac{1}{2}2^{-b}}{1-|\alpha|}$$

The above equation defines the dead band of the filter.

Problem:
Explain limit cycle oscillation for $y(n) = 0.95y(n-1) + x(n)$ & determine dead band of filter.

Soln:

$y(n) = 0.95y(n-1) + x(n)$, Let quantization be done to 3 bits.

Let $x(n) = \begin{cases} 0.5, & n=0 \\ 0, & \text{otherwise} \end{cases}$

n	x(n)	y(n-1)	0.95y(n-1)	Q[αy(n-1)]	y(n) = x(n)+Q[αy(n-1)]
0	0.5	0.0	0.0	0.000	0.5
1	0	0.5	0.475	0.100	0.5
2	0	0.5	0.475	0.100	0.5
3	0	0.5	0.475	0.100	0.5

$\therefore$ Dead band of the filter = 0.5

(0.5 is the amplitude during limit cycle).

Coefficient Quantization Error:

In the design of digital filter, the coefficients are evaluated with infinite precision. When the coefficients are quantized, the frequency response of the ~~actual~~ quantized filter deviates from actual one.

eg: Given $H(z) = \frac{1}{z - 0.752352}$ consider word length = 10.

$(0.752352)_{10} = (0.1100001000)_2$

$(-0.752352)_{10} =$

$\therefore$ After truncation, $H'(z) = \frac{1}{z-0.75}$

(ii) 6 Bit word length: ~~(optimistic) 6th bit is 0~~

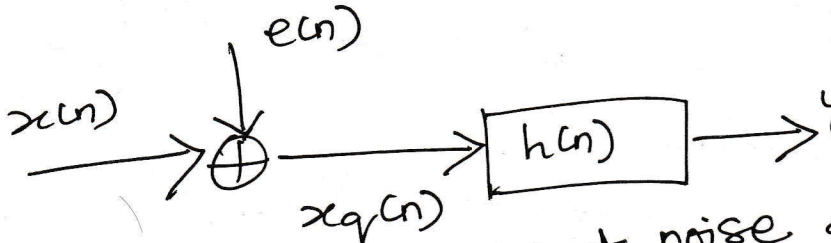
$(-0.752352)_{10} = (1.110000)_2 = -0.75$

$\therefore$ After truncation, $H'(z) = \frac{1}{z-0.75}$

x ——————— y.

Steady State Output Noise power:

Representation of A/D conversion noise:



$\Rightarrow$ Let $\varepsilon(n)$ be output noise due to quantization of input

$\Rightarrow \varepsilon(n) = e(n) * h(n)$

$= \sum_{k=0}^{n} h(k)\, e(n-k).$

$\Rightarrow$ The variance of the output

$$\sigma_\varepsilon^2(n) = \sigma_e^2 \sum_{n=0}^{k} h^2(n)$$

$\Rightarrow$ steady state variance,

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

$$\Rightarrow \sigma_\varepsilon^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n) = \frac{\sigma_e^2}{2\pi j} \oint_e H(z) H(z^{-1}) z^{-1} dz.$$

Problem:

The input to the system

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

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 bit (b) 16 bit.

Soln:

$\Rightarrow y(n) = 0.999\, y(n-1) + x(n)$.

$\Rightarrow$ Taking z- transform

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

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

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

* Taking inverse z-transform

$$h(n) = (0.999)^n u(n)$$

* quantization noise power at the output of the filter:

$$= \sigma_e^2 \sum_{k=0}^{\infty} (0.999)^{2k} \qquad \left(\because \sum_{n=0}^{\infty} a^n = \frac{1}{1-a}\right)$$

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

$$= \sigma_e^2 (500.25) \qquad \left(\because \sigma_e^2 = \frac{2^{-2b}}{12}\right)$$

$$\therefore \boxed{\sigma_\varepsilon^2 = \frac{2^{-2b}}{12} 500.25}$$

(a) using sign bit

$b+1 = 8$ bit for 8 bit word length

$b = 7$

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

(b) using sign bit

$b+1 = 16$ bit for 16 bit word length.

$b = 15$

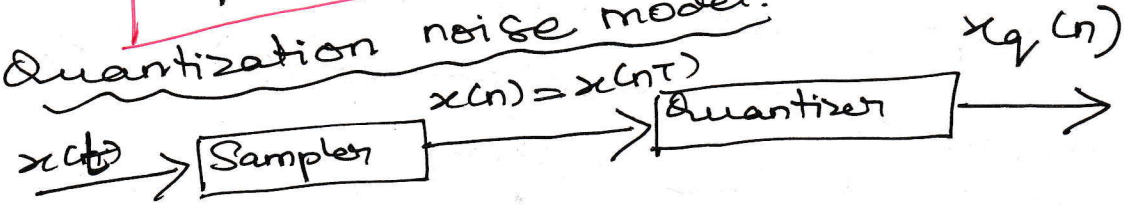
$$\sigma_\varepsilon^2 = \frac{2^{-30}}{12}(500.25) = 3.882 \times 10^{-8}$$

Steady State Input Noise power.

Let $e(n)$ be error due to quantization,

$$\boxed{x_q(n) = x(n) + e(n)}$$

Quantization noise model:



e(n)

* If rounding is used for quantization then quantization error $e(n) = x_q(n) - x(n)$

$$\text{if } -\frac{q}{2} \le e(n) \le \frac{q}{2} \quad (q = 2^{-b})$$

* Properties satisfied by e(n)

① e(n) is a sample sequence of stationary random process.

② e(n) is uncorrelated with x(n)

③ e(n) is a white noise process with uniform amplitude probability distribution over the range of quantization error.

* Variance of e(n): (a) Rounding:

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

$$E[e^2(n)] = \int_{-\infty}^{\infty} e^2(n) p(e)\, de$$

* $p(e) = 1/q$

* $E[e(n)] = 0$ (mean of e(n)).

$$\sigma_e^2 = \frac{1}{q} \int_{-q/2}^{q/2} e^2(n)\, de - 0 = \frac{1}{q}\left[\frac{e^3(n)}{3}\right]_{-q/2}^{q/2}$$

$$= \frac{1}{3q}\left[\frac{q^3}{8} + \frac{q^3}{8}\right] = \frac{2q^3}{3 \cdot q \cdot 8 \cdot 4} = \frac{q^2}{12}$$

$(\because q = 2^{-b})$

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

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

* $E[e(n)] = -\frac{q}{2}$

$$\sigma_e^2 = \int_{-q}^{0} e^2(n)\, p(e)\, de - \frac{q^2}{4}$$

$$= \frac{1}{q}\left[\frac{e^3(n)}{3}\right]_{-q}^{0} - \frac{q^2}{4} = \frac{q^3}{3q} - \frac{q^2}{4}$$

$$= \frac{q^2}{12}$$

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

σ_e^2 is also called steady state noise power due to input quantization.

$\Rightarrow$ Let $x(n)$ has variance σ_x^2.

Ratio of signal to noise power:

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

$$\therefore \text{SNR in dB} = 10\log_{10}\frac{\sigma_x^2}{\sigma_e^2} = 10\log\left[12(2^{2b})\sigma_x^2\right]$$

$$= 10\log 12 + 10\log 2^{2b} + 10\log_{10}\sigma_x^2$$

$$= \boxed{10.79 + 6.02b + 10\log_{10}\sigma_x^2}$$

if input is $Ax(n)$

$$\boxed{\text{SNR} = 10.79 + 6.02b + 10\log_{10}\sigma_x^2 + 20\log_{10}A}$$