Module 4

FINITE WORD LENGTH EFFECTS

Finite Word length

- The designing of a digital filter means figuring out its coefficients.
- We store the values of these coefficients in binary registers. These registers are just digital memories in the DSP system.
- Generally, we use infinite precision arithmetic for describing filter coefficients in the interest of accuracy.
- But practically, it is not possible to store all the bits in a register. Thus we need to find a way to pack these filter coefficients into a finite length register.
- So we use the fixed-point representation of binary numbers.

Quantization

- quantization is the process of reducing the number of bits to ensure the storage of the filter coefficients in the Digital Signal Processing system's register.
- In this post, we will study two types of Quantization methods:
  - Truncation
  - Rounding

What is Truncation?

- Truncation is a type of quantization where extra bits get ‘truncated.’
- Basically, in the truncation process, all bits less significant than the desired LSB (Least Significant Bit) are discarded.

Truncation-Example

- For example, suppose we wish to truncate the following 8-bit number to 4-bits.
  - X = 0.01101011 truncates to X = 0.0110
  - Converting the above to decimal we can see that there is a large change in value. (0.01101011 equals 0.418 and 0.0110 equals 0.375).

Thus, truncation is an poorer method of quantization since it has a high margin for error.

- The error from quantization using truncation is given by the formula:
- For a positive number/2s complement

$$-2^{-b} \leq e \leq 0$$

For a negative number/1s complement

$$0 \leq e \leq 2^{-b}$$

- **What is Rounding?**

1. **Rounding** is a quantization method where we '**round**-up' a particular number to the desired number of bits.
2. Basically, **rounding** is the process of reducing the size of a binary number to some desirable finite size. ...
3. Interestingly, the **rounding** process is a combination of **truncation** and addition.

- **Quantization error** is the difference between the analog signal and the closest available digital value at each sampling instant from the A/D converter.
- **Quantization error** also introduces noise, called **quantization** noise, to the sample signal.

Rounding Example

- Suppose we wish to truncate the following 8-bit number to 4-bits.
  - X = 0.01101011 truncates to X = 0.0110
  - Since the number next to the current LSB was 1, we add 1 to the current LSB.
  - Thus X is now 0.0111
  - Converting both the unquantized and rounded off numbers to decimal, we notice that the magnitude of error is less relative to truncation. (0.01101011 equals 0.418 and 0.0111 equals 0.438).
- Thus rounding is preferable than truncation.

Rounding error

- The magnitude of error in rounding is given by the formula:

$$\frac{-2^{-b}}{2} \leq e \leq \frac{2^{-b}}{2}$$

- What is the concept behind the quantization of filter coefficients?
- How to reduce the quantization effect on filter coefficients?
- Example of the effect of quantization on a filter's frequency response
  - Direct form realization
  - Cascade form realization

quantization of filter coefficients?

- DSP systems, we can say that the number of bits that we use in designing a filter is limited by the word length of the register used to the store them.

- The fact of the matter, however, is that most of the DSP systems that we use have a fixed number of bits in their registers. The capacity of registers is limited, practically. So how do we fit infinite arithmetic numbers in some finite space?
- Easy. We quantize them. Generally, we use quantization methods like rounding or truncating to quantize the filter coefficients to the word size of the register.

- The location of poles and zeros of any digital filter directly depends on the value of the filter coefficients. But since we are quantizing the values of the filter coefficients to fit them into the register, there will be a change in the values of the poles and zeros.

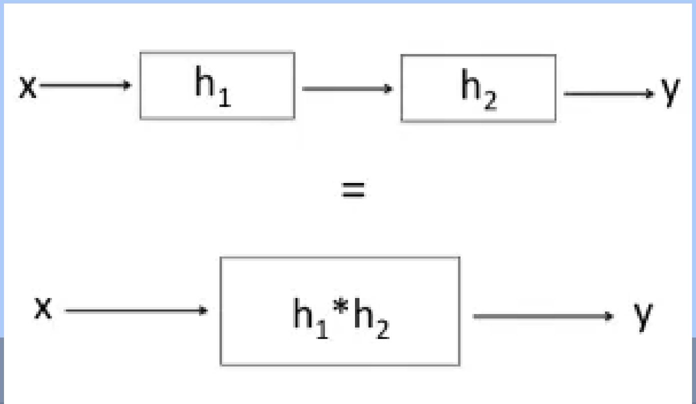
- This, in turn, causes the location of the poles and zeros to shift from the desired location. ***Thus the quantization of filter coefficients creates a deviation in the frequency response of the system.***

- In summary, after quantization, we get a filter that has a frequency response that is different from the frequency response of the filter with unquantized coefficients.

filter coefficients?

