7. Quantization Noise in DSP

- *Reference*: Sections 6.6 to 6.9, 9.7 of Text

- In our discussions of the filtering and FFT operations, it was implicitly assumed that all the computations can be performed with infinite precision. This is quite true with floating point implementation.

- With fixed point implementation,

  - the filter parameters will be quantized (finite word length representation) and
  - any intermediate result arising from the multiplication and/or addition of numbers will be quantized (finite precision computation)

  Some effects of quantization are:

  - it changes the locations of the poles and zeros of a filter,
  - it creates the equivalent of an additive noise term at the filter/FFT output .

7.1 Number Representation and Quantization

- In some DSP chips or special purpose hardware, numbers/signals are represented and manipulated using fixed-point binary arithmetic.

- Examples of fixed-point binary representations are *sign and magnitude, one's complement*, and *two's complement*.

  The two's complement representation is the most common format.

- In the two's complement numbering system, a real number x within the range

$$-X_m \le x \le X_m$$

  is quantized to the number

$$\hat{x} = X_m \left(-b_0 + \sum_{i=1}^{B} b_i 2^{-i} \right) = X_m \hat{x}_B ,$$

  where b_i, $i = 0,1,...,B$ are binary (0,1) numbers, with b_0 being the sign bit.

  With this numbering scheme, the **smallest** difference between numbers is

$$\Delta = X_m 2^{-B}$$

  and the fractional part of a number, i.e. the term

$$\hat{x}_B = -b_0 + \sum_{i=1}^{B} b_i 2^{-i} ,$$

  can be represented by the binary pattern

$$\hat{x}_B \equiv b_0 \circ b_1 b_2 ... b_B ,$$

  where $\circ$ is the binary point.

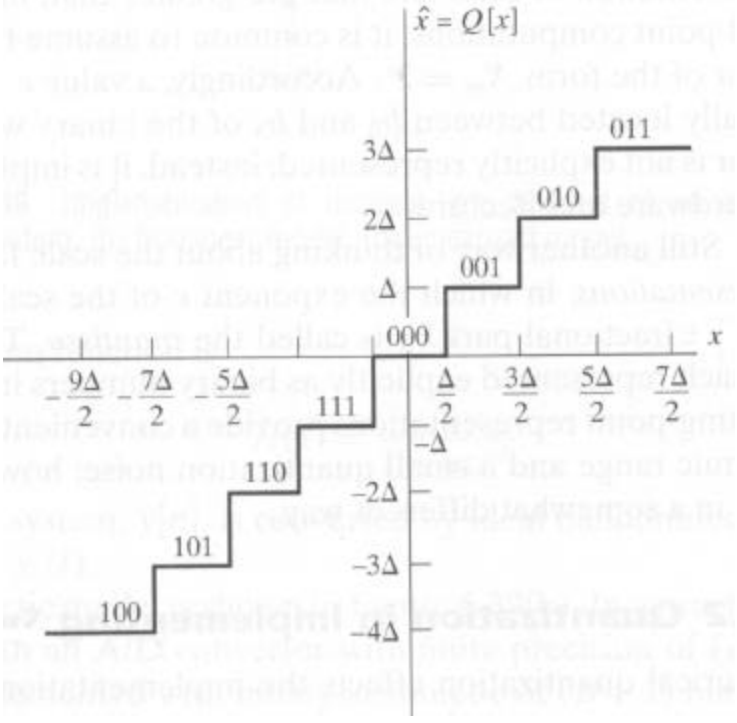
The number X_m must be large enough that the risk of overflow is small, and small enough that the quantization noise is kept to an acceptable level.

When $B \to \infty$, $\hat{x}$ becomes the real number x.

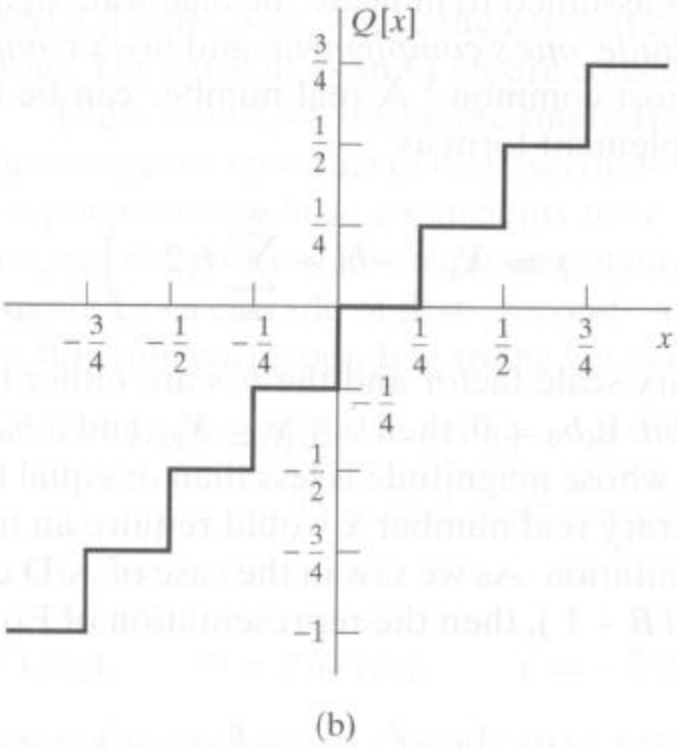
- The quantization of x to $\hat{x}$ can be done through

  1. rounding, or
  2. truncation.

  The figures below illustrate these two operations for the case of $B = 2$. Clearly the mapping from x to $\hat{x}$ is nonlinear in both cases.



rounding



(b)

truncation

- Let $Q_B[\bullet]$ denotes a $B+1$ bit quantizer and let the quantization error be

$$\begin{aligned} e &= \hat{x} - x \\ &= Q_B[x] - x \,. \end{aligned}$$

Then it can be easily deduced from the above figures that for rounding,

$$-\Delta/2 < e \le \Delta/2 \,,$$

and for truncation,

$$-\Delta < e \le 0 \,.$$

- In studying the effect of quantization, the quantization error is usually modelled as a uniform random variable. For rounding, the probability density function (pdf) of this random variable is

$$p(e) = \begin{cases} 1/\Delta & -\Delta/2 < e \le \Delta/2 \\ 0 & \text{otherwise} \end{cases},$$

and for truncation, the pdf is

$$p(e) = \begin{cases} 1/\Delta & -\Delta < e \le 0 \\ 0 & \text{otherwise} \end{cases}.$$

- The mean square quantization error for rounding is

$$s_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 p(e)\, de = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{\Delta^2}{12},$$

and for truncation is

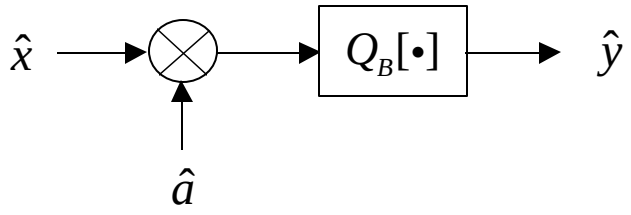
$$s_e^2 = \int_{-\Delta}^{0} e^2 p(e)\, de = \frac{1}{\Delta} \int_{-\Delta}^{0} e^2 de = \frac{\Delta^2}{3}.$$

Thus rounding is clearly more desirable than truncation.

- Consider the multiplication of the quantized numbers $\hat{x}$ and $\hat{a}$ during a DSP operation. The product,

$$y = \hat{x}\hat{a},$$

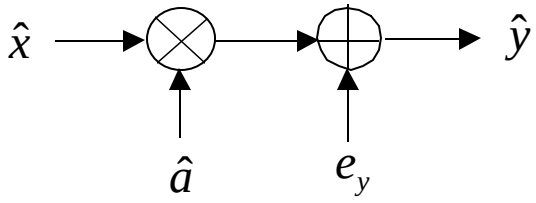
itself will be quantized to a $B+1$ number $\hat{y}=Q_B[y]$.



