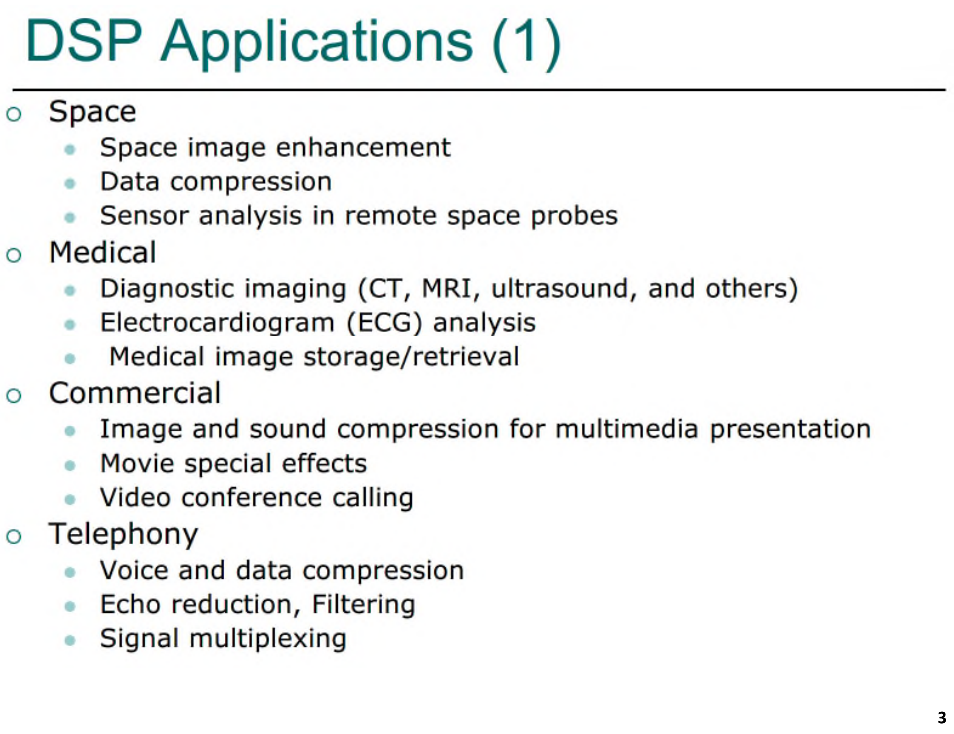
EE6402 Real-time DSP Design and Applications

Anamitra Makur

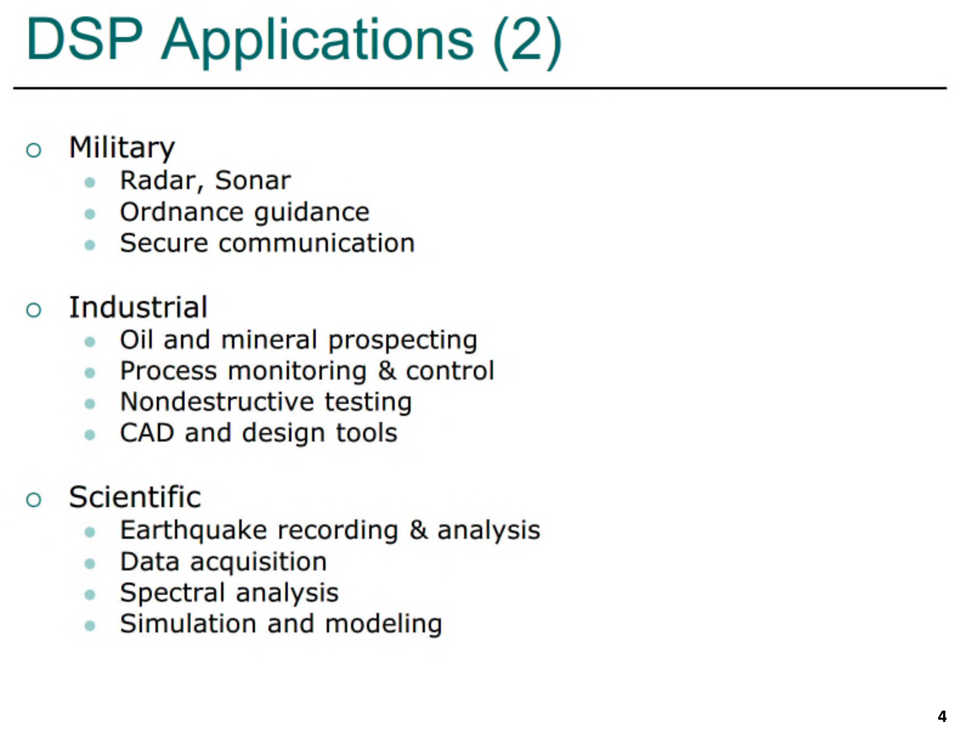
School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore

Introduction



DSP Applications (1)

- Space
  - Space image enhancement
  - Data compression
  - Sensor analysis in remote space probes
- Medical
  - Diagnostic imaging (CT, MRI, ultrasound, and others)
  - Electrocardiogram (ECG) analysis
  - Medical image storage/retrieval
- Commercial
  - Image and sound compression for multimedia presentation
  - Movie special effects
  - Video conference calling
- Telephony
  - Voice and data compression
  - Echo reduction, Filtering
  - Signal multiplexing



DSP Applications (2)

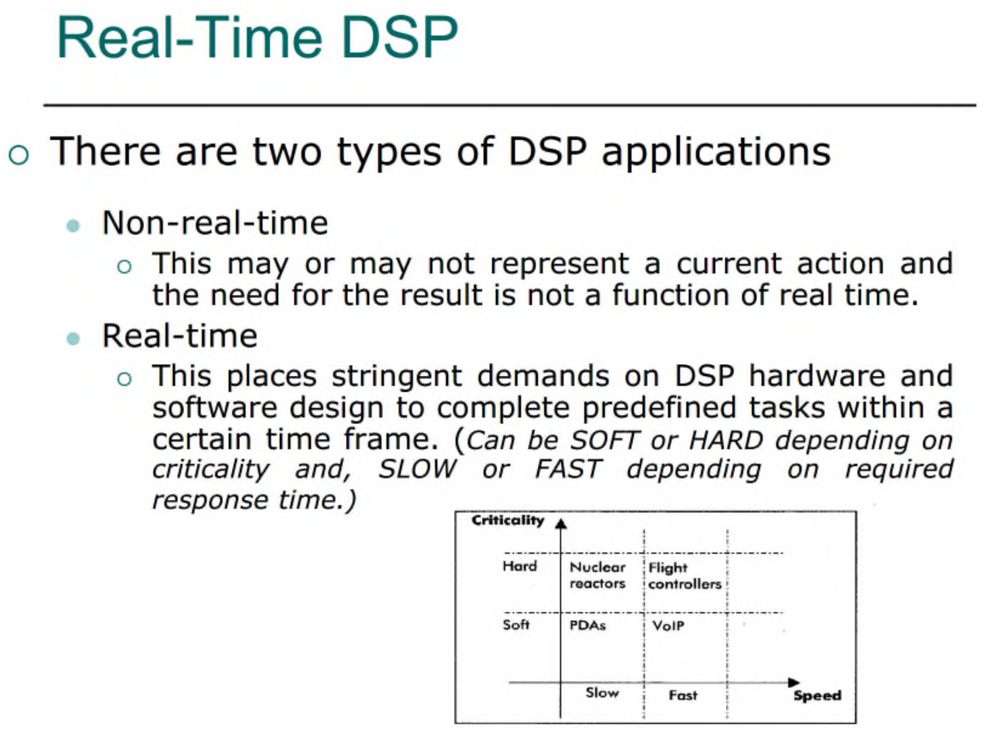
- Military
  - Radar, Sonar
  - Ordnance guidance
  - Secure communication
- Industrial
  - Oil and mineral prospecting
  - Process monitoring & control
  - Nondestructive testing
  - CAD and design tools
- Scientific
  - Earthquake recording & analysis
  - Data acquisition
  - Spectral analysis
  - Simulation and modeling

Why DSP (1)?

- Programmability
  - A DSP system is programmable and can be reprogrammed, even in field, to perform different tasks. In contrast, analog systems require physically different components to perform different tasks
- Stability
  - Insensitive to environment (e.g. Temperature)
  - insensitive to component tolerance (resistors, capacitors)
- Repeatability
  - It is possible to design systems having exact, known responses that do not vary
- Error correction
  - Use of error correction, e.g. Compact discs, data modem

Why DSP (2)?

- Easier implementation of algorithms
  - Adaptive SP, e.g. Noise cancellation
  - Linear phase filters, lossless data compression
  - Data transmission and storage
  - Data compression
- Lower cost (technology driven)
  - With the rapid evolution in semiconductor technology, DSP systems have a lower overall cost compared to analog systems in many cases
- Easy to develop, analyse, simulate and test (technology driven)
  - Using low-cost general purpose computers such as a PC



Real-Time DSP

- There are two types of DSP applications
  - Non-real-time
    - This may or may not represent a current action and the need for the result is not a function of real time.
  - Real-time
    - This places stringent demands on DSP hardware and software design to complete predefined tasks within a certain time frame. (*Can be SOFT or HARD depending on criticality and, SLOW or FAST depending on required response time.)*

	Slow	Fast
Hard	Nuclear reactors	Flight controllers
Soft	PDAs	VoIP

Part 1
Finite Precision Implementation Issues

- Fixed point and floating point number representations
- Round-off and truncation errors
- Finite word-length effects

Fixed Point and Floating Point Number Representations

Decimal Number System

The decimal number system uses the 10 symbols (0 , 1, 2 , 3 , 4 , 5 , 6 , 7, 8 , 9) to represent a number.

Example:

$(456)_{10} = 4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$

$$(3705.86)_{10} = 3 \times 10^3 + 7 \times 10^2 + 0 \times 10^1 + 5 \times 10^0 + 8 \times 10^{-1} + 6 \times 10^{-2}$$

Binary Number System

In binary number system, only two symbols (0 and 1) are used to represent a number.

Examples:

$$
\begin{aligned}
(\underset{2^4}{1}\underset{2^3}{1}\underset{2^2}{0}\underset{2^1}{0}\underset{2^0}{1})_2 &= (2^4)_{10} + (2^3)_{10} + (2^0)_{10} \\
&= (16)_{10} + (8)_{10} + (1)_{10} \\
&= (25)_{10}
\end{aligned}
$$

$$
\begin{aligned}
(101.01)_2 &= (2^2)_{10} + (2^0)_{10} + (2^{-2})_{10} \\
&= (4)_{10} + (1)_{10} + (0.25)_{10} \\
&= (5.25)_{10}
\end{aligned}
$$

Binary number

If there are N bits, the largest number that can be represented is $2^N - 1$. The smallest number is 0.

Example: A 3-bit number $\boxed{\bullet}\boxed{\bullet}\boxed{\bullet}$

The largest 3-bit number is $(111)_2 = 7 = 2^3 - 1 = 7$.

Negative number is not represented. To represent negative number, one extra bit, called the sign bit, is needed.

The 2's complement (two's complement) representation is one of the commonly used representation for representing negative numbers.

Two's Complement

The MSB (Most Significant Bit) carries a negative weight.

Example:

$$\begin{aligned}
(1101)_{2's} &= -2^3 + 2^2 + 2^0 &&= -8 + 4 + 1 &&= -3 \\
(1001)_{2's} &= -2^3 + 2^0 &&= -8 + 1 &&= -7 \\
(0110)_{2's} &= 2^2 + 2^1 &&= 4 + 2 &&= 6 \\
(110)_{2's} &= -2^2 + 2^1 &&= -4 + 2 &&= -2
\end{aligned}$$

Sign Extension

$$\begin{aligned}
(101)_{2's} &= -4 + 1 &&= -3 \\
(1101)_{2's} &= -8 + 4 + 1 &&= -3 \\
(11101)_{2's} &= -16 + 8 + 4 + 1 &&= -3 \\
(111101)_{2's} &= -32 + 16 + 8 + 4 + 1 &&= -3 \\
\therefore\ (101)_{2's} &= (1101)_{2's} = (11101)_{2's} = (1\cdots 101)_{2's}
\end{aligned}$$

$$\begin{aligned}
(0101)_{2's} &= 4 + 1 = 5 \\
(00101)_{2's} &= 4 + 1 = 5 \\
\therefore\ (0101)_{2's} &= (00101)_{2's} = (000101)_{2's} = (0\cdots 0101)_{2's}
\end{aligned}$$

This is called sign extension.

If there are N bits, the largest number that can be represented in 2's complement is $2^{N-1}-1$. The smallest number is -2^{N-1}.

