# Coefficient quantization

- During the approximation step, the coefficients of a digital filter are calculated with the high accuracy inherent to the computer employed in the design.
- When these coefficients are quantized for practical implementations, commonly using rounding, the time and frequency responses of the realized digital filter deviate from the ideal response.
- In fact, the quantized filter may even fail to meet the prescribed specifications.
- The sensitivity of the filter response to errors in the coefficients is highly dependent on the type of structure.
- This fact is one of the motivations for considering alternative realizations having low sensitivity, such as those presented in Chapter 13.

# Coefficient quantization

- Among the several sensitivity criteria that evaluate the effect of the fixed-point coefficient quantization on the digital filter transfer function, the most widely used are

$$ {}_{I}S_{m_i}^{H(z)}(z) = \frac{\partial H(z)}{\partial m_i} \tag{125} $$

$$ {}_{II}S_{m_i}^{H(z)}(z) = \frac{1}{H(z)} \frac{\partial H(z)}{\partial m_i} \tag{126} $$

- For the floating-point representation, the sensitivity criterion must take into account the relative variation of $H(z)$ due to a relative variation in a multiplier coefficient. We then must use

$$ {}_{III}S_{m_i}^{H(z)}(z) = \frac{m_i}{H(z)} \frac{\partial H(z)}{\partial m_i} \tag{127} $$

# Coefficient quantization

- With such a formulation, it is possible to use the value of the multiplier coefficient to determine ${}_{III}S_{m}^{H(z)}(z)$.

- A simple example illustrating the importance of this fact is given by the quantization of the system

$$y(n) = (1+m)x(n); \quad \text{for } |m| \ll 1 \tag{128}$$

- Using equation (125), ${}_{I}S_{m_i}^{H(z)}(z) = 1$, regardless of the value of $m$, while using equation (127), ${}_{III}S_{m_i}^{H(z)}(z) = m/(m+1)$, indicating that a smaller value for the magnitude of $m$ leads to a smaller sensitivity of $H(z)$ with respect to $m$.

- This is true for the floating-point representation, as long as the number of bits in the exponent is enough to represent the exponent of $m$.

# Example 11.6

- Determine the possible pole positions for a direct-form second-order section with denominator

$$D(z) = z^2 + a_1 z + a_2 \tag{129}$$

when the filter coefficients $a_1$ and $a_2$ are represented with 6 bits, including the sign bit, using standard binary representation.

- Repeat your analysis using a state-space structure characterized by $a_{11} = a_{22} = a$ and $a_{21} = -a_{12} = \zeta$, when $a$ and $\zeta$ are represented with 6 bits. Such a structure, when the values of at least two different coefficients are dependent on a single parameter, is referred to as a coupled-form state-space structure.

# Example 11.6 - Solution

- Figure 22a depicts the possible pole placements in the first quadrant within the $z$-domain unit circle for the direct-form second-order section.
- In the remaining quadrants, pole placements are symmetric mirrored copies of the ones seen in this figure.
- As can be observed, the pole grid becomes very sparse around to the real axis, particularly close to $z = 0$ or $z = 1$.
- This explains the implementation inaccuracy achieved by this section type in applications with high sampling rate, since these cases often require a filter with poles close to the real axis.
- The same phenomenon occurs if the poles are required to be close to $z = 1$ or $z = -1$.