The quantized product can be written in terms of the unquantized product $\hat{x}\hat{a}$ and the quantization error e_y as

$$\hat{y}=Q_B\left[y\right]=y+e_y=\hat{x}\hat{a}+e_y$$

If the quantization error is modelled as a uniform random variable, the model for a fixed-point multiplier becomes



This linearlized model will be adopted in our study of the effect of rounding error in IIR filters later on.

7.2 The Effects of Coefficient Quantization

- The transfer function of an IIR filter can be written as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \frac{B(z)}{A(z)},$$

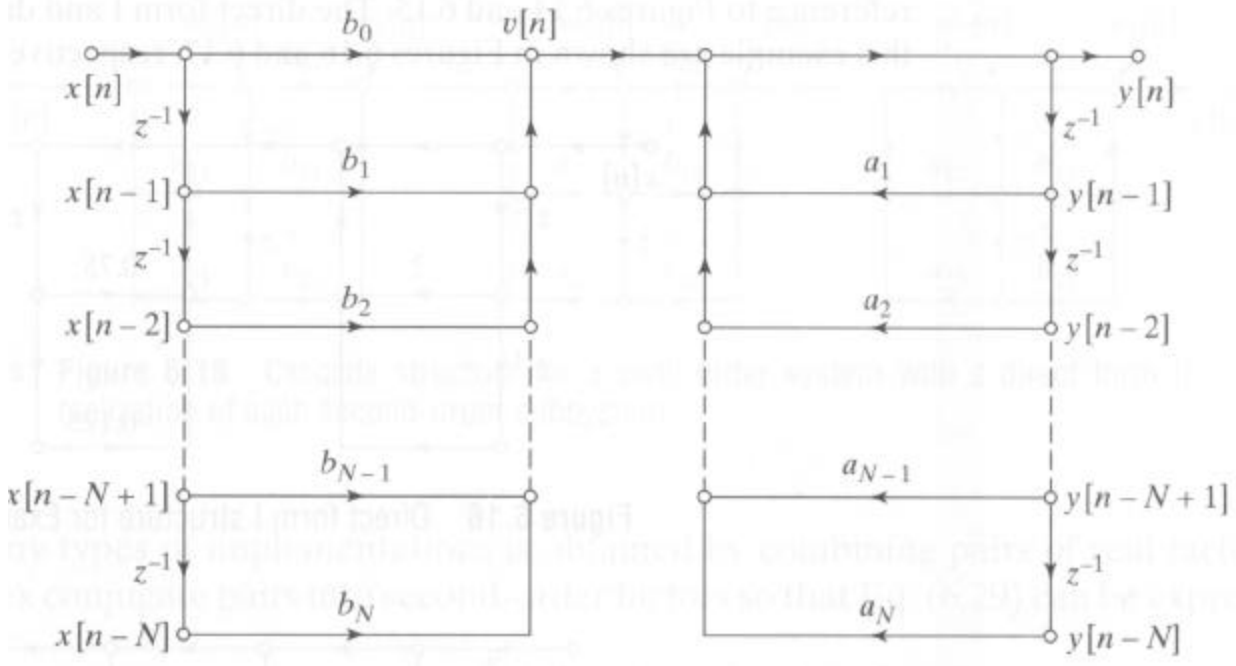
where

$$A(z) = 1 - \sum_{k=1}^{N} a_k z^{-k}$$

and

$$B(z) = \sum_{k=0}^{M} b_k z^{-k} .$$

The Direct form implementation of this filter is shown below. All the filter coefficients are at their original unquantized values.



We want to address in this section the issue of poles and zeros relocation when the filter coefficients are quantized.

- Consider the denominator polynomial $A(z)$. This polynomial can be written in product form as

$$A(z) = 1 - \sum_{k=1}^{N} a_k z^{-k} = \prod_{k=1}^{N} \left(1 - p_k z^{-1}\right),$$

where the p_k's are the roots of $A(z)$, or equivalently the poles of $H(z)$. It is understood that the p_k's are nonlinear functions of the a_k's. For example when $N = 2$, then $a_1 = p_1 + p_2$ and $a_2 = -p_1 p_2$.

So how would the p_k's be affected when the a_k's are quantized?

- Lets take the partial derivative of $A(z)$ with respect to a_i. From the summation form, we obtain

$$\frac{\partial A(z)}{\partial a_i} = -z^{-i} \tag{1}$$

From the product form we obtain

$$\frac{\partial A(z)}{\partial a_i} = -\sum_{n=1}^{N} \left\{ \prod_{\substack{k=1 \\ k \neq n}}^{N} \left(1 - p_k z^{-1}\right) \right\} z^{-1} \frac{\partial p_n}{\partial a_i} \tag{2}$$

Suppose we want to determine $\partial p_j / \partial a_i$. What we can do is to evaluate both (1) and (2) at $z = p_j$ and equate them. The end result is

$$\frac{\partial p_j}{\partial a_i} = \frac{p_j^{N-i}}{\prod_{\substack{k=1 \\ k \neq j}}^{N} \left(p_j - p_k \right)}$$

- **Example**: When $N = 3$, $A(z)$ can be written as

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3}$$

or

$$A(z) = \left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right)\left(1 - p_3 z^{-1}\right).$$

The partial derivative of the first expression with respect to a_1 yields

$$\frac{\partial A(z)}{\partial a_1} = -z^{-1}.$$

On the other hand, the partial derivative of the second expression with respect to a_1 yields

$$\frac{\partial A(z)}{\partial a_1} = \left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right)\left(-\frac{\partial p_3}{\partial a_1} z^{-1} \right) + \left(1 - p_1 z^{-1}\right)\left(-\frac{\partial p_2}{\partial a_1} z^{-1} \right)\left(1 - p_3 z^{-1}\right) + \left(-\frac{\partial p_1}{\partial a_1} z^{-1} \right)\left(1 - p_2 z^{-1}\right)\left(1 - p_3 z^{-1}\right)$$

If we evaluate the two expressions at $z = p_3$ and equate them, we obtain

$$p_3^{-1} = \left(1 - p_1 p_3^{-1}\right)\left(1 - p_2 p_3^{-1}\right)\left(-\frac{\partial p_3}{\partial a_1} p_3^{-1} \right)$$

which can be simplified to

$$\frac{\partial p_3}{\partial a_1} = \frac{p_3^2}{(p_3 - p_1)(p_3 - p_2)}$$

- Let Δa_k, $k = 1,2,\ldots, N$, be the quantization errors in the a_k's when the IIR filter is implemented using the Direct form computational structure with fixed-point arithmetic. Then the poles will be shifted by the amounts

$$\Delta p_j = \sum_{i=1}^{N} \frac{\partial p_j}{\partial a_i} \Delta a_i; \qquad j = 1,2,\ldots, N$$

Because of the term

$$\prod_{\substack{k=1 \\ k \neq j}}^{N} (p_j - p_k)$$

in the denominator of $\partial p_j / \partial a_i$, we can deduce that if the poles are clustered together, i.e. when

$$|p_j - p_k| << 1,$$

there could be big changes to the poles' locations. Consequently, the direct form implementation structure is quite sensitive to quantization errors in the filter coefficients.

- As shown in Section 6.4, the cascade form of an IIR filter is made up of a serial concatenation of second order direct form subsystems. The poles

(zeros) of all the subsystems together form the poles (zeros) of the IIR filter.

Quantization of the filter coefficients in a subsystem will only affect the two poles (and 2 zeros) of that subsystem. In other word, quantization errors are localized.

Thus the cascade form is generally much less sensitive to coefficient quantization than the direct form.

- Let us focus on a 2nd order IIR filter of the form

$$H(z) = \frac{1}{\left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right)},$$

where

$$p_1 = re^{jq}$$

and

$$p_2 = re^{-jq}$$

are the conjugate pole-pair of the filter. The two poles are located on a circle with radius r, with one of them at phase angle of q and the other at an angle of $-q$.