- We can minimize this drastic effect of quantization on the filter coefficients. The corresponding change in the frequency response can be minimized by realizing a filter with a large number of poles and zeros as an interconnection of second-order sections.
- That is, the physical realization of these filters can be done in a particular manner that reduced the effect of the quantization of filter coefficients.
- Spoiler! Coefficient quantization has less effect on cascade realization when compared to other realizations.

- That is, the physical realization of these filters can be done in a particular manner that reduced the effect of the quantization of filter coefficients.
- Spoiler! Coefficient quantization has less effect on cascade realization when compared to other realizations.



filter's frequency response

- Let's take up a transfer function of a random filter and realize it using direct and cascade forms. We'll arrive at the conclusion that the shifting of poles and zeros (i.e the frequency response) is closer to the ideally intended filter in the case of cascade realization.

- Consider a second-order filter of having a transfer function given by

$$\frac{1}{(1-0.9z^{-1})(1-0.8z^{-1})}$$

- **Direct form realization**
- We can rearrange the above transfer function to be written as
- H(z) = $$\frac{z^2}{(z-0.9)(z-0.8)}$$
- Thus, we c … poles of the system lie at P1 = 0.9 and P2 = 0.8

- Solving the brackets of the original form of the transfer function

$$\frac{1}{1-1.7z^{-1}+0.72z^{-2}}$$

- Let’s quantize the coefficients by truncating them to 3-bits.
- $1.7 \xrightarrow{Converting\ to\ Binary} 1.1011$
- $1.1011 \xrightarrow{Truncating\ to\ 3\ bits} 1.101$

$1.1\ 01 \xrightarrow{Converting\ to\ Decimal} 1.625$

-

- Let’s quantize the coefficients by truncating them to 3-bits.
- 0.72 $\xrightarrow{\text{Converting to Binary}}$ 0.1011
- 0.1011 $\xrightarrow{\text{Truncating to 3 bits}}$ 0.101

0.101 $\xrightarrow{\text{Converting to Decimal}}$ 0.625

- Let H'(z) be the transfer function after quantization of coefficients
- H'(z) =

$$\frac{1}{1-2.625z^{-1}+0.625z^{-2}}$$

- The new poles are at P1′ = 2.625 and P2′ = 0.625
- Thus we can see a huge shift in the position of the poles.

- **Cascade form realization**
- In the cascade realization method, the transfer function can be written as follows:
- $H(z) = H1(z) . H2(z)$
- $H1(z) =$
- $H2(z) =$
- Let’s quantize the coefficients by truncating them to 3-bits.
-

- 0.90. $\xrightarrow{\textit{Converting to Binary}}$ 0.1110

- 0.1110 $\xrightarrow{\textit{Truncating to 3 bits}}$ 0.111

- 0.111 $\xrightarrow{\textit{Converting to Decimal}}$ 0.875

1. **For the second order IIR filter, the system function is,**

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

Find the effect of shift in pole location with 3 bit coefficient representation in direct and cascade forms.(MAY- 2012).

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) is $P_1 = 0.5$ and $P_2 = 0.45$.

CASE 1: DIRECT FORM

$$H(Z) = \frac{1}{(1 - 0.5z^{-1})(1 - 0.45z^{-1})}$$
$$= \frac{1}{(1 - 0.95z^{-1} + 0.225z^{-2})}$$

Quantization of coefficient by truncation

convert to binary truncate to 3 bits convert to decimal

$\mathbf{0.95}_{10}$ 0.1111_2 0.111_2

0.225_{10}......... convert to binary **0.0011_2**

truncate to 3 **bits 0.001**

convert to decimal0**.125**$_{10}$

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

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

Case *(ii)* Cascade Form

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

Quantization of coefficient by truncation

$\mathbf{0.5}_{10}$ .... convert to binary 0.1000_2

0.1000_2..... truncate to 3 bits ...... 0.100_2

0.100_2..... convert to decimal ..... 0.5_{10}

- **Quantization of coefficient by truncation** convert to binary truncate to 3 bits convert to decimal
- **0.45_{10}** 0.0111_2 0.011_2 0.375_{10}
-
- $H(Z) = \frac{1}{(1-0.5z^{-1})(1-0.375z^{-1})}$
-
- 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.

Different errors due to quantization

1) Input quantization error.
2) Product quantization error.
3) Co-efficient quantization error.

Quantization error

Input quantization error:

- The conversion of a continuous time **input signal** into digital value produces an error, which is known as **input quantization error.**
- This error arises due to the representation of the input signal by a fixed number of digits in A/D conversion process.