Example: A 5-bit number $\boxed{\bullet}\boxed{\bullet}\boxed{\bullet}\boxed{\bullet}\boxed{\bullet}$

The largest number is $(01111)_{2's} = 15 \longleftarrow 2^4 - 1$.

The smallest number is $(10000)_{2's} = -16 \longleftarrow -2^4$.

Two's complement arithmetic

$(3)_{10} = (0011)_{2's}$
$(2)_{10} = (0010)_{2's}$

$$\begin{array}{r} 3 \\ +\,2 \\ \hline 5 \end{array} \qquad \begin{array}{r} 0\;0\;1\;1 \\ +\;\;0\;0\;1\;0 \\ \hline \underbrace{0\;1\;0\;1}_{5} \end{array}$$

$(-3)_{10} = (1101)_{2's}$
$(-2)_{10} = (1110)_{2's}$

$$\begin{array}{r} (-3) \\ +\,(-2) \\ \hline (-5) \end{array} \qquad \begin{array}{r} 1\;1\;0\;1 \\ +\;\;1\;1\;1\;0 \\ \hline \underset{\text{discard}}{\textcircled{1}}\,\underbrace{1\;0\;1\;1}_{-5} \end{array}$$

$$\begin{array}{r} (-3) \\ +\;\;2 \\ \hline (-1) \end{array} \qquad \begin{array}{r} 1\;1\;0\;1 \\ +\;\;0\;0\;1\;0 \\ \hline \underbrace{1\;1\;1\;1}_{-1} \end{array}$$

$$\begin{array}{r} 3 \\ +\ 7 \\ \hline 10 \end{array} \qquad\qquad \begin{array}{r} 0\ 0\ 1\ 1 \\ +\ 0\ 1\ 1\ 1 \\ \hline \underbrace{1\ 0\ 1\ 0}_{-6\ \ \text{Negative !}} \end{array}$$

This is because a 4-bit 2's complement number has a range of -2^3 to $2^3 - 1$, i.e. -8 to 7. $\therefore$ it cannot represent the result 10. Overflow has occurred. At least 5 bits are needed to represent the number $(10)_{10}$ in 2's complement.

$$\begin{array}{r} 3 \\ +\ 7 \\ \hline 10 \end{array} \qquad\qquad \begin{array}{r} 0\ 0\ 0\ 1\ 1 \\ +\ 0\ 0\ 1\ 1\ 1 \\ \hline \underbrace{0\ 1\ 0\ 1\ 0}_{10} \end{array}$$

$$\begin{array}{r} 3 \\ +\ \ 7 \\ \hline 10 \\ +\,(-5) \\ \hline 5 \end{array} \qquad\qquad \begin{array}{rr} & 0\ 0\ 1\ 1 \\ + & 0\ 1\ 1\ 1 \\ \hline & 1\ 0\ 1\ 0 \longrightarrow -6\ \ \text{Negative !} \\ + & 1\ 0\ 1\ 1 \\ \hline \underset{\text{discard}}{\textcircled{1}} & \underbrace{0\ 1\ 0\ 1}_{5\ \Rightarrow\ \text{correct result}} \end{array}$$

This is because the operands can be sign extended.

$$\begin{array}{r} 3 \\ +\ \ 7 \\ \hline 10 \\ +\,(-5) \\ \hline 5 \end{array} \qquad\qquad \begin{array}{rrr} & 0 & 0\ 0\ 1\ 1 \\ + & 0 & 0\ 1\ 1\ 1 \\ \hline & 0 & 1\ 0\ 1\ 0 \longrightarrow 10\ \text{(correct result)} \\ + & 1 & 1\ 0\ 1\ 1 \\ \hline & 0 & \underbrace{0\ 1\ 0\ 1}_{5\ \Rightarrow\ \text{correct result}} \end{array}$$

Virtual sign extension

Multiplication of 2 *n*-bit numbers gives a 2*n* bit product.

$(3)_{10} = (011)_{2's}$
$(2)_{10} = (010)_{2's}$
(simple since
both positive)

```
     011
    ×010
    ----
     000
    011
   000
  ------
  000110 = (6)10
```

$(-3)_{10} = (101)_{2's}$ $\quad$ $(3)_{10} = (011)_{2's}$
$(2)_{10} = (010)_{2's}$ $\quad$ $(-2)_{10} = (110)_{2's}$

use sign extension to 2*n* bits

```
      111101              000011
     ×000010             ×111110
      ------              ------
      000000              000000
     *11101              *00011
    **0000              **0011
   ***000              ***011
  ****00              ****11
 *****0              *****1
      ------              ------
      111010 = (–6)10     111010 = (–6)10
```

Fixed point number

Q*m.n* format: *n* fractional bits, *N* = *n*+*m*+1 bit number

Example: Q2.5 format $\quad$ $(010.10010)_{2's} = 2.5625$

Typical formats:
Q0.15 (or Q.15 or Q15) format, 16-bit fraction

$(0.000000000000001)_{2's} = 0.0000305…$

Q15.0 format, 16-bit integer

$(0100000000000000)_{2's} = 16384$

Dynamic range =

$$20log_{10}\left(\frac{\text{largest number}}{\text{smallest positive number excluding 0}}\right)$$

Example: Q0.15 format
Largest number = $(0.111111111111111)_{2\text{'s}}$ = $1 - 2^{-15}$
Smallest number = $(0.000000000000001)_{2\text{'s}}$ = 2^{-15}
Dynamic range = 90 dB

Example: Q15.0 format
Largest number = $(0111111111111111)_{2\text{'s}}$ = $2^{15} - 1$
Smallest number = $(0000000000000001)_{2\text{'s}}$ = 1
Dynamic range = 90 dB

In fact, any other 16-bit 2’s complement Q*m.n* format, also has a dynamic range of 90 dB

Precision = difference between 2 consecutive numbers

Examples:

Q0.15 format, precision = 2^{-15}

Q15.0 format, precision = 1

Addition of fixed point numbers:
Adding two *N*-bit Q*m.n* numbers requires (*N*+1)-bit Q(*m* +1).*n* format to avoid overflow

Multiplication of fixed point numbers:
Multiplying two *N*-bit Q*m.n* numbers requires 2*N*-bit Q(2*m*+1).(2*n*) format to avoid overflow

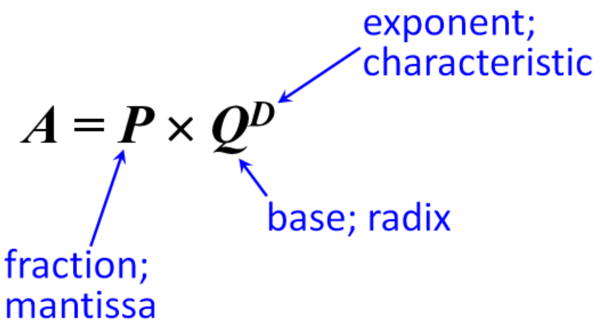
Example: Q3.0 format

$$\begin{array}{rr} 7 & (0111)_{2's} \\ \times\ 6 & (0110)_{2's} \\ \hline 42 & (00101010)_{2's} \end{array}$$

Q0.3 format

$$\begin{array}{rr} 0.875 & (0.111)_{2's} \\ \times\ 0.75 & (0.110)_{2's} \\ \hline 0.65625 & (00.101010)_{2's} \end{array}$$

Floating point number



$$A = P \times Q^{D}$$

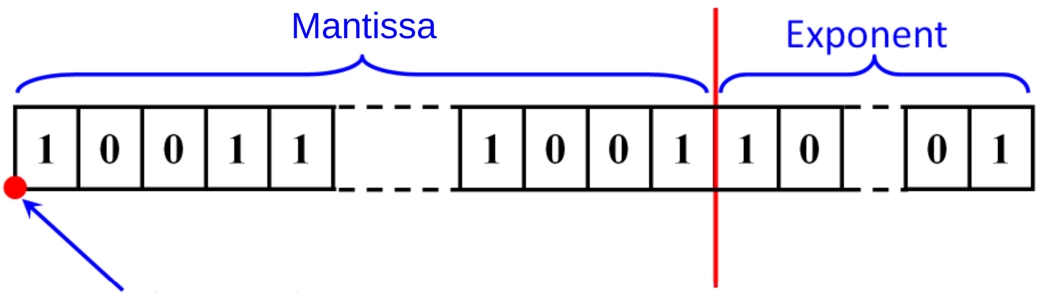
exponent; characteristic

base; radix

fraction; mantissa

Example: 8934 can be written as 0.8934×10^4

Binary representation



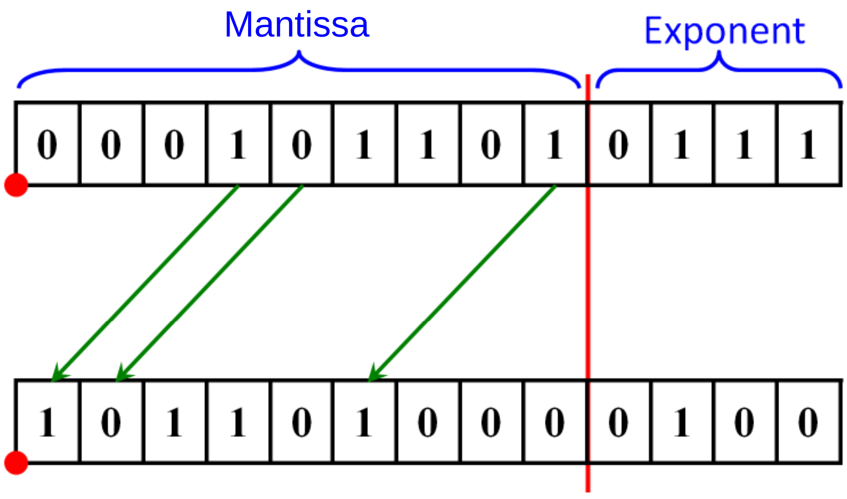
Mantissa

Exponent

1 0 0 1 1 - - - 1 0 0 1 1 0 - - 0 1

Binary point

For maximum precision, the number is normalized until the first digit is a "1".



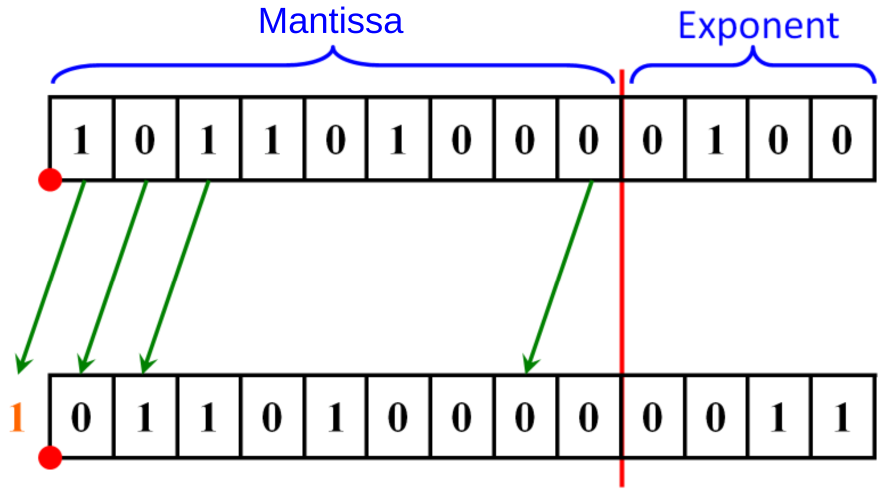
Mantissa

Exponent

0 0 0 1 0 1 1 0 1 0 1 1 1

1 0 1 1 0 1 0 0 0 0 1 0 0

Since the first digit is always a "1", it is not necessary to record the first digit. We know that it is a "1"! Thus, it is discarded.



IEEE 754

IEEE Standard for Binary Floating-Point Arithmetic

IEEE 754 specifies four formats for representing floating-point values.

(1) Single-precision (32-bit)

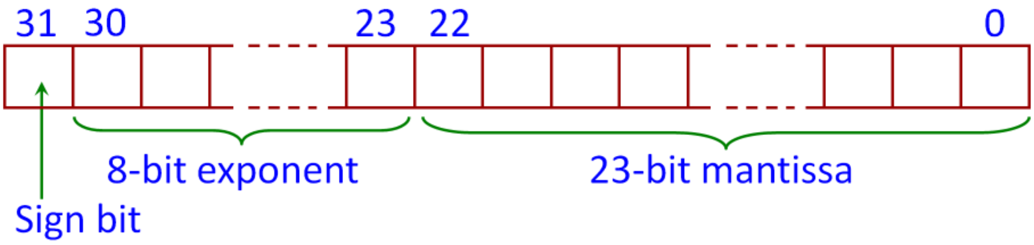
(2) Double-precision (64-bit)

(3) Single-extended precision (≥ 43-bit, seldom used)

(4) Double-extended precision (≥ 79-bit, usually 80 bits).