The denominator polynomial can be written as

$$\begin{aligned} A(z) &= \left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right) \\ &= 1 - \left(p_1 + p_2\right) z^{-1} + p_1 p_2 z^{-2} \\ &= 1 - 2r\cos\left(q\right) z^{-1} + r^2 z^{-2} \end{aligned}$$

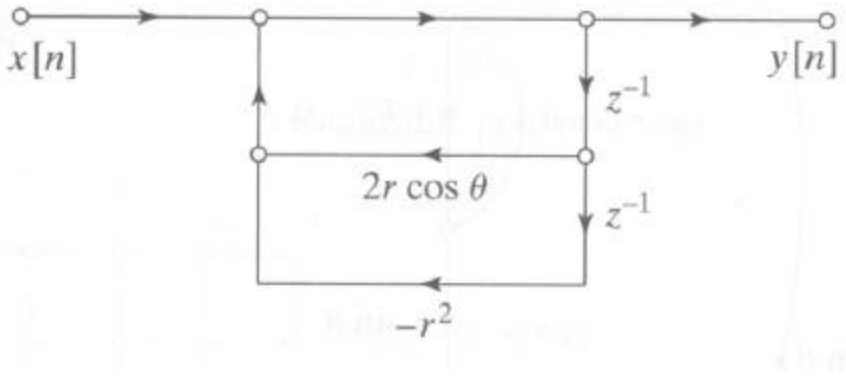
Consequently the coefficients in the feedback portion of the IIR filter are

$$a_1 = 2r\cos(q)$$

and

$$a_2 = -r^2$$

The implementation structure of this filter is as shown below

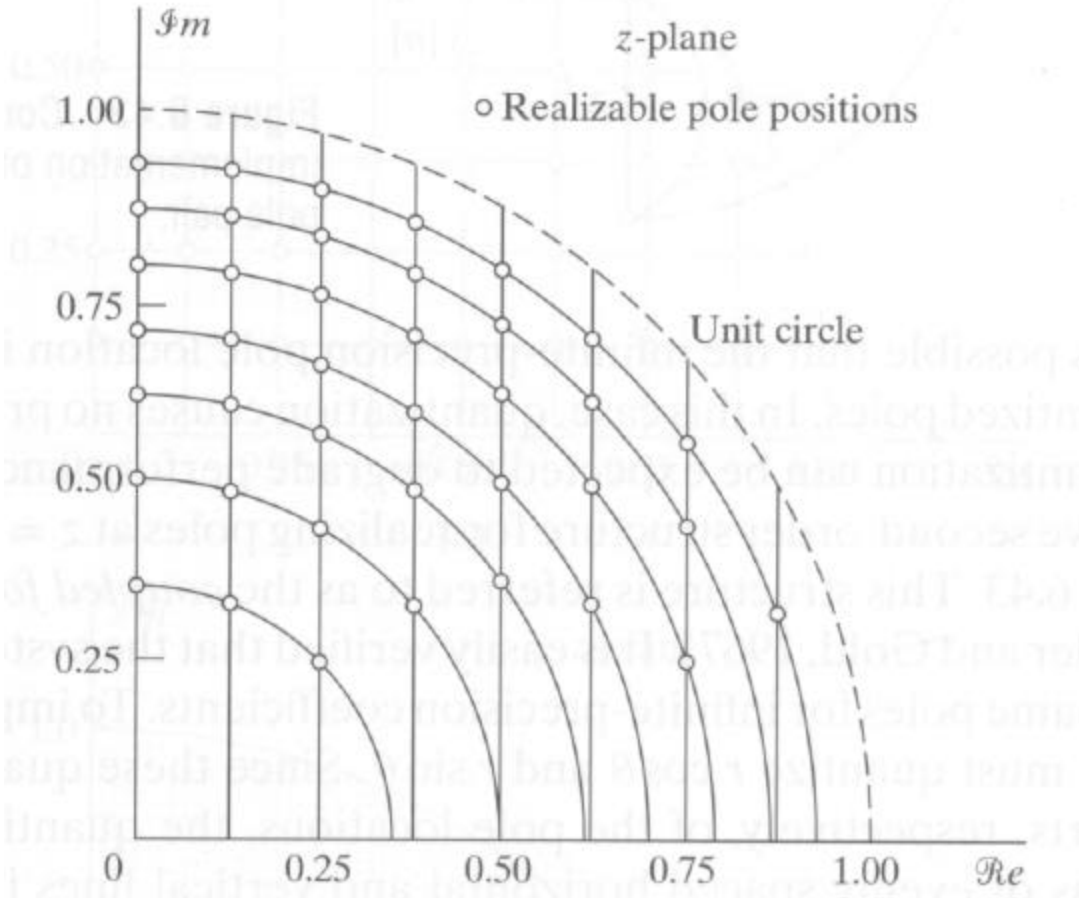


If we assume 4-bit quantization of a_1 and a_2 in the interval [-1,+1]. Then $-r^2$ and $2r\cos q$ are numbers from the set:

$$-1, -\tfrac{7}{8}, -\tfrac{3}{4}, -\tfrac{5}{8}, -\tfrac{1}{2}, -\tfrac{3}{8}, -\tfrac{1}{4}, -\tfrac{1}{8}, 0, \tfrac{1}{8}, \tfrac{1}{4}, \tfrac{3}{8}, \tfrac{1}{2}, \tfrac{5}{8}, \tfrac{3}{4}, \tfrac{7}{8}.$$

This means after quantization, the poles can only take on values from the set shown in diagram in the next page (only the first quadrant in the z-plane are shown).

From the diagram, we can deduce that poles that are originally around $|q| = 0$ or $|q| = p$ are more affected by quantization than those around $|q| = p/2$.



- **Exercise**: Determine

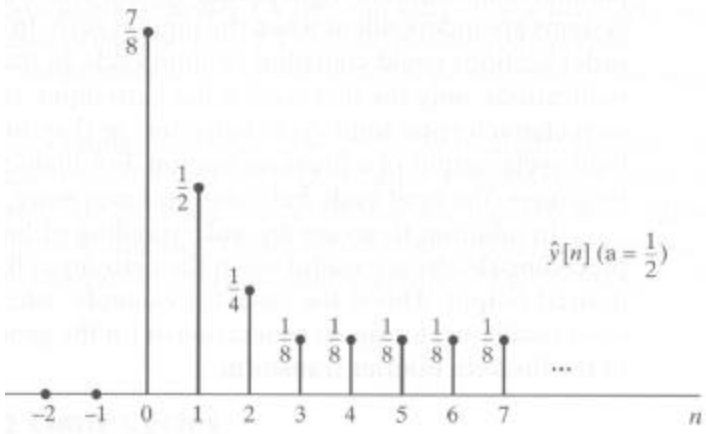
$$\frac{\partial p_1}{\partial a_1}, \frac{\partial p_1}{\partial a_2}, \frac{\partial p_2}{\partial a_1}, \text{ and } \frac{\partial p_2}{\partial a_2}$$

for the above second order IIR filter. Assuming the original poles are on a circle of radius $r = 0.96$ with phases of $q = \pm p/36$. Where are the locations of the new poles after 4-bit quantization?

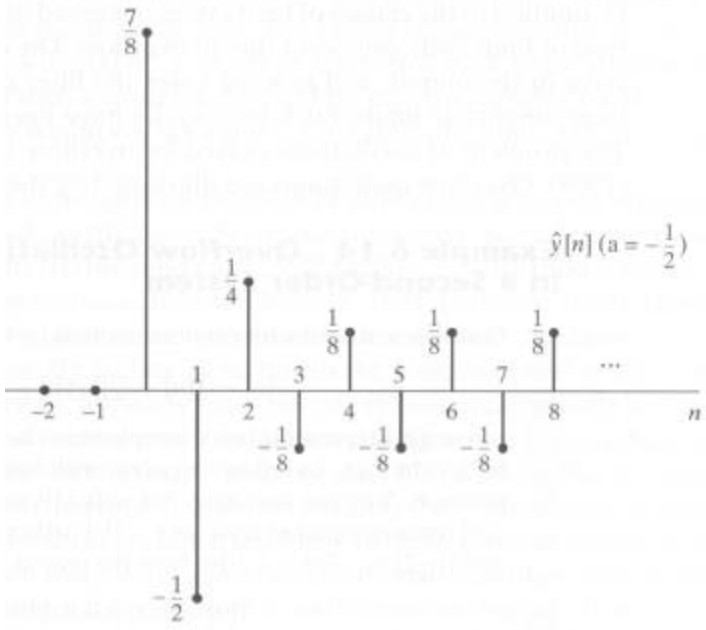
7.3 Zero Input Limit Cycle

- For a stable digital filter, if the input is set to zero at some point in time, the output should decay to zero.