Quantization error

- **Coefficient quantization error:**
- **The filter coefficients** are computed to infinite precision in theory.
- If they are quantized, the frequency response of the resulting filter may **differ from the desired response..**
- If the poles of the desired filter are close to the unit circle, then those of the filter with quantized coefficient may we outside the unit circle **leading to instability.**

Quantization error:

- **Product quantization error:**
- Product quantization error arise at the **output of a multiplier**. Multiplication of a b-bit data with a b-bit coefficient results a product having 2b bits.
- Since a b-bit register is used, the multiplier output must be rounded or truncated to b-bits which produced an error.

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 it is necessary to quantize the product to *b* bits, which produce an error known as **product quantization error or product round off noise.**

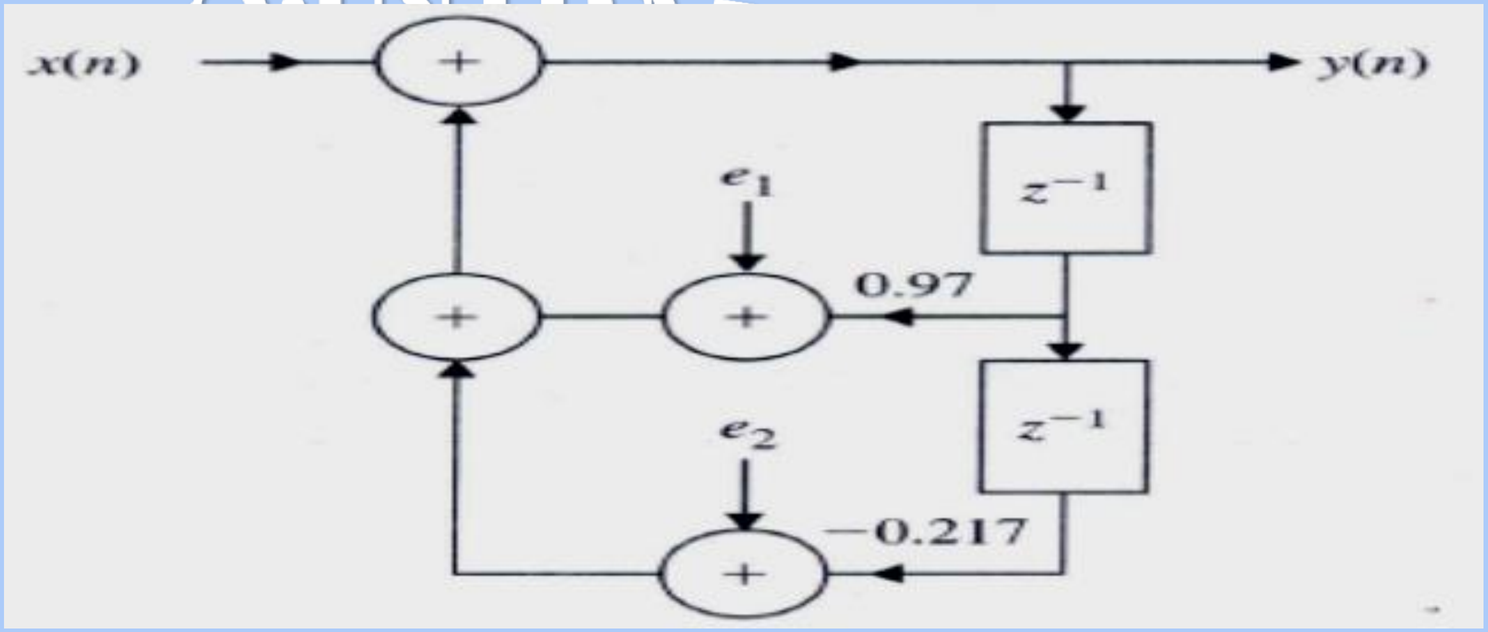
Product quantization error:

- 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 .

(next slide)

Product quantization error:

$$H(Z) = \frac{1}{(1 - 0.97z^{-1} + 0.217z^{-2})}$$



Product quantization error:

- The multiplication is modeled as an infinite precision multipliers followed by an adder where round off noise is added to the product so that overall result equals some quantization level.

Product quantization error:

- 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.