Only the 32-bit format is required by the standard; the others are optional.

32-bit single precision format



Sign bit

Number is positive if sign bit is "0".

Number is negative if sign bit is "1".

Biased exponent

The exponent have to be signed values in order to be able to represent both large and small magnitudes. 2’s complement is not used because it would be harder to compare two numbers. Thus, a constant equal to $2^{E-1}-1$ is added to the exponent to put the exponent within an unsigned range for recording in the "exponent" bit. For example, for E = 8, The exponent bias is 2^7-1 = 127. If the exponent is –3, it will be recorded as –3 + 127 = 124, i.e. $(01111100)_2$

Exponent:
Normal range 1 to 2^E – 2 (1 to 254)

Special cases:
For 0 and denormalized numbers, exponent = 0
For ±∞ and NaNs, exponent = 2^E – 1 (255)

Mantissa:
D = denormalized mantissa value

M = 1.(denormalized mantissa value) = 1.D

Mantissa value

Mantissa value = 1.(denormalized mantissa value) = 1.D

Example

	22																						0
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23-bit mantissa

$D = 2^{-2} + 2^{-3} = 0.25 + 0.125 = 0.375$

Mantissa value = $1 + 2^{-2} + 2^{-3}$

$= 1 + 0.25 + 0.125$

$= 1.375$

IEEE 754 single precision

	Exponent	Mantissa
Normalized numbers	1 to $2^E - 2$ (254), biased binary	Any number
Denormalized numbers	0	Non-zero
Zeroes	0	0
Infinities	$2^E - 1$ (255)	0
NaNs	$2^E - 1$ (255)	Non-zero

Value of normalized number:

Given

S	Exp	D

$$\text{Value} = (-1)^S \times 2^{Exp-127} \times M$$

For example, given

0	10000010	10000000000000000000000

Sign S = 0
Exponent $Exp = (10000010)_2 = 130$
Denormalized mantissa $D = (100\ldots)_2$
Mantissa $M = (1.100\ldots)_2 = 1.5$
So, value = $(-1)^0 \times 2^{130-127} \times 1.5 = 12$

Value of denormalized number = $(-1)^{S} \times 2^{-(Bias-1)} \times D$

where $Bias = 2^{E-1} - 1$.

$2^{-(Bias-1)} = 2^{-126}$ for single precision format

D = denomalized mantissa value

= 0.(value represented by mantissa bit)

S	00000000	01011000000000000000000

For example: Mantissa = $(01011000\ldots0)_2$

Denormalized mantissa value

$D = 2^{-2} + 2^{-4} + 2^{-5} = 0.25 + 0.0625 + 0.03125 = 0.34375$

$\therefore$ value = $(-1)^{S} \times 2^{-126} \times 0.34375$

Dynamic range of single precision format:

Largest number: $\boxed{S}\boxed{Exp}\boxed{D}$ = $\boxed{0}\boxed{254}\boxed{\text{all 1s}}$

$M = 1.D = 2 - 2^{-23}$

$(-1)^0 \times (2 - 2^{-23}) \times 2^{254-127} = 3.403 \times 10^{38}$

Smallest number: $\boxed{0}\boxed{1}\boxed{\text{all 0s}}$

$M = 1.0$

$(-1)^0 \times 1.0 \times 2^{1-127} = 1.176 \times 10^{-38}$

Dynamic range = 1529 dB

Alternately, 8-bit exponent: 2^{-126} to 2^{127} (254 ranges of 2)

Each range of 2 = 20 log 2 = 6.02 dB dynamic range

So, 254 ranges of 2 = 254 x 6.02 dB = 1529 dB

Precision for floating point = difference between 2 consecutive numbers with a common exponent

Precision of single precision format: 2^{-23}

Fixed point versus floating point:

For same number of bits, floating point format provides a larger dynamic range due to the scaling by the exponent field.

However, fixed point format provides a finer precision since it has more bits than the mantissa field.

Floating point addition:

Example: $\boxed{0}\boxed{128}\boxed{0.22} = (-1)^0 \text{ x } 1.22 \text{ x } 2^{128-127} = 2.44$

$+ \ \boxed{1}\boxed{130}\boxed{0.52} = (-1)^1 \text{ x } 1.52 \text{ x } 2^{130-127} = -12.16$

Adjust the smaller (absolute) number’s exponent equal to the other exponent:

$$1.22 \text{ x } 2^{128-127} = 0.305 \text{ x } 2^{130-127}$$

Add both numbers:

$$(0.305-1.52) \text{ x } 2^{130-127} = -1.215 \text{ x } 2^{130-127}$$

Solution $\boxed{1}\boxed{130}\boxed{0.215} = (-1)^1 \text{ x } 1.215 \text{ x } 2^{130-127} = -9.72$

Floating point multiplication:

Example: $\boxed{0}\boxed{128}\boxed{0.22} = (-1)^0 \text{ x } 1.22 \text{ x } 2^{128-127} = 2.44$

$\text{x } \boxed{1}\boxed{130}\boxed{0.52} = (-1)^1 \text{ x } 1.52 \text{ x } 2^{130-127} = -12.16$

Multiply $(-1)^{0+1} \text{ x } (1.22 \text{ x } 1.52) \text{ x } 2^{(128-127)+(130-127)}$

$= (-1)^1 \text{ x } 1.8544 \text{ x } 2^4$

$= (-1)^1 \text{ x } 1.8544 \text{ x } 2^{131-127}$

$= \boxed{1}\boxed{131}\boxed{0.8544}$

$= -29.6704$

Overflow occurs in floating point arithmetic, too. However, throwing away the LSBs of the mantissa gives little error.

Quantization, Truncation, and Round-off

Quantizer

- Range of input signal = V_{min} to V_{max} (typically $V_{min} = -V_{max}$)
- Quantizer output = n bits

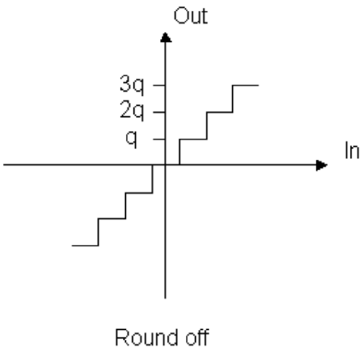
Typically, a quantizer is mid-tread (origin on tread of staircase)

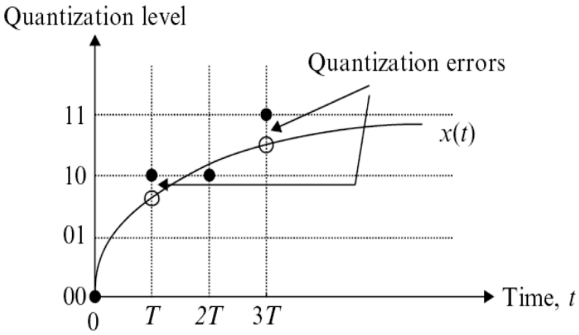
- Number of quantizer levels = $2^n - 1$
- Quantization step size $$Q = \frac{V_{max} - V_{min}}{2^n - 1}$$

A mid-rise quantizer (origin on rise of staircase)

- Number of quantizer levels = 2^n
- Quantization step size $$Q = \frac{V_{max} - V_{min}}{2^n}$$

Mid-tread Quantizer





Output values … $-Q, 0, Q, 2Q, 3Q, \ldots$

Rounding: assign the input value to the nearest output value

- Such as, $-Q/2 <$ input $< Q/2$ is assigned to output 0
- Decision boundaries … $-Q/2, Q/2, 3Q/2, \ldots$

Quantization error

Recall rounding results:
Additive noise model:
quantized output = input + quantization error

$$[x(n)] = x(n) + e(n)$$

- The probability density function for $e(n)$ is uniform from $-Q/2$ to $Q/2$
- mean of $e(n)$ = 0
- maximum $e(n) = Q/2$
- mean square error = variance of $e(n) = Q^2/12$
- root mean square error = std dev of $e(n) = 0.29Q$

Error $e(n)$ is white noise (two different error values are uncorrelated)

Error $e(n)$ is uncorrelated with the input signal $x(n)$

Addition of uncorrelated noise sources:

- mean = $\eta_1 + \eta_2$
- variance = $\sigma_1^2 + \sigma_2^2$

Example:
Compare the rms error of a 12-bit ADC with that of a 8-bit ADC for an input having maximum amplitude of 0.5V.

rms error = $0.29Q = 0.29(V_{max} - V_{min})/2^n$

For 12-bit ADC, it is $0.29(0.5+0.5)/2^{12}$ = 71μV

For 8-bit ADC, it is $0.29(0.5+0.5)/2^{8}$ = 1.1mV

Example:
An input to a 8-bit ADC consists of a signal having range 0V to 1V plus an uncorrelated noise having rms value of 1mV. What is the rms noise at the output?

quantization error variance = $Q^2/12 = 0.0000013$
output noise variance = $0.0000013+(0.001)^2 = 0.0000023$
rms output noise = $\sqrt{0.0000023}$ = 1.5mV

How to decide on the number of bits required for a system:

- Find how much noise is already present in the input
- Find how much noise can be tolerated in the system

Signal to Quantization Noise Ratio (SQNR)

- Range of input signal = $-V_{max}$ to V_{max}
- Signal variance = σ_x^2
- Quantization noise variance = σ_e^2

$$
\begin{aligned}
SQNR &= 10\log\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \\
&= 10\log\left(\frac{\sigma_x^2}{Q^2/12}\right) \\
&= 10\log\left(\frac{3\sigma_x^2 2^{2n}}{V_{max}^2}\right) \\
&= 6.02n + 4.77 + 20\log(\sigma_x/V_{max})
\end{aligned}
$$

For speech or music, to avoid peak clipping, set the gain of the filter and the amplifier preceding the ADC so that

$$\sigma_x = V_{max}/4$$

Then SQNR ≈ $6.02n - 7.27$ dB

SQNR improves by 6 dB per additional bit of ADC

SQNR can also be improved by over-sampling and low-pass filtering. Doubling the sampling rate improves the SQNR by 3 dB.

Matlab demo: Q, SQNR of $e(n)$

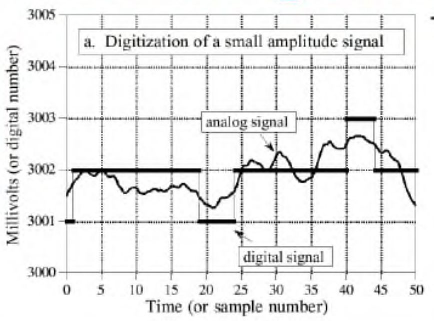
Dithering

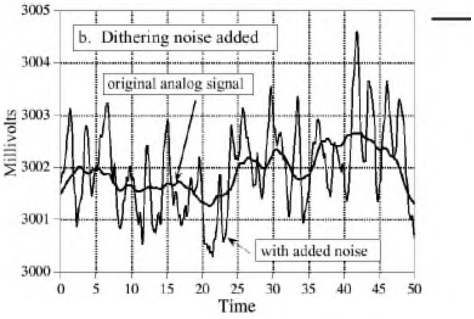
Input signal remains nearly constant (or slowly varying) for many samples
➔Quantization error is not random/uncorrelated
➔Additive noise model is no longer valid
Quantization looks like thresholding

Solution: dithering

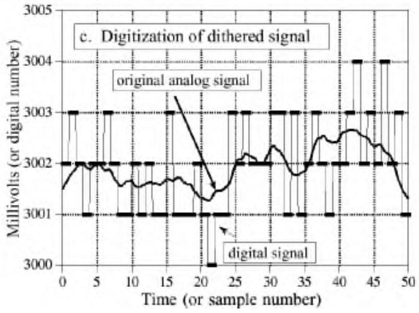
- Add a small random noise to the input signal
- Typically, uniform/triangular/Gaussian noise is taken
- Even if input signal is constant, ADC output randomly changes between adjacent levels due to the added noise

Dithering Example





The added noise causes the digitized signal to toggle between adjacent quantization levels, providing more information about the original signal.