- For finite precision implementations, the output may decay to a non-zero amplitude and then oscillate. This phenomenon is known as the *zero-input limit cycle*; see for example the following results for a first order IIR filter with 4-bit quantization between –1 and +1.



(a)



- Zero-input limit cycle can be very annoying for applications such as speech, as tones will be generated during the silence periods.

- Zero-input limit cycle is unique to IIR filters because of the feedback mechanism in these filters. No need to worry about this phenomenon for FIR filters.

- As an example, consider a first order IIR filter

$$y[n] = ay[n-1] + x[n]; \qquad |a| < 1$$

With $(B+1)$-bit fixed-point implementation, the output becomes

$$\hat{y}[n] = Q_B\left[a\hat{y}[n-1]\right] + x[n]; \qquad |a| < 1$$

Here, we assume rounding with a quantization step size of Δ.

Suppose the input becomes zero when $n \geq n_o$. Then the unquantized output at time n_o is $a\hat{y}[n_o - 1]$. Assuming that both $\hat{y}[n_o - 1]$ and the filter parameter a are positive, then $a\hat{y}[n_o - 1]$ will be quantized back to $\hat{y}[n_o - 1]$ if

$$a\hat{y}[n_o - 1] - \hat{y}[n_o - 1] \geq -\Delta/2$$

or when

$$\hat{y}[n_o - 1] \leq \frac{\Delta}{2(1-a)}; \qquad (\hat{y}[n_o - 1] > 0, a > 0)$$

In general, it can be shown that for a 1st order IIR filter, the zero-input limit cycle exsists when

$$\left|\hat{y}[n_o - 1]\right| \le \frac{\Delta}{2\left(1-\left|a\right|\right)}$$

This is known as the *dead band* of the first order IIR filter.

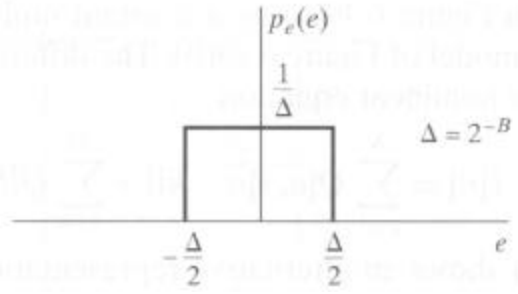
- Limit cycles can also be caused by overflow. In this case, the oscillation is between large limits.

- Limit cycles can be eliminated by adopting computation structures that do not support them. But these structures are usually more computationally intensive than the cascade form we discussed.

  More bits in the quantization process will reduce the chance of limit cycles.

7.4 Effects of Round-off Noise in IIR Filters

- In Section 7.1, we showed that a fixed-point multiplier can be replaced by an unconstrained (real) multiplier in concatenation with an additive, uniform noise source that models the round-off error.



PDF of round-off noise in a B+1 bit quantizer

- We want to analyze in this section the effect of the round-off noise on the output of an IIR filter. With infinite precision implementation, the input output relationship of such a filter is given by

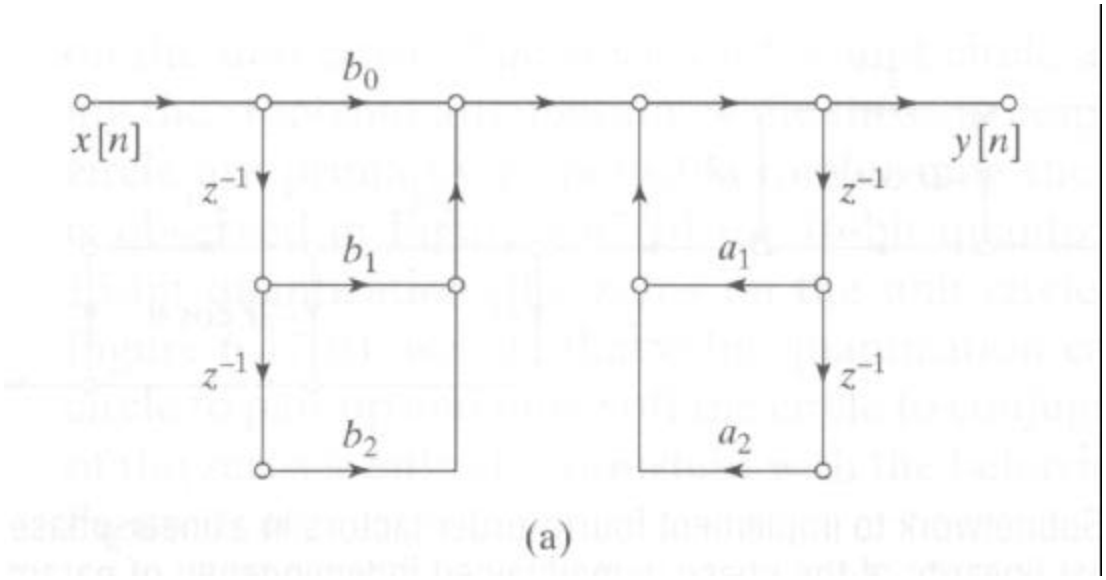
$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k].$$

With rounding, this becomes

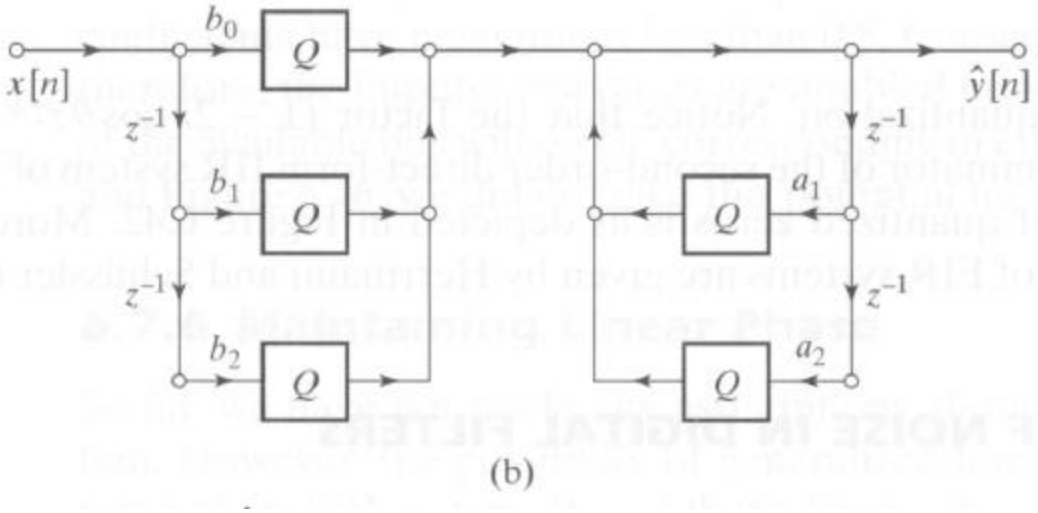
$$\hat{y}[n] = \sum_{k=1}^{N} Q_B\left[a_k \hat{y}[n-k]\right] + \sum_{k=0}^{M} Q_B\left[b_k x[n-k]\right],$$

where we assume that the input and the filter coefficients are already quantized.