Product quantization error:

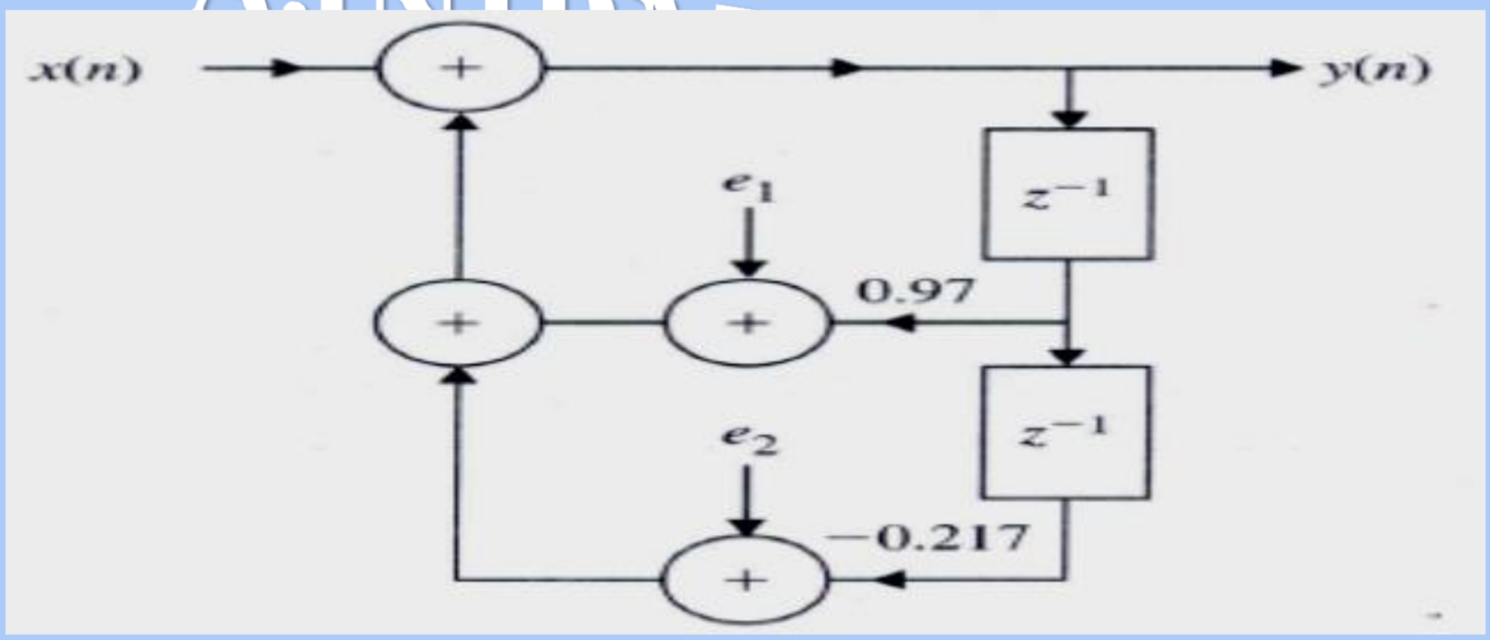
- **In general the following assumptions are made regarding the statistical independence of the various noise sources in the digital filter**.
- I. Any two different samples from the same noise source are uncorrelatcd.
- 2. Any two different noise source, when considered as random processes are uncorrelated.

Product quantization error:

- 3. Each noise source is uncorrelated with the input sequence.
- 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.
- Variance of k^{th} noise source $\sigma^2_{ek} = \frac{q^2}{12} = \frac{2^{-2b}}{3}$

Product quantization error:

$$H(Z) = \frac{1}{(1 - 0.97z^{-1} + 0.217z^{-2})}$$



PROBLEM 1

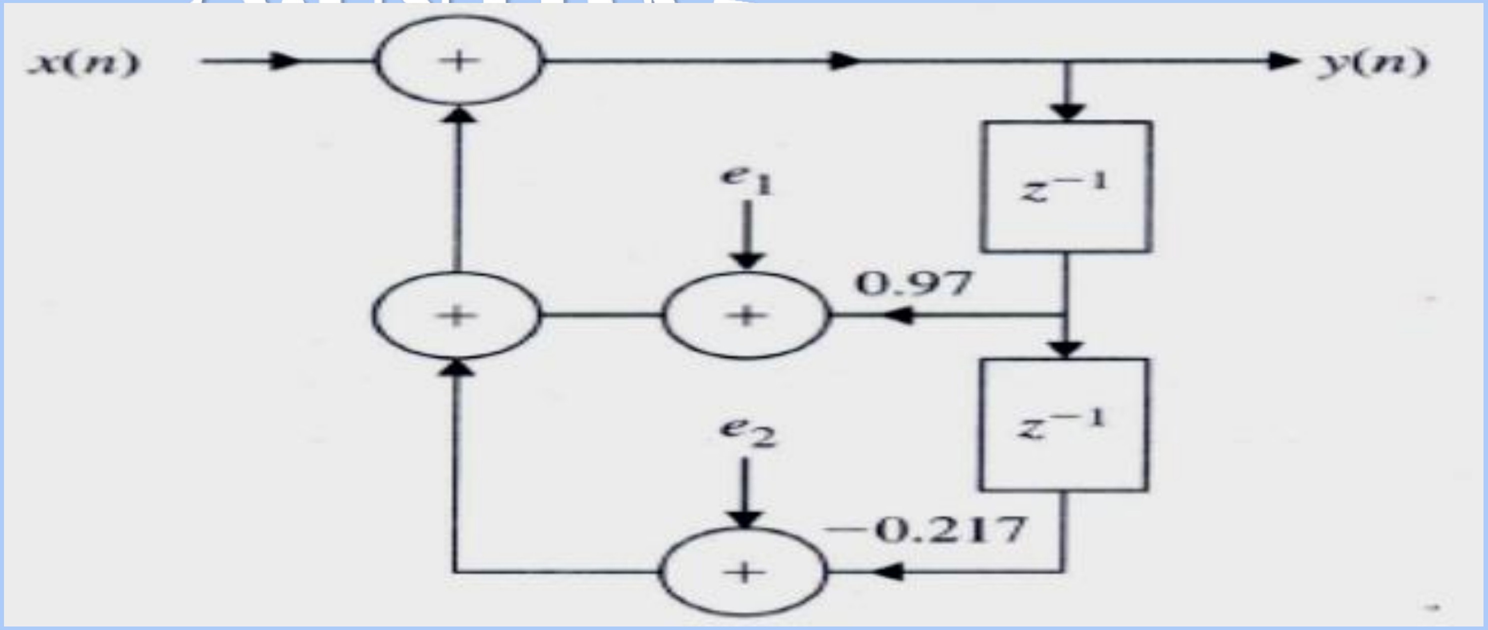
- In the IIR system given below the products are rounded to **4 bits (including sign bits).** The system function is
- $H(Z) = \frac{1}{(1-0.35z^{-1})(1-0.62z^{-1})}$
- Find the output roundoff noise power in *(a)* direct form realization and *(b)* cascade form realization.

Solution

- **<u>Direct Form Realization</u>**
- $H(Z) = \frac{1}{(1-0.35z^{-1})(1-0.62z^{-1})}$
- $H(Z) = \frac{1}{(1-0.97z^{-1}+0.217z^{-2})}$
- Direct form realization of *H (z)* is shown in Figure

Product quantization error:

$$H(Z) = \frac{1}{(1 - 0.97z^{-1} + 0.217z^{-2})}$$



Product quantization error:

- The variance of the error signal is,
- Here R is not given. So take $R = 2$ and $b = 4$ bits
- $q = \frac{R}{2^b} = \frac{2}{2^4} = \frac{1}{2^3} = \frac{1}{8}$
- $\sigma_e^2 = \frac{(\frac{1}{8})^2}{12} = \frac{q^2}{12}$
- $\sigma_e^2 = 1.3021X\ 10^{-3}$

Product quantization error:

Output noise power due to the noise signal $e_l\,(n)$ is,

$$\sigma_{e01}^2 = \frac{\sigma e^2}{2\pi j} \oint H(z)\ H(z^{-1})\, z^{-1} dz$$

Here,

$$H(Z) = \frac{1}{(1 - 0.35z^{-1})(1 - 0.62z^{-1})}$$

$$H(Z) = \frac{z^2}{(z - 0.35)(z - 0.62)}$$

Product quantization error:

Therefore,

$$\sigma_{e01}^2 = \frac{\sigma e^2}{2\pi j}\oint\left(\frac{z^2}{(z-0.35)(z-0.62)}\right)\left(\frac{z^{-2}}{(z^{-1}-0.35)\ (z^{-1}-0.62)}\right) z^{-1}dz$$

Product quantization error:

- $$\sigma_{e01}^2 = \frac{\sigma e^2}{2\pi j} \oint \left(\frac{z^{-1}}{(z-0.35)(z-0.62)\,(z^{-1}-0.35)\,(z^{-1}-0.62)} \right) dz$$
-
- **The stable poles of *H(z)* are P_1 = 0.35 and P_2= 0.62 and unstable poles of H(z) are P3 = 2.86 and P4 = 1.62. For taking residue only consider the stable poles.**

$$\mathrm{Res}[\mathrm{H}(\mathrm{z})\mathrm{H}(\mathrm{z}^{-1})\mathrm{z}^{-1}]|(z=0.35)=$$

$$=(z-0.35)\frac{z^{-1}}{(z-0.35)(z-0.62)\ (z^{-1}-0.35)\ (z^{-1}-0.62)}\Bigg|$$

At $z=0.35$ = **-1.8867.**

$$\mathrm{Res}[\mathrm{H(z)H(z^{-1})z^{-1}}]|(z = 0.62) =$$

$$= (z - 0.62) \frac{z^{-1}}{(z-0.35)(z-0.62)\,(z^{-1}-0.35)\,(z^{-1}-0.62)}\Bigg|$$

At $z = 0.62$

=4.7640.