For example, input signal = 2.1, Q = 1

- Conventional quantization output = 2 for all samples

- Dithering with uniform noise of range –0.5 to 0.5
- 90% inputs from 1.6 to 2.5 → output 2
- 10% inputs from 2.5 to 2.6 → output 3
- Average output = 2.1

- Dithering with uniform noise of range –1 to 1
- 20% inputs from 1.1 to 1.5 → output 1
- 50% inputs from 1.5 to 2.5 → output 2
- 30% inputs from 2.5 to 3.1 → output 3
- Average output = 2.1

Advantages and disadvantages of dithering

Advantge:

- Adding noise provides more information!
- Actually, averaging of output samples leads to removal of noise and more information

Disadvantage:

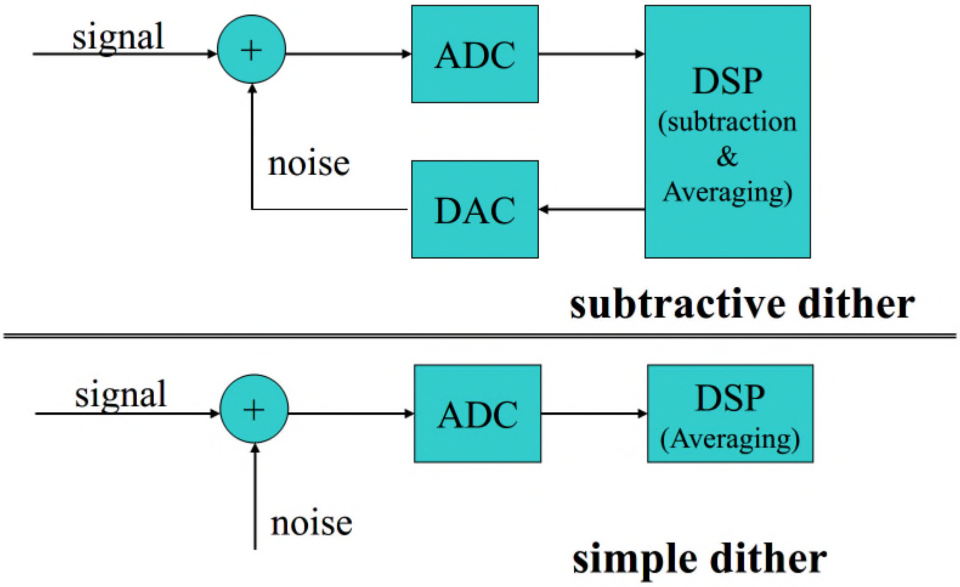
- Constant/slowly varying input signal is required for (averaging of) multiple similar samples
- Otherwise, dithering increases output noise

Subtractive dithering

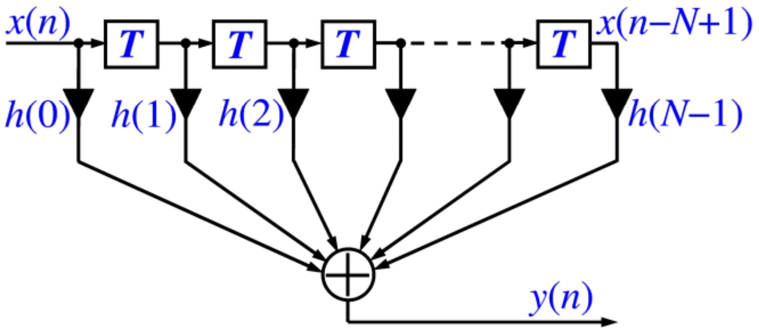
- Generate random digital noise using random number generator
- Use DAC to produce random analog noise
- Add this noise
- After ADC, subtract this noise from the digital signal

Advantage:

- Doesn't increase output noise unlike simple dithering



The computation of the output of a filter requires many multiplication operations. The word-length of the product of multiplication is equal to the sum of the word-lengths of the operands, i.e. the product is double word-length. The word-length of the product is usually truncated to reduce complexity.



Methods for word-length truncation

Two of the most popular methods are:

Truncation:	Rounding:
275.386 → 275	275.386 → 275
275.836 → 275	275.836 → 276

In general, word-length truncation causes error. Word-length truncation may be modeled as a noise injection.

Example: $\underbrace{275}_{\text{Truncated value}} = \underbrace{275.386}_{\text{Original value}} + \underbrace{(-0.386)}_{\text{Error = noise}}$

Quantization Step Size

The difference between consecutive discrete values of a discrete space is called the quantization step size.

Example:

In the integer space, 1, 2, 3, …, 274, 275, 276, …. The difference between consecutive integers is "1". The quantization step size is "1".

Truncation:	Rounding:
$275.386 \rightarrow 275$	$275.386 \rightarrow 275$
$275.836 \rightarrow 275$	$275.836 \rightarrow 276$

In the above example, the quantization step size is "1".

If data are rounded or truncated to 1 decimal place, the quantization step size is 0.1.

Truncation:	Rounding:
$275.386 \rightarrow 275.3$	$275.386 \rightarrow 275.4$
$275.836 \rightarrow 275.8$	$275.836 \rightarrow 275.8$

5	4	3.	2	1	0	Quantization step size = 10^{-3}
5	4	3.	2	1	X	Quantization step size = 10^{-2}
5	4	3.	2	X	X	Quantization step size = 10^{-1}
5	4	X.	X	X	X	Quantization step size = 10.

Similarly in binary

$$0.11001101 = 0.1101 - 0.00000011$$

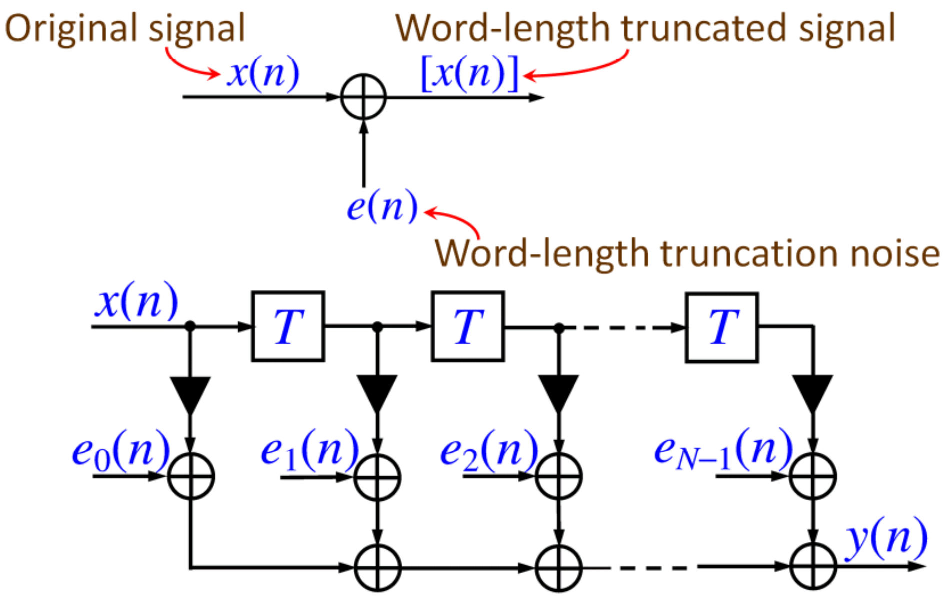
$$\therefore\ \underbrace{0.1101}_{\text{Rounded value}} = \underbrace{0.11001101}_{\text{Original value}} + \underbrace{0.00000011}_{\text{Error = noise}}$$

Thus, we have

Rounded value = Original value + error

In this particular example, the rounded value is quantized to 4 binary bits after the binary point. The quantization step size is $(0.0001)_2$.

Modeling of word-length truncation error as a noise injection



Original signal

Word-length truncated signal

Word-length truncation noise

$$[x(n)] = x(n) + e(n)$$

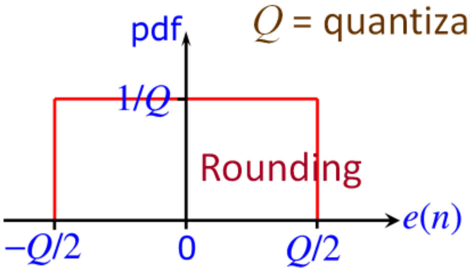
Properties of $e(n)$ assuming $x(n)$ has infinite precision.

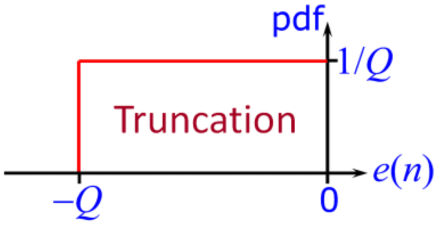
Example: $x(n) = \underbrace{0.101100110101010111000101\cdots\cdots}_{\text{Infinite number of bits}}$

The probability of $x(n)$ having any specific value is zero.

The probability density function for $e(n)$ is uniform.

Q = quantization step size.





Matlab demo: pdf of $e(n)$

Error Analysis

Statistical properties of the error

1) Mean and variance

$e(n)$ is a uniform random variable.
Let its probability density function be:

$$f(x) = \begin{cases} \dfrac{1}{B-A} & A \leq x \leq B \\ 0 & \text{otherwise} \end{cases}$$

Then, expectation of any function of x, say $g(x)$, is

$$E\{g(x)\} = \int g(x) f(x) dx = \frac{1}{B-A} \int_A^B g(x) dx$$

mean of $e(n)$ = average value of $e(n)$

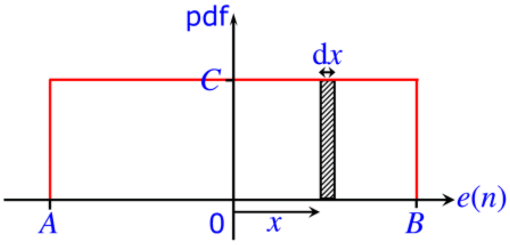
$$= \mathrm{E}\{x\}$$

$$= \frac{1}{B-A} \int_A^B x dx$$

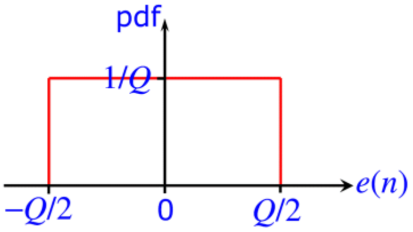
$$= \frac{1}{B-A} \left[\frac{x^2}{2}\right]_A^B$$

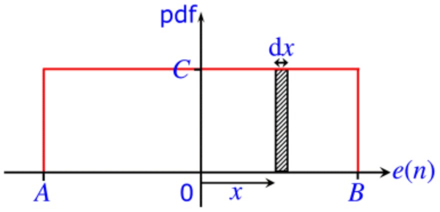
$$= \frac{A+B}{2}$$

$$= \eta$$



Rounding:

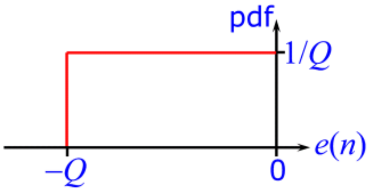


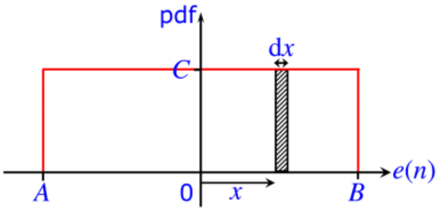


$$\eta = \frac{A+B}{2}$$

For rounding, $A = -Q/2$, $B = Q/2$, thus, $\eta = 0$

Truncation:



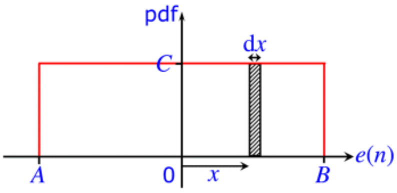


$$\eta = \frac{A+B}{2}$$

For truncation, $A = -Q$, $B = 0$, thus, $\eta = -\frac{Q}{2}$

Variance of $e(n)$ $= E\{(x-\eta)^2\} = \int \left(x^2 - 2\eta x + \eta^2\right) f(x)dx$

$$= \int x^2 f(x)dx - 2\eta \int x f(x)dx + \eta^2 = \int x^2 f(x)dx - 2\eta^2 + \eta^2$$



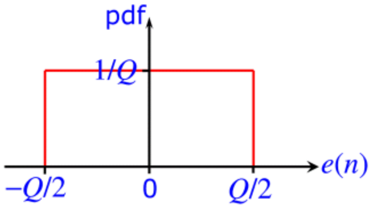
$$= \frac{1}{B-A}\int_A^B x^2 dx - \eta^2$$

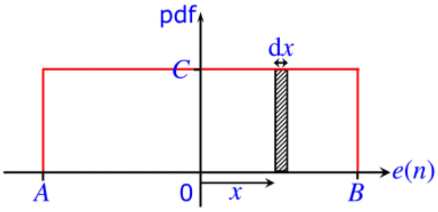
$$= \frac{1}{(B-A)}\left[\frac{x^3}{3}\right]_A^B - \left(\frac{A+B}{2}\right)^2$$

$$= \frac{B^3 - A^3}{3(B-A)} - \left(\frac{A+B}{2}\right)^2 = \frac{B^2 + AB + A^2}{3} - \frac{A^2 + 2AB + B^2}{4}$$

$$= \frac{(B-A)^2}{12} \qquad = \sigma^2$$

Rounding:

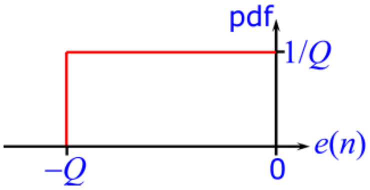


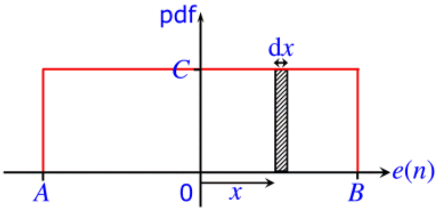


$$\sigma^2 = \frac{(B-A)^2}{12}$$

For rounding, $A = –Q/2$, $B = Q/2$. Thus, $\sigma^2 = \frac{Q^2}{12}$

Truncation:





$$\sigma^2 = \frac{(B-A)^2}{12}$$

For truncation, $A = -Q$, $B = 0$. Thus, $\sigma^2 = \frac{Q^2}{12}$

Summary:

Rounding: mean = 0
variance = $Q^2/12$

Truncation: mean = $-Q/2$
variance = $Q^2/12$

Matlab demo: mean, variance of $e(n)$

2) Auto-correlation and power spectral density of zero-mean noise

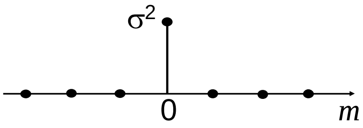
Let the error $e(n)$ be random with the following properties.