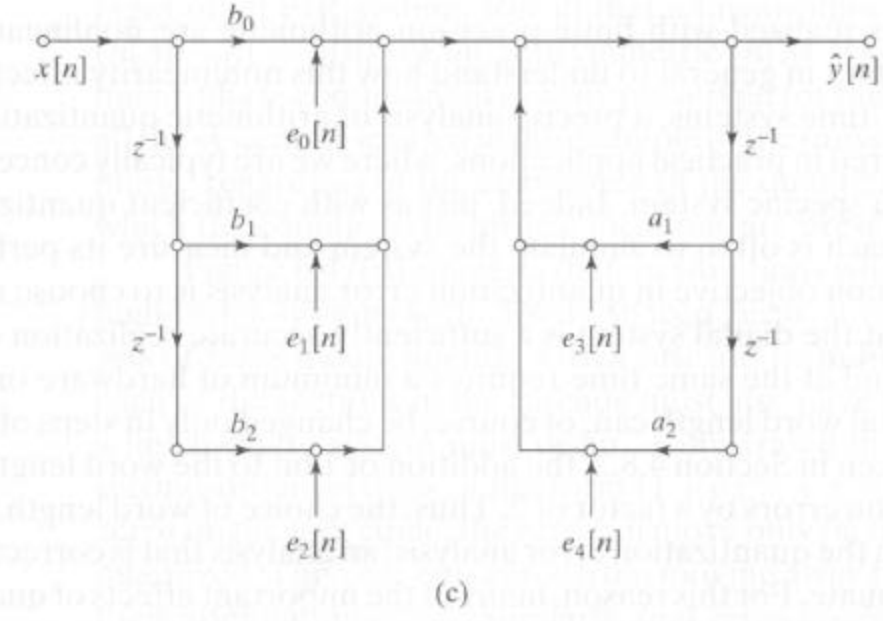
- The figures below show the additive noise model for the fixed point implementation of a 2nd-order IIR filter using the Direct Form I structure.



(a) Direct Form I (floating point) implementation of a 2nd order IIR Filter



(b) Fixed point implementation of a 2nd order IIR filter using the Direct Form I structure



(c) Additive noise model for the fixed-point 2nd order IIR filter in (b).

The terms $e_0[n]$, $e_1[n]$, $e_2[n]$, $e_3[n]$, and $e_4[n]$ in Diagram (c) represent the round-off noises in the five fixed-point multipliers. Each of these noise terms has zero mean and a variance of

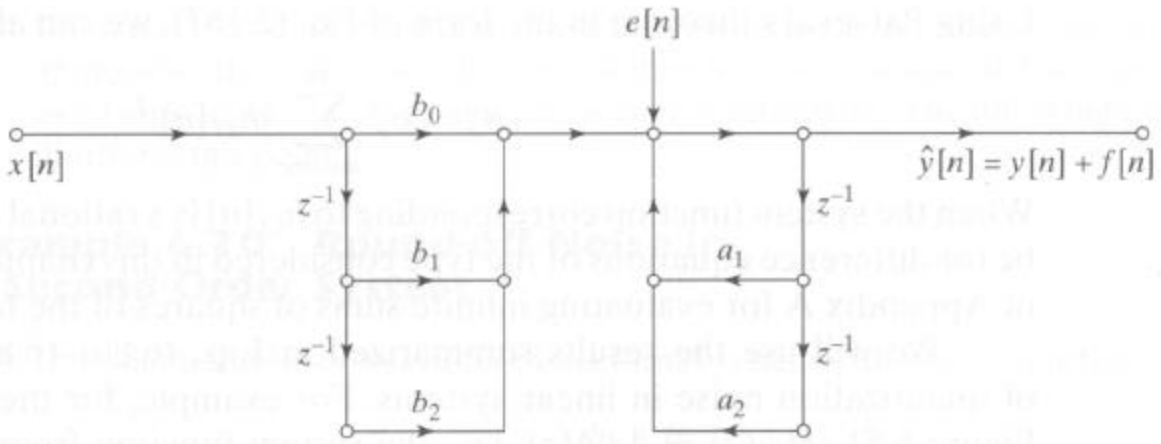
$$s^2 = \frac{\Delta^2}{12} = \frac{X_m^2 2^{-2B}}{12}$$

Furthermore, we assume that each $e_i[n]$ is white (i.e. $E[e_i[n]e_i[n+m]] = 0$) and statistically independent of one another.

- Since all the nodes in Diagram (c) represent adders, and since the output of the first stage is the input to the second stage, we can lump all the noise sources together into a single noise source

$$e[n] = e_0[n] + e_1[n] + e_2[n] + e_3[n] + e_4[n]$$

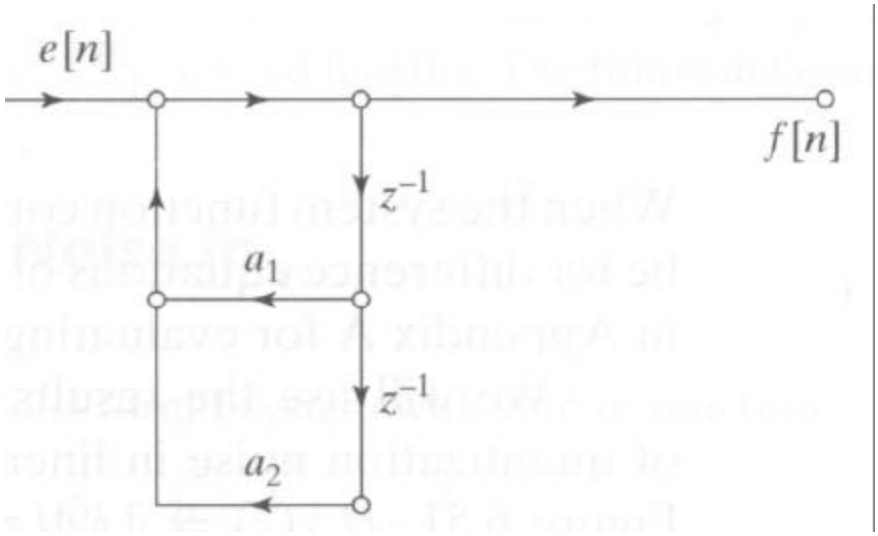
and place it at the input to the second stage; see diagram below.



Like the individual $e_i[n]$'s, the combined noise term $e[n]$ has zero mean. Its variance, however, is $s_e^2 = 5s^2$.

Since each $e_i[n]$ is white, so $e[n]$ is also white. Consequently, the power spectral density of $e[n]$ is $\Phi_{ee}(e^{jw}) = 5s^2$.

The output $\hat{y}[n]$ of the linearized model has two components, the desired (i.e. real value) output $y[n]$ and the noise term $f[n]$. The output noise term is obtained by disabling the input line and feeding the combined noise term to the feedback filter; see diagram below.