Total = $\mathrm{Res}[\mathrm{H(z)H(z^{-1})z^{-1}}]|(z = 0.35) +$
$\mathrm{Res}[\mathrm{H(z)H(z^{-1})z^{-1}}]|(z = 0.62)$

$=$ **-1.8867 + 4.7640.**

= 2.8773.

Therefore,

$$\sigma_{e01}^2 = \frac{\sigma e^2}{2\pi j}\oint H(z)\ H(z^{-1})\ z^{-1}dz$$

$= 1.3021 \mathrm{X}\ 10^{-3} \mathrm{X}\ 2.8733$

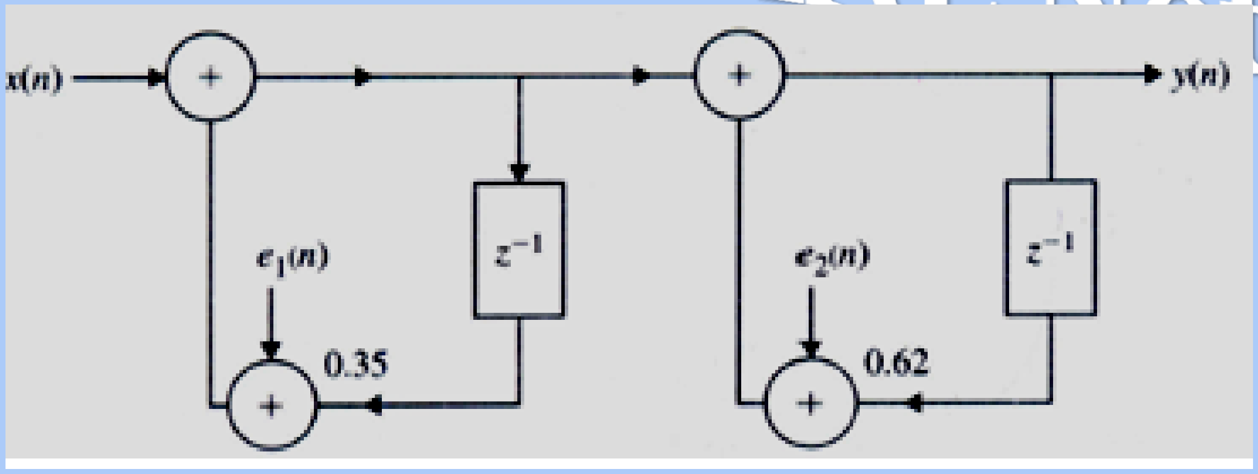
$= 3.7465 \mathrm{X}\ 10^{-3}$

- Here the output noise due to error source $e_2(n)$ is same as that of $e_1(n)$, *i.e.,*
- ***$e_2(n)$* noise power = noise power of *$e_1(n)$***
- $\sigma_{e01}^2 = \sigma_{e02}^2$
-
- $= 3.7465 \times 10^{-3}$
- **Total output noise power due to all the noise sources is,**
- $\boldsymbol{\sigma_{e0}^2 = \sigma_{e01}^2 + \sigma_{e02}^2}$
-
- $\sigma_{e0}^2 = 7.493\ X\ 10^{-3}$

Case *(ii)*

- $\underline{H(z) = H_1(Z)H_2(Z)}$
- The cascade form realization of *H(z)* is shown in

Figure



- The order of cascading is $H_1(Z)H_2(Z)$. Output noise power due to errorsignal $e_1(n)$ is
  - $\sigma_{e01}^2 = \frac{\sigma e^2}{2\pi j}\oint H(z)\ H(z^{-1})\ z^{-1}dz$
- $= 3.7465 \text{ X } 10^{-3}$[refer direct form]

- Output noise power due to the error, signal *e2(n)* is
- $\sigma_{e02}^2 = \frac{\sigma e^2}{2\pi j} \oint H_2(z) H_2(z^{-1}) z^{-1}\, dz$

$$H_2(z)H_2(z^{-1})z^{-1} = \frac{z^{-1}}{(z-0.62)(z^{-1}-0.62)}$$

$$\text{Res}[H_2(z)H_2(z^{-1})z^{-1}]|(z = 0.62)$$

$$= (z-0.62)\frac{z^{-1}}{(z-0.62)\ (z^{-1}-0.62)}\Bigg|z = 0.62$$

$$=1.6244$$

$$\sigma_{e02}^2 = \frac{\sigma e^2}{2\pi j} \oint H_2(z) H_2(z^{-1}) z^{-1}\, dz$$