Any two different error values $e(n)$, $e(n+m)$ are uncorrelated.
Auto-correlation of $e(n)$ is

$$R(m) = E\{e(n)e(n+m)\} = \begin{cases} E\{e(n)^2\} & m = 0 \\ 0 & m \neq 0 \end{cases}$$

But $E\{e(n)^2\} = \sigma^2$, variance of $e(n)$

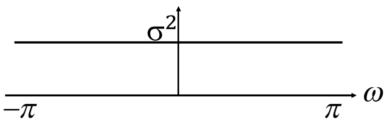
Or, $R(m) = \sigma^2\,\delta(m)$
where $\delta(m)$ = Kronecker delta function



Since uncorrelated, it is not possible to predict or estimate the value of $e(n)$ (apart from its mean value) from the past values $e(n–1)$, $e(n–2)$, $e(n–3)$, etc.

The power spectral density of $e(n)$ is the Fourier transform of its auto-correlation:

$$\begin{aligned} S\left(e^{j\omega}\right) &= \mathcal{F}\{R(m)\} \\ &= \mathcal{F}\{\sigma^2\delta(m)\} \\ &= \sigma^2 \end{aligned}$$

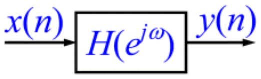


The power spectrum is flat
→ All frequencies are equally present
→ White noise
For example, round off noise, where $\sigma^2 = \dfrac{Q^2}{12}$

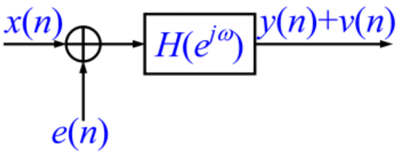
Matlab demo: auto-corr, power spectral density of $e(n)$

3) Effect of input noise at the output of a system

An input $x(n)$ passes through a system with frequency response $H(e^{j\omega})$ to give an output $y(n)$.



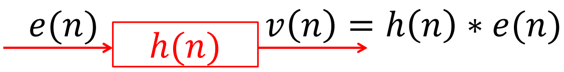
With quantization, $x(n)+e(n)$ passes through the system to give an output $y(n)+v(n)$.



Due to linearity, $v(n)$ is the output noise when $e(n)$ is the input noise.

The mean and the variance of $v(n)$ are related to the mean and the variance of $e(n)$.

The output is the convolution of the input with the impulse response:



$$v(n) = \sum_k h(k)e(n-k)$$

Taking expectation of both sides,

$$E\{v(n)\} = E\{\sum_k h(k)e(n-k)\}$$
$$= \sum_k h(k)E\{e(n-k)\}$$

or $$\eta_v = \sum_k h(k)\eta_e$$

Using $H\left(e^{j\omega}\right) = \sum_n h(n)e^{-j\omega n}$, we obtain $H\left(e^{j0}\right) = \sum_n h(n)$

Therefore, the mean of $v(n)$ is: $\eta_v = H\left(e^{j0}\right)\eta_e$

Or, mean of $v(n)$ = gain of the system at ω=0 × mean of $e(n)$

For any input/output signal $\xrightarrow{x(n)} \boxed{h(n)} \xrightarrow{y(n)}$

from $y(n) = \sum_k h(k)x(n-k)$ we find

$$x(n)y(n+m) = x(n)\sum_k h(k)x(n+m-k)$$

Taking expectation of both sides,

$$\begin{aligned} E\{x(n)y(n+m)\} &= E\{\sum_k h(k)x(n)x(n+m-k)\} \\ &= \sum_k h(k)E\{x(n)x(n+m-k)\} \end{aligned}$$

or $\quad R_{cross}(m) = \sum_k h(k)R_x(m-k) = h(m) * R_x(m)$

where $R_{cross}(m)$ is the cross-correlation between $x(n)$ and $y(n)$, and $R_x(m)$ is the auto-correlation of $x(n)$.

Taking Fourier transform of both sides,

$$S_{cross}(e^{j\omega}) = H(e^{j\omega})S_x(e^{j\omega})$$

where $S_{cross}(e^{j\omega})$ is the cross-power spectral density between $x(n)$ and $y(n)$, and $S_x(e^{j\omega})$ is the power spectral density of $x(n)$.

Similarly, it may be shown that $S_y(e^{j\omega}) = H^*(e^{j\omega})S_{cross}(e^{j\omega})$

Therefore, $S_y(e^{j\omega}) = H(e^{j\omega})H^*(e^{j\omega})S_x(e^{j\omega})$

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

This result is true for any input and output:

$$S_{in}(e^{j\omega}) \longrightarrow \boxed{H(e^{j\omega})} \longrightarrow S_{out}(e^{j\omega}) = \left|H(e^{j\omega})\right|^2 S_{in}(e^{j\omega})$$

$$e(n) \rightarrow \boxed{H(e^{j\omega})} \rightarrow v(n)$$

Let $e(n) = \eta_e + u(n)$
→ $u(n)$ is a zero-mean white noise with variance $\sigma_u^2 = \sigma_e^2$ and power spectral density $S_u(e^{j\omega}) = \sigma_e^2$

Let $v(n) = \eta_v + w(n)$
→ $w(n)$ is a zero-mean noise with variance $\sigma_w^2 = \sigma_v^2$

It has already been shown that $\eta_e \rightarrow \boxed{H(e^{j\omega})} \rightarrow \eta_v$

→ From linearity, $u(n) \rightarrow \boxed{H(e^{j\omega})} \rightarrow w(n)$

→ $S_w(e^{j\omega}) = |H(e^{j\omega})|^2 S_u(e^{j\omega}) = |H(e^{j\omega})|^2 \sigma_e^2$

Variance of $w(n)$ $\quad \sigma_w^2 = E\{w(n)^2\} = R_w(0)$

But $\quad R_w(m) = \mathcal{F}^{-1}\{S_w(e^{j\omega})\} = \int_{-\pi}^{\pi} S_w(e^{j\omega}) e^{j\omega m} \frac{d\omega}{2\pi}$

Then $\sigma_w^2 = R_w(0) = \int_{-\pi}^{\pi} S_w(e^{j\omega}) e^{j\omega 0} \frac{d\omega}{2\pi}$

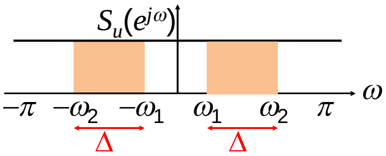
$$= \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \sigma_e^2 \frac{d\omega}{2\pi}$$

Variance of a zero-mean signal = area under its power spectral density

So $\sigma_v^2 = \|H(e^{j\omega})\|_2^2 \sigma_e^2$ where $\|H(e^{j\omega})\|_2^2 = \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \frac{d\omega}{2\pi}$

Or, variance of $v(n)$ = power gain of $H(e^{j\omega})$ × variance of $e(n)$

For example, let $\left|H\left(e^{j\omega}\right)\right| = \begin{cases} 1 & \omega_1 \le \omega \le \omega_2 \\ 0 & otherwise \end{cases}$



passband from ω_1 to ω_2
$\Delta = \omega_2 - \omega_1$

For a real signal, a passband from ω_1 to ω_2 means the band from $-\omega_2$ to $-\omega_1$ is automatically included, or the bandwidth is actually 2Δ.

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

If $e(n)$ is quantization noise so that $\sigma_e^2 = \frac{Q^2}{12}$, then $\sigma_v^2 = \frac{Q^2\Delta}{12\pi}$

Summary:

mean of output noise $\eta_v = H\left(e^{j0}\right)\eta_e$

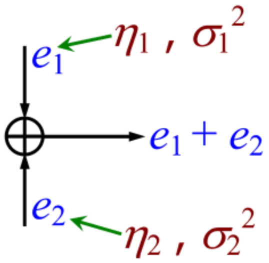
variance of output noise $\sigma_v^2 = \left(\int_{-\pi}^{\pi} \left|H\left(e^{j\omega}\right)\right|^2 \frac{d\omega}{2\pi} \right) \sigma_e^2$

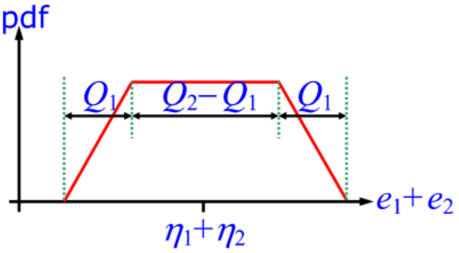
input and output power spectral density are related by

$$S_{out}(e^{j\omega}) = \left|H\left(e^{j\omega}\right)\right|^2 S_{in}\left(e^{j\omega}\right)$$

4) Addition of uncorrelated noise sources

Assume that e_1 and e_2 are uncorrelated and that the quantization step size $Q_2 \geq Q_1$. (Note: If x and y are uncorrelated , then the pdf of $x + y$ equals the convolution of their respective pdf.)





mean = $\eta_1 + \eta_2$
variance = $\sigma_1^2 + \sigma_2^2$

Addition of multiple noise sources:

Suppose that $e_0(n)$, $e_1(n)$, …, $e_{N-1}(n)$ are N uncorrelated noise sources each with mean value η and variance σ^2. The mean and variance of $e(n) = e_0(n) + e_1(n) + \ldots + e_{N-1}(n)$ are $N\eta$ and $N\sigma^2$, respectively. The probability density function of $e(n)$ approaches Gaussian as N approaches infinity (central limit theorem).

mean = $N\eta$
variance = $N\sigma^2$

Power spectral density of sum of uncorrelated noise sources:

Suppose that $e_0(n)$ and $e_1(n)$ are uncorrelated noise sources with auto-correlations $R_i(m)$, $i = 0{,}1.$

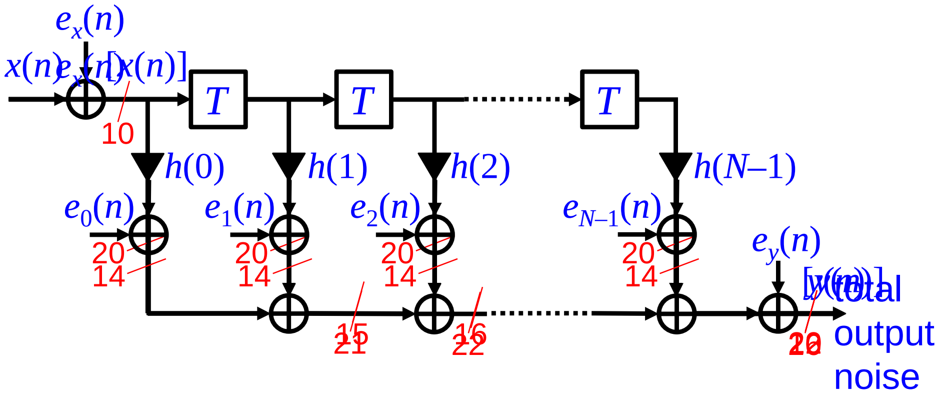
Then, auto-correlation of the sum $e(n) = e_0(n) + e_1(n)$ is

$$\begin{aligned} R(m) &= E\{e(n)e(n+m)\} \\ &= E\{\left(e_0(n)+e_1(n)\right)\left(e_0(n+m)+e_1(n+m)\right)\} \\ &= E\{e_0(n)e_0(n+m)\} + E\{e_0(n)e_1(n+m)\} \\ &\quad +E\{e_1(n)e_0(n+m)\} + E\{e_1(n)e_1(n+m)\} \\ &= R_0(m) + 0 + 0 + R_1(m) \end{aligned}$$

Taking Fourier transform, power spectral density of the sum is

$$S(e^{j\omega}) = S_0\left(e^{j\omega}\right) + S_1\left(e^{j\omega}\right)$$

The sum of N white noises is another white noise.