Since the frequency response of the feedback filter is

$$H_{ef}\left(e^{jw}\right)=\frac{1}{1-a_1e^{-jw}-a_2e^{-j2w}},$$

the PSD of the output noise is

$$\Phi_{ff}\left(e^{jw}\right)=\Phi_{ee}\left(e^{jw}\right)\left|H_{ef}\left(e^{jw}\right)\right|^2=5s^{2}\frac{1}{\left|1-a_1e^{-jw}-a_2e^{-j2w}\right|^2}$$

- In general, for an IIR filter with N feedback taps and $M+1$ feedforward taps, the variance of the combined noise term $e[n]$ in a Direct Form I fixed-point implementation is

$$\sigma_e^2 = (N+M+1)\sigma^2 = \frac{(N+M+1)X_m^2 2^{-2B}}{12},$$

and its power spectral density is

$$\Phi_{ee}(e^{jw}) = \sigma_e^2 = \frac{(N+M+1)X_m^2 2^{-2B}}{12}.$$

Since this combined noise term is injected into the input of the feedback filter, the PSD of the output quantization noise, $f[n]$, is

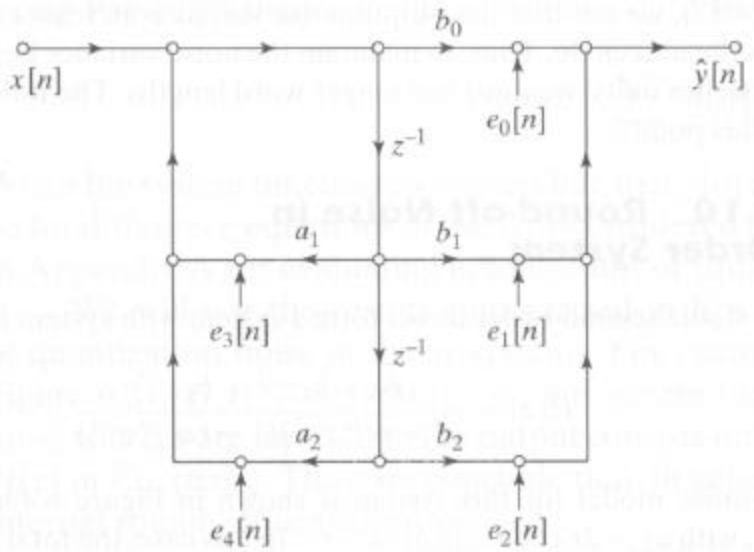
$$\begin{aligned}\Phi_{ff}(e^{jw}) &= \Phi_{ee}(e^{jw})\left|H_{ef}(e^{jw})\right|^2 \\ &= (N+M-1)\sigma^2 \frac{1}{\left|1-\sum_{k=1}^{N} a_k e^{-jwk}\right|^2} \\ &= (N+M+1)\left(\frac{X_m^2 2^{-2B}}{12}\right)\frac{1}{\left|1-\sum_{k=1}^{N} a_k e^{-jwk}\right|^2},\end{aligned}$$

and the output noise power is

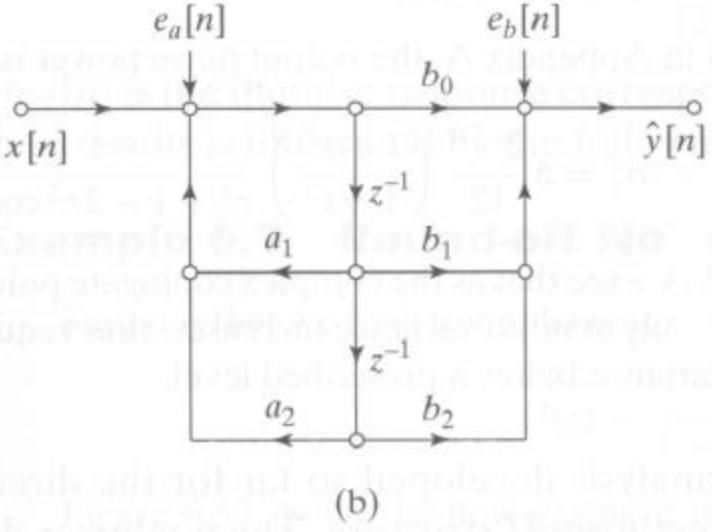
$$\sigma_f^2 = \frac{1}{2\pi}\int_{w=-\pi}^{\pi}\Phi_{ff}(e^{jw})dw = (N+M+1)\left(\frac{X_m^2 2^{-2B}}{12}\right)\frac{1}{2\pi}\int_{w=-\pi}^{\pi}\frac{dw}{\left|1-\sum_{k=1}^{N} a_k e^{-jwk}\right|^2}$$

- While the Cascade Form is preferred over the Direct Forms in terms of poles/zeros relocation caused by quantization, there are NO similar rules for output quantization error. A lot actually depends on the coefficients of the feedforward and feedback filters.

The diagrams below show the noise model for a fixed-point implementation of a 2nd order IIR filter using the Direct Form II structure. It is clear that round-off error affects Direct Forms I and II differently.



(a) Individual quantization noise sources in 2nd order Direct Form II IIR filter



(b) Combined quantization noise sources in 2nd order Direct Form II IIR filter

7.5 Quantization Noise in FFT

- The DFT coefficients of a N-sample long signal $x[n]$ are:

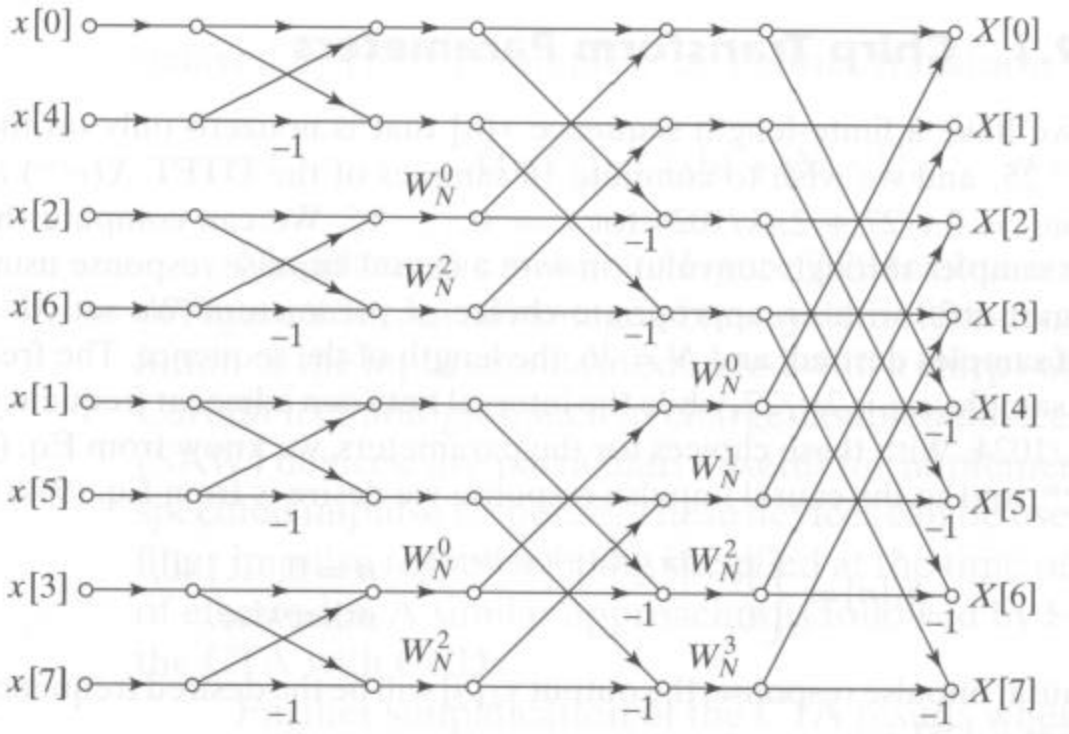
$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}; \qquad k = 0,1,..., N-1,$$

where

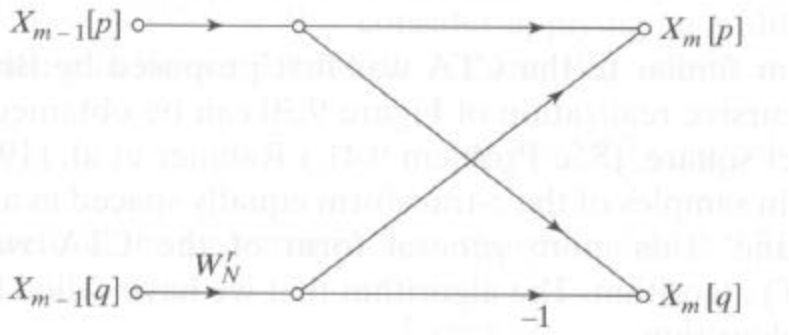
$$W_N = e^{-j2p/N}.$$

The DFT coefficients are actually uniformly spaced samples of $H\left(e^{jw}\right)$, the spectrum of $x[n]$, between $0 \le w < 2p$.