$=1.3021 \text{ X } 10^{-3} \text{ X } 1.6244$

$= 2.1151 \text{ X } 10^{-3}$

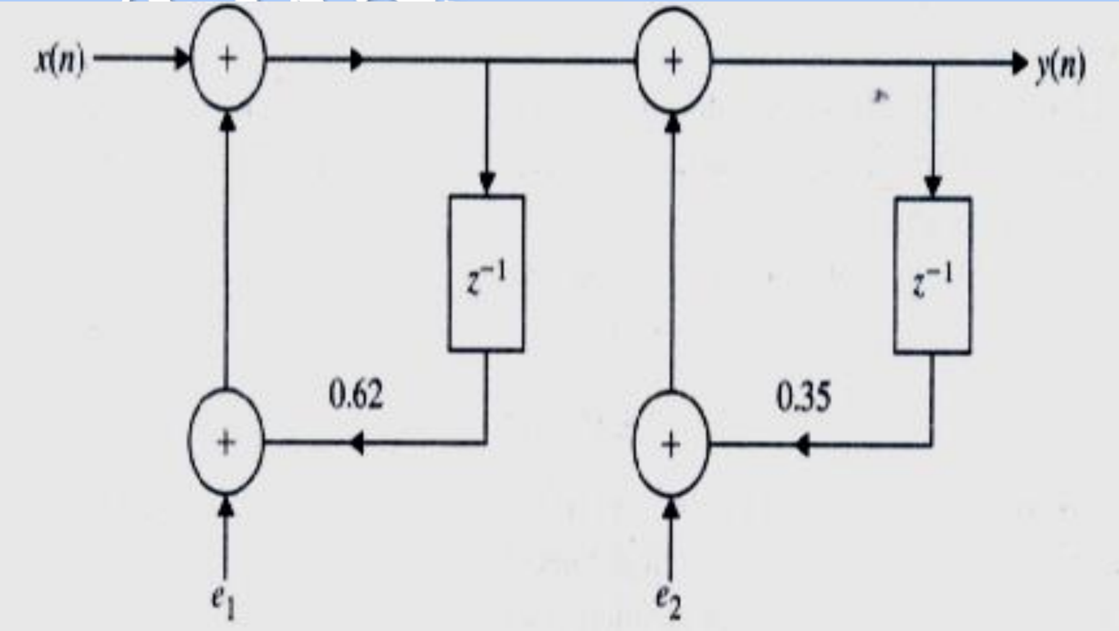
- **Total Output noise power**

$$\sigma_{e0}^2 = \sigma_{e01+}^2\sigma_{e02}^2$$

$$= 3.7465 \text{ x } 10^{-3} + 2.1151 \text{ x } 10^{-3}$$

- $\sigma_{e0}^2 = 5.8616\ X\ 10^{-3}$

- **Case *(ii)*** The order of cascading is $H(z) = H_2(Z)H_1(z)$ and is shown in Figure



The output noise power due to error source e_1 is,

$$\sigma^2_{e01} = 3.7465\ X\ 10^{-3}$$

The output noise power due to error source $e_1(n)$ is,

$$\sigma^2_{e02} = \frac{\sigma e^2}{2\pi j} \oint H_1(z) H_1(z^{-1}) z^{-1}\, dz$$