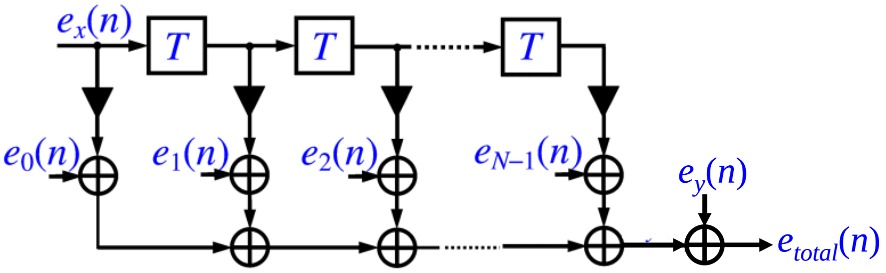


FIR filter: infinite precision
input quantization
multiplier quantization
output quantization

What is the total output noise?

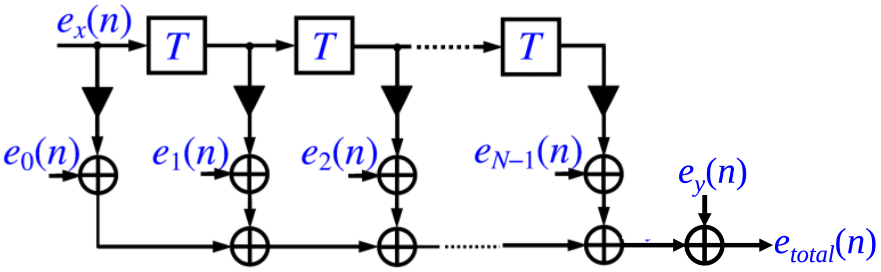
Output noise of an FIR filter

$e_x(n)$=input quantization noise, $e_0(n)\ldots e_{N-1}(n)$=multiply-add quantization noises, $e_y(n)$=output quantization noise, all uncorrelated. The total output noise of the filter is $e_{total}(n)$.



Denote each mean by $\eta_x, \eta_0, \ldots \eta_{N-1}, \eta_y, \eta_{total}$.
Then η_{total} is η_x filtered by $H(e^{j\omega})$, plus the means $\eta_0 \ldots \eta_{N-1}, \eta_y$.

$$\eta_{total} = H\left(e^{j0}\right)\eta_x + \eta_0 + \cdots + \eta_{N-1} + \eta_y$$



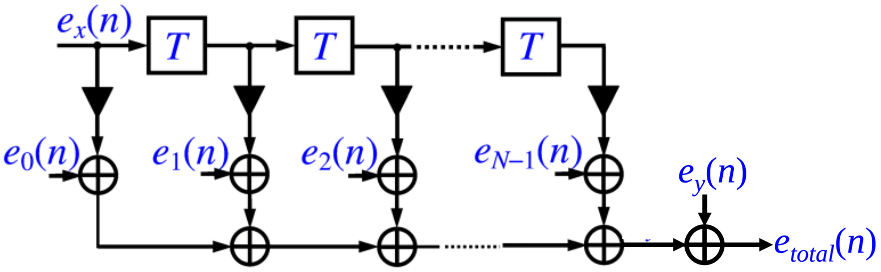
Denote each variance by $\sigma_x^2, \sigma_0^2, \ldots \sigma_{N-1}^2, \sigma_y^2, \sigma_{total}^2$.

Then σ_{total}^2 is σ_x^2 filtered by $H(e^{j\omega})$, plus the variances $\sigma_0^2, \ldots \sigma_{N-1}^2, \sigma_y^2$.

$$\sigma_{total}^2 = \left\|H\left(e^{j\omega}\right)\right\|_2^2 \sigma_x^2 + \sigma_0^2 + \cdots + \sigma_{N-1}^2 + \sigma_y^2$$

If $e_x(n)$ is from quantization step size Q_x, $e_0(n)$ to $e_{N-1}(n)$ are from quantization step size Q, and $e_y(n)$ is from quantization step size Q_y, then

$$\sigma_{total}^2 = \left\|H\left(e^{j\omega}\right)\right\|_2^2 \frac{Q_x^2}{12} + \frac{NQ^2}{12} + \frac{Q_y^2}{12}$$



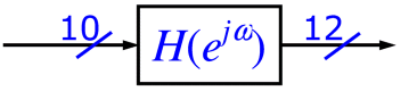
Denote each power spectral density by $S_x(e^{j\omega}), S_0(e^{j\omega}), \ldots S_{N-1}(e^{j\omega}), S_y(e^{j\omega}), S_{total}(e^{j\omega})$. Then $S_{total}(e^{j\omega})$ is $S_x(e^{j\omega})$ filtered by $H(e^{j\omega})$ plus $S_0(e^{j\omega}),..,S_{N-1}(e^{j\omega})$, $S_y(e^{j\omega})$.

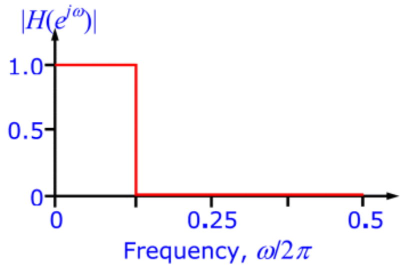
$$S_{total}(e^{j\omega}) = |H(e^{j\omega})|^2 S_x(e^{j\omega}) + S_0(e^{j\omega}) + \cdots + S_{N-1}(e^{j\omega}) + S_y(e^{j\omega})$$

If $e_x(n)$ is white noise from rounding with quantization step size Q_x, $e_0(n)$ to $e_{N-1}(n)$ are white noises from rounding with quantization step size Q, and $e_y(n)$ is white noise from rounding with quantization step size Q_y, then $S_{total}(e^{j\omega}) = |H(e^{j\omega})|^2 \frac{Q_x^2}{12} + \frac{NQ^2}{12} + \frac{Q_y^2}{12}$

Example:

Consider an example where an input 10-bit signal is filtered by $H(e^{j\omega})$. The output is rounded to 12 bits. The frequency response of the filter is shown below.



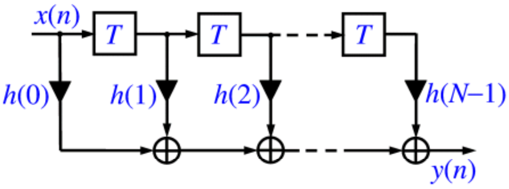


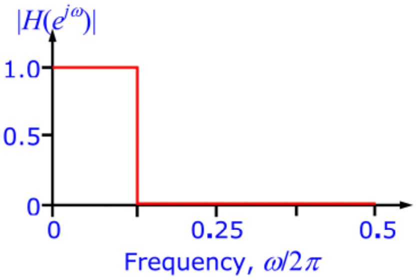
Let the quantization step size of the input signal be Q. The quantization noise variance of the input signal is $Q^2/12$. The bandwidth of the filter is ¼ of the full band bandwidth. Hence, total output noise variance due to input quantization noise is $(Q^2/12)\times¼ = Q^2/(4\times12)$.

Further, the total output noise variance due to rounding the output to 12 bits is $\frac{(Q/4)^2}{12} = \frac{Q^2}{16\times 12}$.

Hence, the total output noise variance= $\frac{Q^2}{4\times 12} + \frac{Q^2}{16\times 12} = \frac{5Q^2}{16\times 12}$

Example





Input quantization step size = Q.

Output quantization step size = Q.

Multiplier output quantization step size = $Q/2^4$ = $Q/16$, i.e. the output of each multiplier has a word-length that is 4 bits longer than the output signal word-length.

Filter length, N = 64.

Find total noise variance at the output.

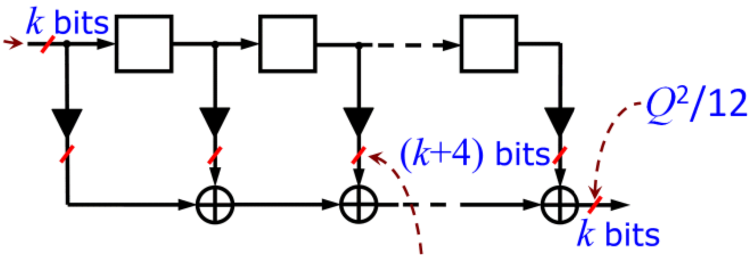
Input quantization noise variance = $Q^2/12$.

The input quantization noise is filtered by the filter.

Total output noise variance due to input quantization noise =

$\underbrace{Q^2/12} \times ¼ = Q^2/48$

because bandwidth of $H(e^{j\omega})$ = ¼ of full-band.



$Q/16$ because quantization step size is $Q/16$.

Total output noise variance for each multiplier output = $(Q/16)^2/12 = Q^2/3072$.

Since the filter length is 64, there are 64 multipliers. Hence, total output noise variance due to 64 multipliers is $64 \times Q^2/3072 = Q^2/48$.

Hence, the total output noise variance

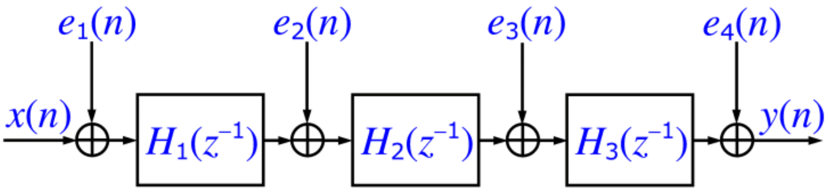
$= Q^2/48 + Q^2/48 + Q^2/12 = 3Q^2/24$.

due to input quantization

due to multiplier output quantization

due to output quantization

Quantization noise in cascaded systems

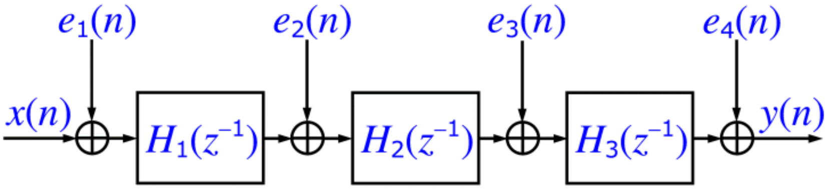


The power spectral density of $e_1(n)$ to $e_4(n)$ is $S_1(e^{j\omega})$ to $S_4(e^{j\omega})$.

Since $e_1(n)$ is filtered by $H_1(e^{j\omega})H_2(e^{j\omega})H_3(e^{j\omega})$, its power spectral density at the output is

$$S_1(e^{j\omega})\left|H_1(e^{j\omega})H_2(e^{j\omega})H_3(e^{j\omega})\right|^2$$
$$= S_1(e^{j\omega})\left|H_1(e^{j\omega})\right|^2\left|H_2(e^{j\omega})\right|^2\left|H_3(e^{j\omega})\right|^2$$

using $S_{out}(e^{j\omega}) = \left|H(e^{j\omega})\right|^2 S_{in}(e^{j\omega})$



Similarly, power spectral density of $e_2(n)$ and $e_3(n)$ at the output are $S_2(e^{j\omega})\left|H_2(e^{j\omega})\right|^2\left|H_3(e^{j\omega})\right|^2$ and $S_3(e^{j\omega})\left|H_3(e^{j\omega})\right|^2$.