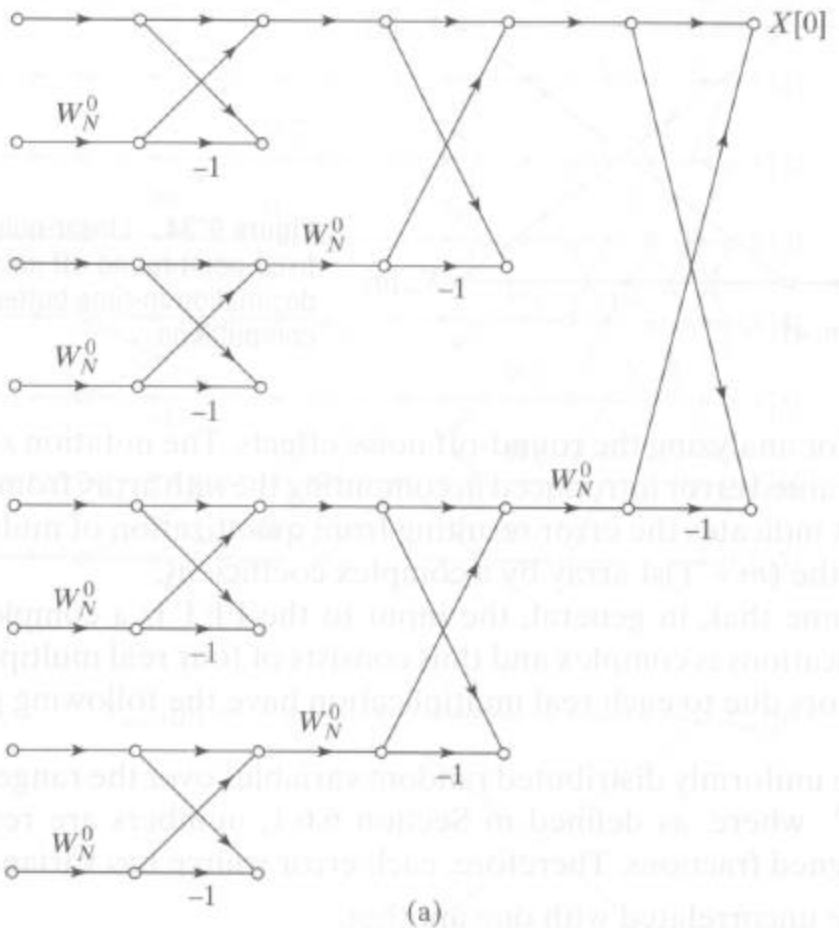
- The DFT coefficients can be computed efficiently using a FFT algorithm. The signal flow graph for the the decimation in time FFT algorithm, with $N = 8$, is shown below.



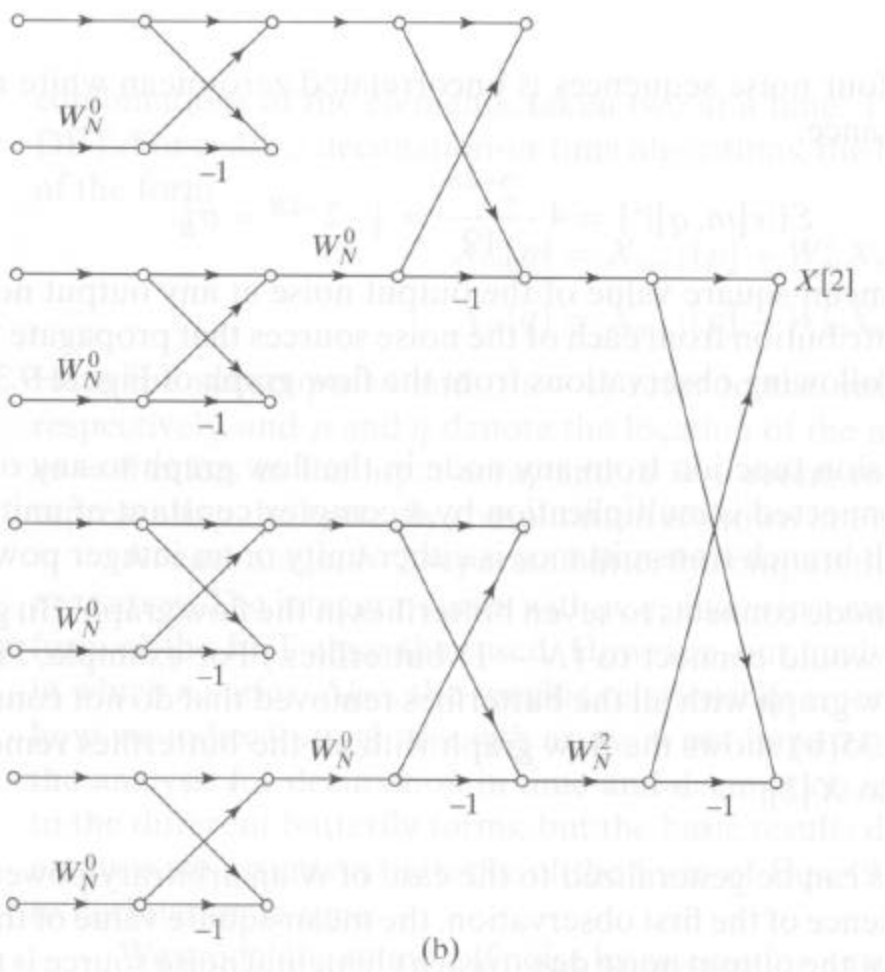
- For $N=8$, the computation of each DFT coefficient involves 7 butterfly computational structures of the form shown below,



where $X_m[p], X_m[q]$ represent the p-th and q-th intermediate output coefficients of the m-th stage. For example in the case of $X[0]$ we have

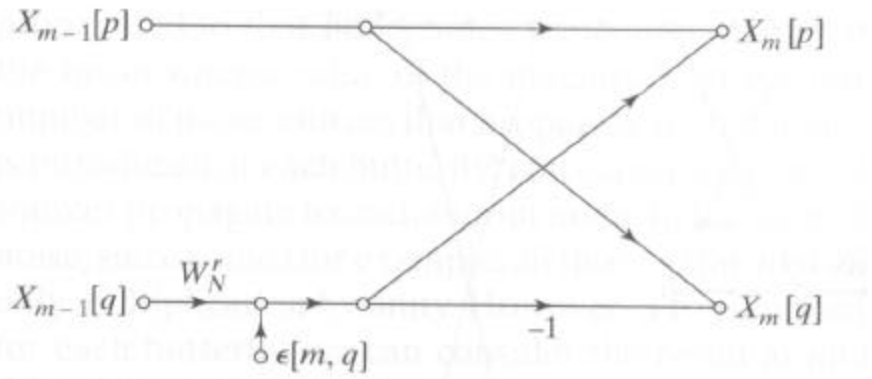


and for $X[2]$ we have



- In general, the computation of a DFT coefficient always involves $N-1$ butterfly structures.

- With fixed-point implementation, the complex multiplication of $X_{m-1}[q]$ by W_N^r in each butterfly introduces a complex quantization noise term $e[m,q]$; see diagram below.