$$H_1(z)H_1(z^{-1})z^{-1} = \frac{z^{-1}}{(z-0.35)(z^{-1}-0.35)}$$

$$\mathrm{Res}[H_1(z)H_1(z^{-1})z^{-1}]|(z = 0.35)$$

$$= (z-0.35)\frac{z^{-1}}{(z-0.35)\,(z^{-1}-0.35)}\Bigg|$$

$$at\ z = 0.35$$

$$= 1.1396$$

$$\sigma_{e02}^2 = 1.1396\ X\ 1.3021\ X\ 10^{-3}$$

$= 1.4839 \text{ X } 10^{-3}$

Total output noise power

$$\sigma_{e0}^{2} = \sigma_{e01+}^{2}\sigma_{e02}^{2}$$

$$=3.7465 \;\; 10^{-3} + 1.4839 \text{ X } 10^{-3}$$

$$\sigma_{e0}^{2} = 5.2304\, X\, 10^{-3}$$

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

AJNIRMAL Notes

Limit cycle oscillations

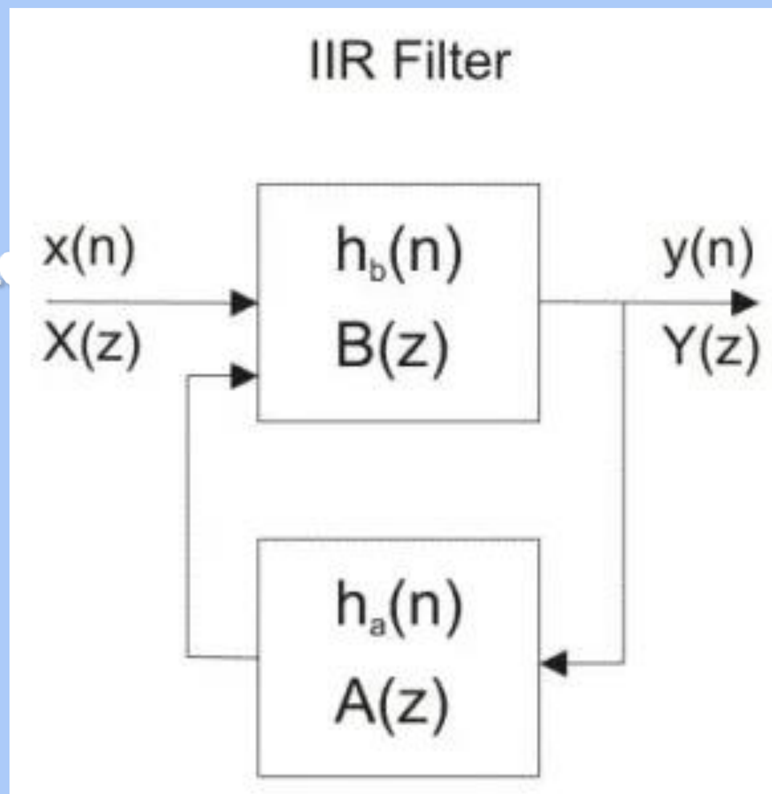
- Quantization is basically reducing the number of bits of a given number.
- The reduction/quantization produces a **non-linearity** in a filter system.
- This gives rise to the **finite word length effects. Limit cycle oscillations** are one of these unwanted effects.

What is limit cycle oscillation ?

- In some systems, when the input is zero or some non zero constant value the nonlinearities due to the finite precision arithmetic operations often cause periodic oscillations to occur in the output. Such oscillations in recursive systems are called limit cycle oscillations.

What is zero input limit cycle oscillations?

- Limit cycle oscillations will continue to remain in limit cycle even when the input is made zero. Hence, these limit cycle are also called zero input limit cycles


IIR Filter
x(n)
X(z)
h_b(n)
B(z)
y(n)
Y(z)
h_a(n)
A(z)

- limit cycle oscillations occur only in recursive systems. That is, it’s just the infinite-impulse response (IIR) filters that face this issue. Non-recursive FIR filters don’t experience limit cycle oscillations.
- Technically, in a practical, stable IIR filter excited by a finite sequence, the output will eventually decay to zero. But due to the non-linearities in the system, the issue of the limit cycle will keep some oscillations going in the output.

oscillations?

1. Zero-limit cycle oscillations
2. Overflow limit cycle oscillations

Zero limit cycle oscillations

- When a system output enters the limit cycle oscillation zone and continues to **show the periodic oscillations even after the input is made 0,** it is known as the **zero limit cycle oscillations**.

Overflow limit cycle oscillations

- In the fixed-point addition of two binary numbers, an 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 we call overflow limit cycles.
- 0.011 + 0.101 = 1.000. The 1 in the answer is an overflow output.

Remedy for overflow limit cycle

- We can solve the problem of overflow limit cycle oscillations by using **saturation arithmetic**.
- In saturation arithmetic, when an overflow is sensed, **the output is set to the maximum allowable value.**
- Conversely, when an underflow is detected, the output **will be set to the minimum permissible value**.
- Drawback of saturation arithmetic

It cause another undesirable signal distortion due to the non-linearity of the clipper.

Remedy for nonlinearity produced by saturation arithmetic

- To reduce these new, unexpected distortions, it is crucial to scale the input signal and the unit sample response between the input node and internal summation nodes.

Dead band

- **Dead band**
- During the limit cycle oscillations, the output of the filter oscillates between a finite positive and negative value. This range of values is called the Dead band of the filter. These values can be calculated with the following formula.
- Dead band =

Dead band of the filter.

- The limit cycles occur as a result of the quantization effects in multiplications. The amplitude of the output during a limit cycle are confined to a range of values that is called **the dead band** of the filter.
- The dead band is given by
- $Dead\ band = \pm \frac{2^{-b}}{1-|a|} = \left[\frac{-2^{-b}}{1-|a|}, \frac{2^{-b}}{1-|a|}\right]$