Hence, the total noise power spectral density at the output is

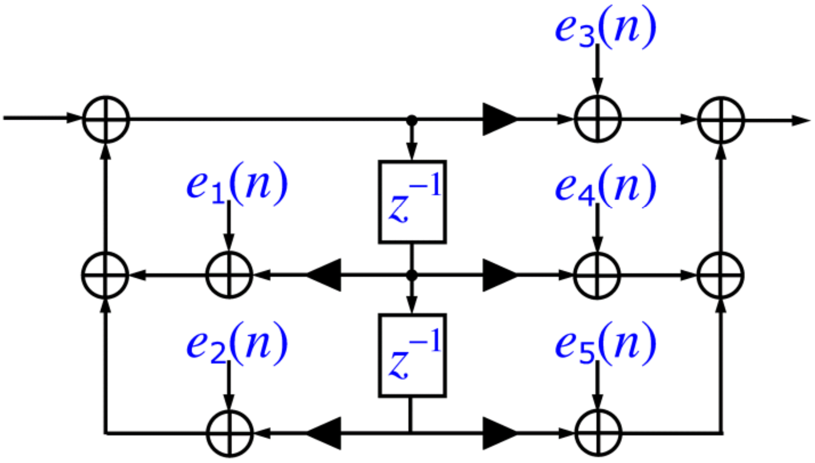
$$S_{total}(e^{j\omega}) = \left\{\left[\left(S_1(e^{j\omega})\left|H_1(e^{j\omega})\right|^2 + S_2(e^{j\omega})\right)\left|H_2(e^{j\omega})\right|^2 + S_3(e^{j\omega})\right]\right\} \times \left|H_3(e^{j\omega})\right|^2 + S_4(\omega)$$

If all quantization involves rounding, then all quantization noises are white with power spectral densities $S_i(e^{j\omega}) = \sigma_i^2$, then total noise power spectral density at the output is

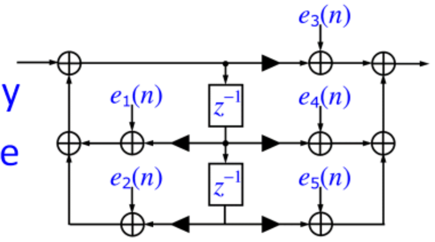
$$S_{total}(e^{j\omega}) = \left\{\left(\sigma_1^2\left|H_1(e^{j\omega})\right|^2 + \sigma_2^2\right)\left|H_2(e^{j\omega})\right|^2 + \sigma_3^2\right\}\left|H_3(e^{j\omega})\right|^2 + \sigma_4^2$$

Quantization noise model for IIR filter

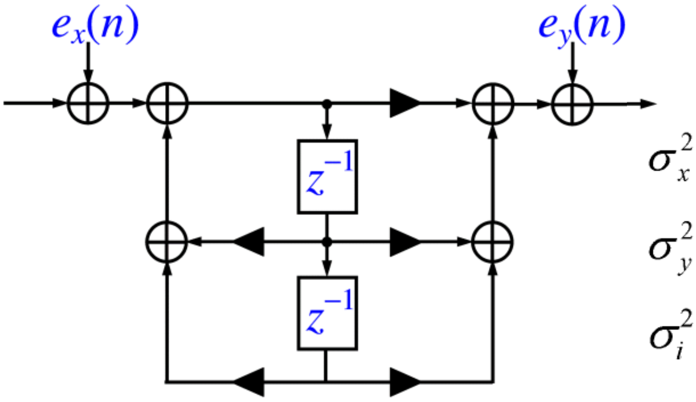
A double word-length product is obtained when a signal data is multiplied by a coefficient data. Suppose that the product is quantized after each multiplication producing a noise $e(n)$. The noise model is shown below.



Let $e_x(n) = e_1(n) + e_2(n)$ and $e_y(n) = e_3(n) + e_4(n) + e_5(n)$



$e_1(n)$ and $e_2(n)$ may be replaced by $e_x(n)$. $e_3(n)$ + $e_4(n)$ + $e_5(n)$ may be replaced by $e_y(n)$ as shown.

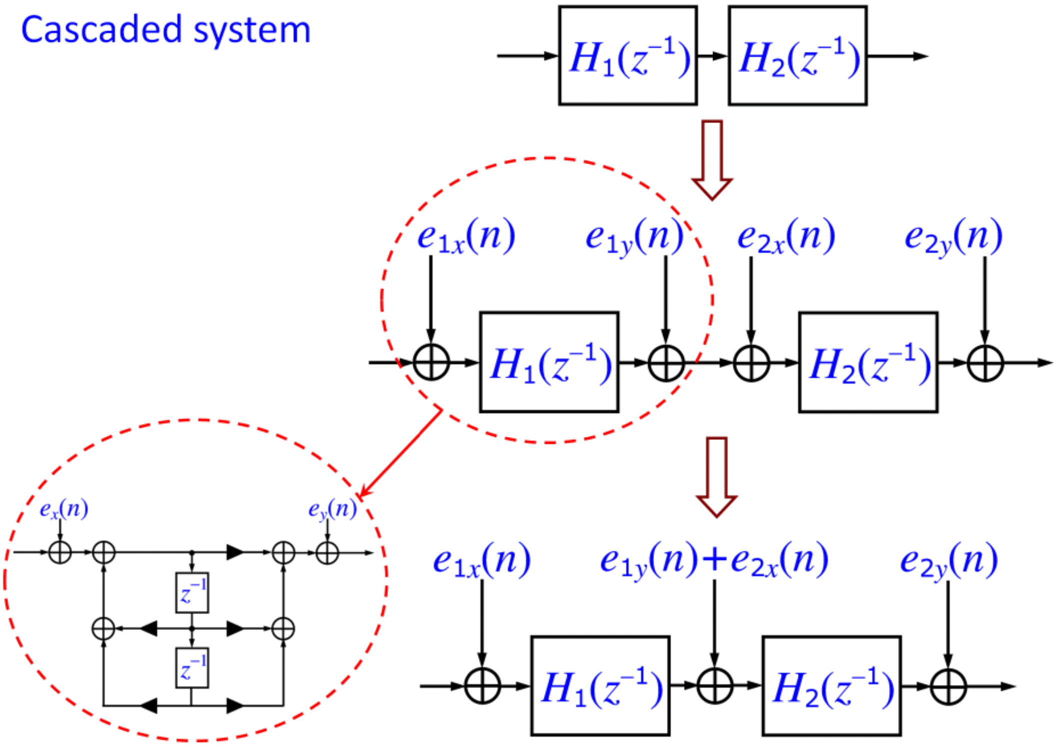


$$\sigma_x^2 = \sigma_1^2 + \sigma_2^2$$

$$\sigma_y^2 = \sigma_3^2 + \sigma_4^2 + \sigma_5^2$$

σ_i^2 is the noise variance of $e_i(n)$.

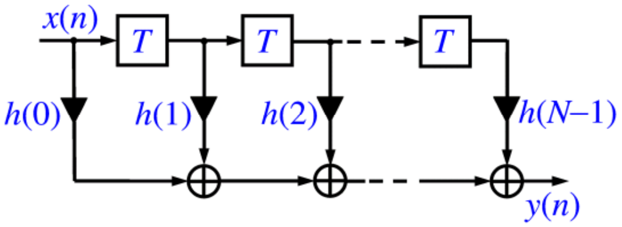
Cascaded system



Finite Word-Length Effects

In the implementation of a digital filter, the coefficient values must be quantized, i.e. the coefficient values used in actual implementation is not the same as the designed value unless the design process produces discrete coefficient values. The performance of the discrete coefficient filter may not be the same as that of the infinite precision coefficient filter.

Coefficient quantization effect in FIR filter



Consider an FIR filter with length N and z-transform transfer function $H(z)$ given by

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

where $h(n)$ is the impulse response at time n.

Suppose that $h(n)$ is rounded to a discrete value $[h(n)]$. Let the error of rounding $h(n)$ to $[h(n)]$ be $e(n)$, i.e.

$$[h(n)] = h(n) + e(n)$$

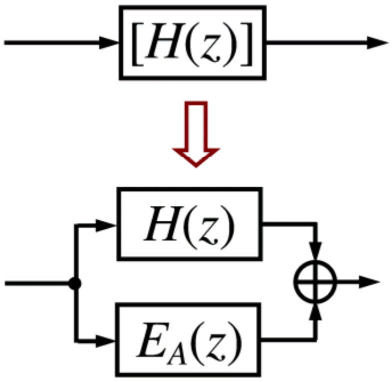
Let $[H(z)]$ be the z-transform transfer function of the filter with discrete value impulse response $[h(n)]$.

$$\begin{aligned}[H(z)] = \sum_{n=0}^{N-1}[h(n)]z^{-n} &= \sum_{n=0}^{N-1}\big(h(n)+e(n)\big)z^{-n} \\ &= \sum_{n=0}^{N-1}h(n)z^{-n} + \sum_{n=0}^{N-1}e(n)z^{-n} = H(z) + E_A(z)\end{aligned}$$

where $E_A(z) = \sum_{n=0}^{N-1}e(n)z^{-n}$ is the z-transform of $e(n)$.

$$[H(z)] = H(z) + E_A(z)$$

Hence, the transfer function $[H(z)]$ of the discrete coefficient value filter is equal to the transfer function $H(z)$ of the infinite precision filter in parallel with an error transfer function $E_A(z)$.



$E_A(e^{j\omega}) = \sum_{n=0}^{N-1} e(n)e^{-j\omega n}$ is the difference between the frequency response $[H(e^{j\omega})]$ of the discrete coefficient filter and the frequency response $H(e^{j\omega})$ of the infinite precision filter.

Assume that the coefficient quantization step size is Q. Assume that $[h(n)]$ is obtained by rounding $h(n)$ to its nearest discrete value. Hence, $|e(n)| \le \frac{Q}{2}$. We have,

$$|E_A(e^{j\omega})| = \left|\sum_{n=0}^{N-1} e(n)e^{-j\omega n}\right| \le \sum_{n=0}^{N-1} \left|e(n)e^{-j\omega n}\right| = \sum_{n=0}^{N-1} |e(n)| \le \sum_{n=0}^{N-1} \frac{Q}{2} = \frac{NQ}{2}$$

Thus, an absolute bound for $|E_A(e^{j\omega})|$ is $|E_A(e^{j\omega})| \le NQ/2$.

The bound $|E_A(e^{j\omega})| \le NQ/2$ is overly pessimistic for most applications.

At any frequency ω, $E_A(e^{j\omega})$ is obtained by summing N complex random variables each with magnitude bounded by $Q/2$.

$$E_A(e^{j\omega}) = \sum_{n=0}^{N-1} e(n)e^{-j\omega n}$$

A more realistic and useful bound will be a statistical bound. To derive such a bound, we shall first determine the variance of $E_A(e^{j\omega})$.

$$E_A\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} e(n)e^{-j\omega n}$$

$$\begin{aligned}\left|E_A\left(e^{j\omega}\right)\right|^2 &= E_A\left(e^{j\omega}\right)E_A\left(e^{-j\omega}\right)\\ &= \sum_{n=0}^{N-1} e(n)e^{-j\omega n}\sum_{m=0}^{N-1} e(m)e^{j\omega m}\\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} e(n)e^{-j\omega n}e(m)e^{j\omega m}\\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} e(n)e(m)e^{j\omega(m-n)}\end{aligned}$$

$$\begin{aligned}E\left\{\left|E_A\left(e^{j\omega}\right)\right|^2\right\} &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} E\{e(n)e(m)\}e^{j\omega(m-n)}\\ &= NE\{e(n)^2\} = \frac{NQ^2}{12}\end{aligned}$$

=0 if $n \neq m$, $e(n)$ is uncorrelated by assumption.

$$E\left\{\left|E_A\left(e^{j\omega}\right)\right|^2\right\} = \frac{NQ^2}{12}$$

The rms value (standard deviation) of $E_A(e^{j\omega})$ denoted by σ_{EA} is given by $\sigma_{EA} = \sqrt{E\{|E_A(e^{j\omega})|^2\}} = Q\sqrt{\frac{N}{12}} = \frac{Q}{2}\sqrt{\frac{N}{3}}$

$|E_A(e^{j\omega})| \leq 2\sigma_{EA}$ with high probability.

The statistical bound $|E_A(e^{j\omega})| \leq 2\sigma_{EA}$ is a more realistic and useful bound than the absolute bound $|E_A(e^{j\omega})| \leq NQ/2$.