- The expected value of $|e[m,q]|^2$ represents the average noise power due to quantization and it can be determined as follows. Let

$$c_1 = a_1 + jb_1$$

and

$$c_2 = a_2 + jb_2$$

be two complex numbers with real components a_1, a_2 and imaginary components b_1, b_2. Their product, with infinite precision implementation, is

$$c = (a_1a_2 - b_1b_2) + j(a_1b_2 + a_2b_1)$$

With fixed-point implementation though, the product of c_1 and c_2 becomes

$$\hat{c} = \{Q_B(a_1a_2) - Q_B(b_1b_2)\} + j\{Q_B(a_1b_2) + Q_B(a_2b_1)\}$$

The quantization error is thus

$$\begin{aligned}\hat{c} - c &= \{[Q_B(a_1a_2) - a_1a_2] - [Q_B(b_1b_2) - b_1b_2]\} + \\ &\quad j\{[Q_B(a_1b_2) - a_1b_2] + [Q_B(a_2b_1) - a_2b_1]\}, \\ &= (e_1 - e_2) + j(e_3 + e_4)\end{aligned}$$

where

$$e_1 = Q_B\left(a_1 a_2\right) - a_1 a_2,$$
$$e_2 = Q_B\left(b_1 b_2\right) - b_1 b_2,$$
$$e_3 = Q_B\left(a_1 b_2\right) - a_1 b_2,$$
$$e_4 = Q_B\left(a_2 b_1\right) - a_2 b_1$$

are the individual real-value quantization errors. It is assumed that these individual error terms are statistically independent. Consequently, the mean magnitude square of the complex quantization erro is

$$\begin{aligned} E\left[\left|\hat{c} - c\right|^2\right] &= E\left[\left(e_1 - e_2\right)^2 + \left(e_3 + e_4\right)^2\right] \\ &= E\left[\left(e_1^2 + e_2^2 - 2e_1 e_2\right) + \left(e_3^2 + e_4^2 + 2e_3 e_4\right)\right] \\ &= E\left[e_1^2\right] + E\left[e_2^2\right] + E\left[e_3^2\right] + E\left[e_4^2\right] \\ &= 4s^{\,2}, \end{aligned}$$

where

$$s^{\,2} = \Delta^2/12$$

is the mean-square quantization error of a $B+1$-bit real multiplier with a step size of Δ. From this analysis, we can conclude that the mean magnitude square of $e[m,q]$ is

$$E\left[\left|e[m,q]\right|^2\right] = 4\frac{\Delta^2}{12} = \frac{\Delta^2}{3} = s_B^{\,2}$$

- It should be pointed out that when the term W_N^r in the butterfly is either $+1, -1, j,$ or $-j$, then there is actually no quantization error.

For simplicity in our analysis though, we assume that each butterfly introduces a complex quantization noise term whose average power is s_B^2.

- From the signal flow graphs associated with the computation of $X[0]$ and $X[2]$ in the $N=8$ case, we see that a quantization noise term introduced in an intermediate butterfly will be multiplied by a sequence of complex exponential terms of the form W_N^r. This operation will not change the average power of that noise source.

  Consequently, an intermediate noise source can be replaced by an equivalent noise source with the same power at the output node.

  So for the $N=8$ example, there will be altogether 7 equivalent noise sources, all with a power of s_B^2, being added to the output node.

  It is reasonable to assume that all the 7 equivalent noise sources are statistically independent. Consequently, they together form a single combined noise source with a power of $7s_B^2$.

- In general, quantization in a *N*-point FFT algorithm produces the equivalence of a noise term of power $(N-1)s_B^2$ in each of the DFT coefficient. In other word, we can express the quantized DFT coefficinet $\hat{X}[k]$ as

$$\hat{X}[k] = X[k] + F[k],$$

  where $F[k]$ is the effective quantization noise and

$$E\left[\left|F[k]\right|^2\right] = (N-1)s_B^2 \approx Ns_B^2$$

- Recall that

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}; \qquad k = 0,1,..., N-1$$

Suppose the signal range for our fixed-point implementation is between –1 and +1. So to avoid overflow,

$$|X[k]| < 1.$$

This can be achieved by scaling the time-domain signal $x[n]$ in such a way that

$$|x[n]| < \frac{1}{N}.$$

This stems from the fact that

$$|X[k]| = \left| \sum_{n=0}^{N-1} x[n] W_N^{kn} \right| \le \sum_{n=0}^{N-1} |x[n]| \qquad .$$

- Assuming that each $|x[n]|$ is a uniform random variable in $[0,1/N]$. Then

$$E\left[|x[n]|^2 \right] = N \int_0^{1/N} y^2 dy = \frac{1}{3N^2}.$$

Furthermore, if we assume that all the $x[n]$'s are statistically independent, then

$$E\left[\left|X[k]\right|^2\right] = E\left[\left|\sum_{n=0}^{N-1} x[n]W_N^{kn}\right|^2\right]$$

$$= E\left[\left(\sum_{n=0}^{N-1} x[n]W_N^{kn}\right)\left(\sum_{m=0}^{N-1} x^*[m]W_N^{-km}\right)\right]$$

$$= E\left[\sum_{n=0}^{N-1}\left|x[n]\right|^2\right] + E\left[\sum_{n=0}^{N-1}\sum_{\substack{m=0 \\ m\neq n}}^{N-1} x^*[m]x[n]W_N^{kn}W_N^{-km}\right]$$

$$= N\frac{1}{3N^2}$$

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

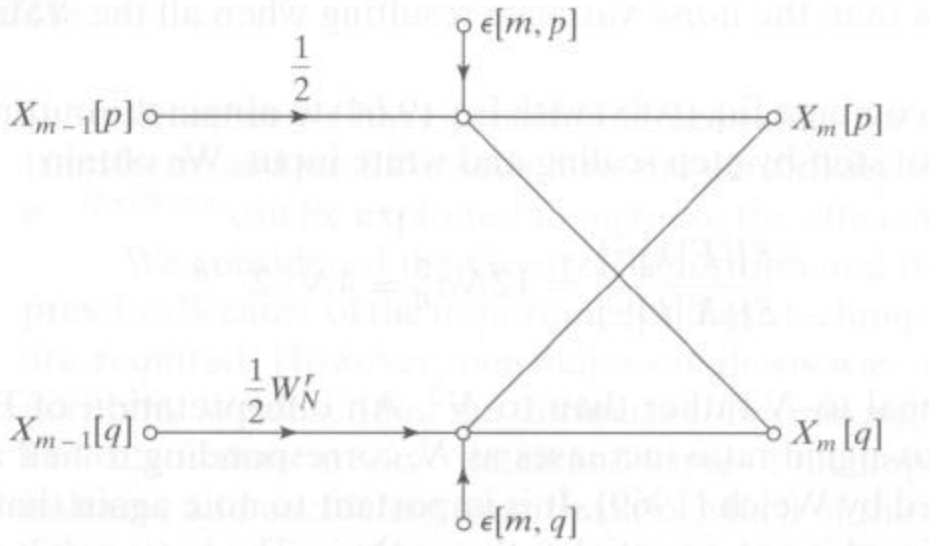
- The signal-to-noise (SNR) ratio in $\hat{X}[k]$ is thus

$$g = \frac{E\left[\left|X[k]\right|^2\right]}{E\left[\left|F[k]\right|^2\right]} = \frac{1/(3N)}{N\sigma_B^2} = \frac{1}{3N^2\sigma_B^2} = \frac{1}{N^2\Delta^2} = \frac{2^{2B}}{N^2},$$

  where $\Delta = 2^{-B}$ is the quantization step size of a (B+1)-bit quantizer when the signal range is between –1 and +1.

  The result indicates that if N is doubled (equivalent to adding 1 more FFT stage), then we must use 1 more bit in the quantization process in order to maintain the same SNR.

- It is possible to reduce the above 1 extra bit per extra stage requirement to ½ bit per stage if the butterfly structure below is adopted instead.



Here, the signal $x[n]$ (i.e. the input to the FFT) is scaled in such a way that

$$|x[n]| < 1.$$

(instead of $|x[n]| < 1/N$). To avoid overflow, the input to any intermediate butterfly is first scaled by ½ before being multiplied and added. The effective scaling factor for $x[n]$ is thus $1/2^v = 1/N$, where $v = \log_2 N$ is the total number of FFT stages.

A noise source in the $m-$th stage, i.e. $e[m, p]$ or $e[m, q]$, will be scaled in magnitude by the factor $2^{-(v-m-1)}$. This is in contrast to the original butterfly structure where all noise sources introduced by quantization has a scaling factor of 1 in magnitude.

This exponential weighting function on noise sources further back in the overall computational structure leads to the improvement in SNR.