Coefficient quantization effect in IIR filter

Let $H(z)$ be the transfer function of the infinite precision IIR filter:

$$H(z) = \frac{\sum_{n=0}^{N} a_n z^{-n}}{1 + \sum_{n=1}^{N} b_n z^{-n}} = \frac{A(z)}{B(z)}$$

Let $X(z)$ be the input and $Y(z)$ be the infinite precision output:

$$\frac{Y(z)}{X(z)} = H(z) \qquad B(z)Y(z) = A(z)X(z)\ldots.(1)$$

a_n and b_n are quantized to discrete values $[a_n]$ and $[b_n]$:

$$[a_n] = a_n + \alpha_n, \qquad [A(z)] = A(z) + \alpha(z)$$

$$[b_n] = b_n + \beta_n, \qquad [B(z)] = B(z) + \beta(z)$$

Let $[H(z)]$ be the transfer function and $[Y(z)]$ be the output of the quantized coefficient filter: $\dfrac{[Y(z)]}{X(z)} = [H(z)]$

$$\{B(z) + \beta(z)\}[Y(z)] = \{A(z) + \alpha(z)\}X(z)\ldots.(2)$$

Subtracting (1) from (2),

$$B(z)\{[Y(z)] - Y(z)\} + \beta(z)[Y(z)] = \alpha(z)X(z)$$

Let $V(z)$ be the difference between $[Y(z)]$ and $Y(z)$:

$$V(z) = [Y(z)] - Y(z)$$

$$B(z)V(z) + \beta(z)\{V(z) + Y(z)\} = \alpha(z)X(z)$$

Since both $\beta(z)$ and $V(z)$ are error terms, $\beta(z)V(z)$ is a second order term that may be neglected.

$$B(z)V(z) + \beta(z)Y(z) = \alpha(z)X(z)$$

$$\begin{aligned} V(z) &= \frac{\alpha(z)}{B(z)}X(z) - \frac{\beta(z)Y(z)}{B(z)} \\ &= \frac{\alpha(z) - \beta(z)H(z)}{B(z)}X(z) \end{aligned}$$

Let $E_R(z)$ be the difference between $[H(z)]$ and $H(z)$:

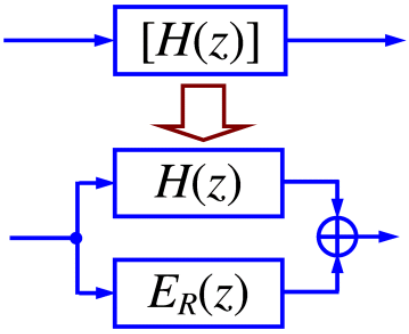
$$\begin{aligned} E_R(z) &= [H(z)] - H(z) \\ &= \frac{[Y(z)]}{X(z)} - \frac{Y(z)}{X(z)} \\ &= \frac{V(z)}{X(z)} \\ &= \frac{\alpha(z) - H(z)\beta(z)}{B(z)} \end{aligned}$$

Thus, the transfer function of the finite precision coefficient filter $[H(z)]$ is that of the infinite precision coefficient filter $H(z)$ connected in parallel with an error transfer function $E_R(z)$ given by

$$E_R(z) = \frac{V(z)}{X(z)} = \frac{\alpha(z) - H(z)\beta(z)}{B(z)}$$

where $\alpha(z) = \sum_{m=0}^{N} \alpha_m z^{-m}$

$$\beta(z) = \sum_{m=1}^{N} \beta_m z^{-m}$$



We shall assume that α_m and β_m are rectangularly distributed with zero mean and variance = $Q^2/12$. We have shown that

$$E_R(e^{j\omega}) = \frac{\sum_{m=0}^{N}\alpha_m e^{-jm\omega} - H(e^{j\omega})\sum_{m=1}^{N}\beta_m e^{-jm\omega}}{B(e^{j\omega})}$$

. Hence,

$$\left|E_R(e^{j\omega})\right|^2 = \frac{\sum_{m=0}^{N}\alpha_m e^{-jm\omega} - H(e^{j\omega})\sum_{m=1}^{N}\beta_m e^{-jm\omega}}{B(e^{j\omega})} \times \frac{\sum_{m=0}^{N}\alpha_m e^{jm\omega} - H(e^{-j\omega})\sum_{m=1}^{N}\beta_m e^{jm\omega}}{B(e^{-j\omega})}$$

$$= \frac{\sum_{m=0}^{N}\sum_{n=0}^{N}\alpha_m\alpha_n e^{j(n-m)\omega} + \left|H(e^{j\omega})\right|^2\sum_{m=1}^{N}\sum_{n=1}^{N}\beta_m\beta_n e^{j(n-m)\omega}}{\left|B(e^{j\omega})\right|^2}$$

$$- \frac{H(e^{-j\omega})\sum_{m=0}^{N}\sum_{n=1}^{N}\alpha_m\beta_n e^{j(n-m)\omega} + H(e^{j\omega})\sum_{m=1}^{N}\sum_{n=0}^{N}\beta_m\alpha_n e^{j(n-m)\omega}}{\left|B(e^{j\omega})\right|^2}$$

$$E\left\{\left|E_R(e^{j\omega})\right|^2\right\} = \frac{\sum_{m=0}^{N}\sum_n E\{\alpha_m\alpha_n\}e^{j(n-m)\omega}}{|B(e^{j\omega})|^2}$$

= $Q^2/12$ if $m = n$

= 0 if $m \neq n$

$$+ \frac{\left|H(e^{j\omega})\right|^2\sum_{m=1}^{N}\sum_n E\{\beta_m\beta_n\}e^{j(n-m)\omega}}{|B(e^{j\omega})|^2}$$

$$- \frac{H(e^{-j\omega})\sum_m\sum_n E\{\alpha_m\beta_n\}e^{j(n-m)\omega}}{|B(e^{j\omega})|^2}$$

= 0

$$- \frac{H(e^{j\omega})\sum_m\sum_n E\{\beta_m\alpha_n\}e^{j(n-m)\omega}}{|B(e^{j\omega})|^2}$$

$$= \frac{\frac{(N+1)Q^2}{12} + \left|H(e^{j\omega})\right|^2\frac{NQ^2}{12}}{|B(e^{j\omega})|^2} = \frac{Q^2}{12} \times \frac{1 + N\left(1 + \left|H(e^{j\omega})\right|^2\right)}{|B(e^{j\omega})|^2}$$

$$E\left\{\left|E_R\left(e^{j\omega}\right)\right|^2\right\} = \frac{Q^2}{12} \times \frac{1 + N\left(1 + \left|H\left(e^{j\omega}\right)\right|^2\right)}{\left|B\left(e^{j\omega}\right)\right|^2}$$

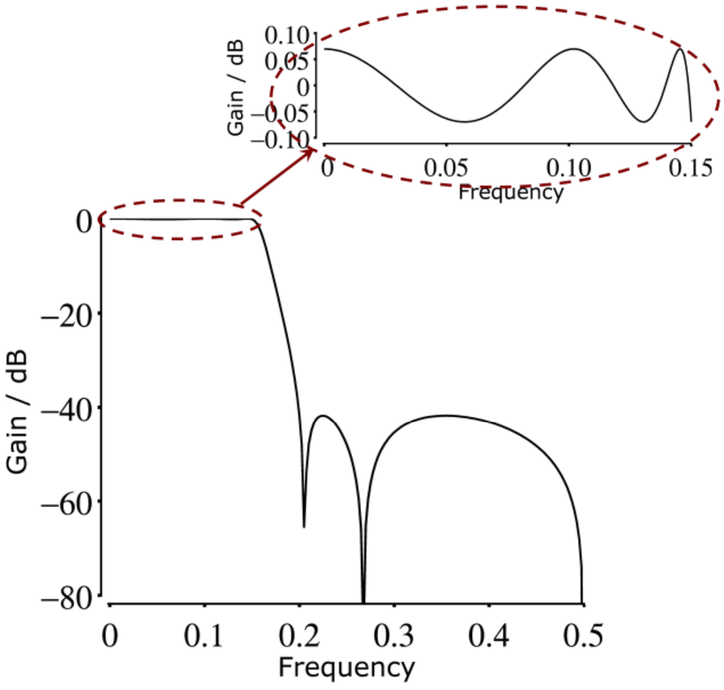
In the pass-band, $|H(e^{j\omega})| \approx 1$ and in the stop-band $|H(e^{j\omega})| \approx 0$. Thus,

$$E\left\{\left|E_R\left(e^{j\omega}\right)\right|^2\right\} = \frac{Q^2}{12} \times \frac{2N+1}{\left|B(e^{j\omega})\right|^2} \quad \text{in the pass-band}$$

$$= \frac{Q^2}{12} \times \frac{N+1}{\left|B(e^{j\omega})\right|^2} \quad \text{in the stop-band}$$

The difference between $2N+1$ and $N+1$ in the numerator is insignificant when compared to the factor $1/|B(e^{j\omega})|^2$.

To have an idea on the factor $1/|B(e^{j\omega})|^2$, consider the following 5th order elliptic filter.



$$H(z) = \frac{\sum_{n=0}^{N} a_n z^{-n}}{1 + \sum_{n=1}^{N} b_n z^{-n}}$$

$a_0 = 0.030516$

$a_1 = 0.020291$

$a_2 = 0.047142$

$a_3 = 0.047142$

$a_4 = 0.020291$

$a_5 = 0.030516$

$b_1 = -2.556875$

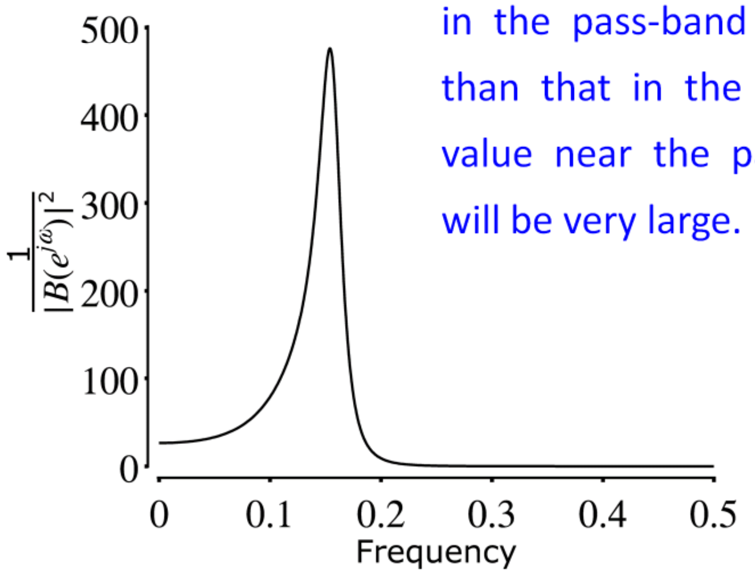
$b_2 = 3.452760$

$b_3 = -2.626267$

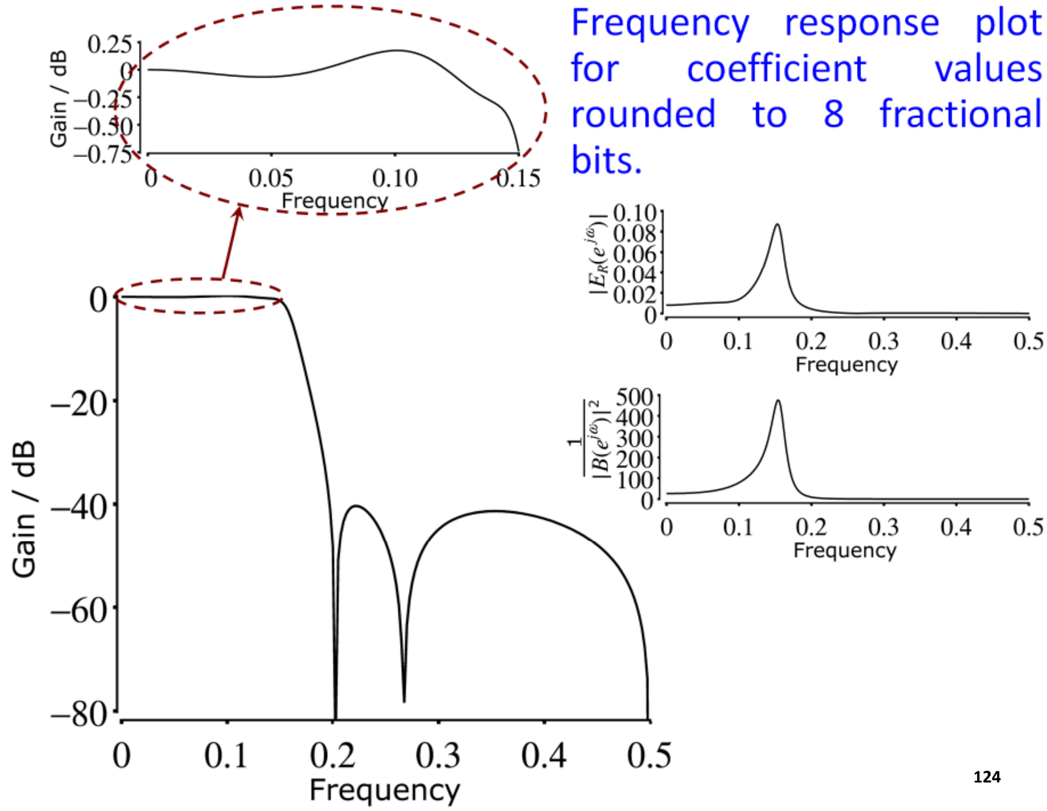
$b_4 = 1.148664$

$b_5 = -0.223953$

We note that $1/|B(e^{j\omega})|^2$ is small in the stop-band, moderately large in the pass-band, and very large near the pass-band edge. Hence, the value of $E\left\{\left|E_R\left(e^{j\omega}\right)\right|^2\right\}$ in the pass-band will be larger than that in the stop-band. Its value near the pass-band edge will be very large.



Frequency response plot for coefficient values rounded to 8 fractional bits.


Gain / dB
0.25
0
−0.25
−0.50
−0.75
0
0.05
0.10
0.15
Frequency
Gain / dB
0
−20
−40
−60
−80
0
0.1
0.2
0.3
0.4
0.5
Frequency
$|E_R(e^{j\omega})|$
0.10
0.08
0.06
0.04
0.02
0
0
0.1
0.2
0.3
0.4
0.5
Frequency
$\frac{1}{|B(e^{j\omega})|^2}$
500
400
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
Frequency