# Example 11.6 - Solution

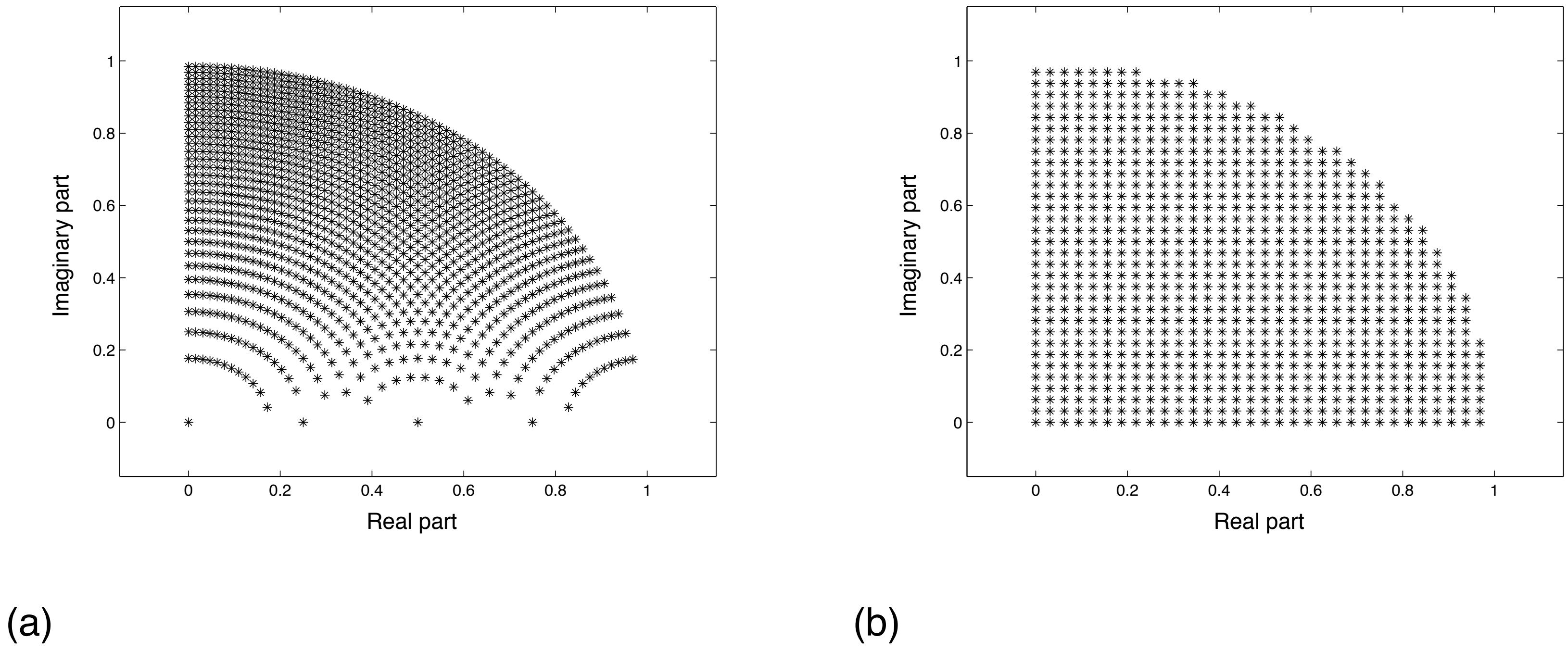


Figure 22: Pole grid for second-order sections with 6-bit coefficients: (a) direct-form; (b) state-space coupled form.

# Example 11.6 - Solution

- For the coupled form, the denominator polynomial becomes

$$D(z) = z^2 - 2az + a^2 + \zeta^2 \tag{130}$$

where $a$ represents the real part of the complex conjugate poles and $\zeta$ their imaginary part.

- The pole-placement analysis for the coefficient quantization in this structure is shown in Figure 22b, where we observe a uniform grid distribution around the entire quadrant.

- This result implies that there is no preferred region for this structure to place the poles, which is an attractive feature obtained at the cost of four multiplier coefficients to position a single pair of complex conjugate poles.

# Deterministic sensitivity criterion

- In practice, one is often interested in the variation of the magnitude of the transfer function, $|H(e^{j\omega})|$, with coefficient quantization. Taking into account the contributions of all multipliers, a useful figure of merit related to this variation would be

$$S(e^{j\omega}) = \sum_{i=1}^{K} \left| S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega}) \right| \tag{131}$$

  where $K$ is the total number of multipliers in the structure, and $S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})$ is computed according to one of the equations (125)–(127), depending on the case.

- However, in general, the sensitivities of $H(e^{j\omega})$ to coefficient quantization are much easier to derive than the ones of $|H(e^{j\omega})|$, and thus it would be convenient if the first one could be used as an estimate of the second.

# Deterministic sensitivity criterion

- In order to investigate this possibility, we write the frequency response in terms of its magnitude and phase as

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\Theta(\omega)} \tag{132}$$

- Then the sensitivity measures, defined in equations (125)–(127), can be written as

$$\left|{}_{I}S_{m_i}^{H(e^{j\omega})}(e^{j\omega})\right| = \sqrt{\left({}_{I}S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})\right)^2 + |H(e^{j\omega})|^2\left(\frac{\partial\Theta(\omega)}{\partial m_i}\right)^2} \tag{133}$$

$$\left|{}_{II}S_{m_i}^{H(e^{j\omega})}(e^{j\omega})\right| = \sqrt{\left({}_{II}S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})\right)^2 + \left(\frac{\partial\Theta(\omega)}{\partial m_i}\right)^2} \tag{134}$$

$$\left|{}_{III}S_{m_i}^{H(e^{j\omega})}(e^{j\omega})\right| = \sqrt{\left({}_{III}S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})\right)^2 + |m_i|^2\left(\frac{\partial\Theta(\omega)}{\partial m_i}\right)^2} \tag{135}$$

- From equations (133)–(135), one can see that $|S_{m_i}^{H(e^{j\omega})}(e^{j\omega})| \geq |S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})|$.

# Deterministic sensitivity criterion

- Thus, $|S_{m_i}^{H(e^{j\omega})}(e^{j\omega})|$ can be used as a conservative estimate of $|S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})|$, in the sense that it guarantees that the transfer function variation will be below a specified tolerance.

- Moreover, it is known that low sensitivity is more critical when implementing filters with poles close to the unit circle, and, in such cases, for $\omega$ close to a pole frequency $\omega_0$, we can show that $|S_{m_i}^{H(e^{j\omega})}(e^{j\omega})| \approx |S_{m_i}^{|H(e^{j\omega})|}(e^{j\omega})|$.

- Therefore, we can rewrite equation (131), yielding the following practical sensitivity figure of merit:

$$S(e^{j\omega}) = \sum_{i=1}^{K} \left| S_{m_i}^{H(e^{j\omega})}(e^{j\omega}) \right| \tag{136}$$

where, depending on the case, $S_{m_i}^{H(e^{j\omega})}(e^{j\omega})$ is given by one of the equations (125)–(127).

# Example 11.7

- Design a lowpass elliptic filter with the following specifications:

$$\left.\begin{aligned} A_{\mathrm{p}} &= 1.0 \text{ dB} \\ A_{\mathrm{r}} &= 40 \text{ dB} \\ \omega_{\mathrm{p}} &= 0.3\pi \text{ rad/sample} \\ \omega_{\mathrm{r}} &= 0.4\pi \text{ rad/sample} \end{aligned}\right\} \tag{137}$$

- Perform the fixed-point sensitivity analysis for the direct-order structure, determining the variation on the ideal magnitude response for an 11-bit quantization of the fractional part, including the sign bit.

# Example 11.7 - Solution

- The coefficients of the lowpass elliptic filter are given in Table 1.

Table 1: Filter coefficients for the specifications (137).

| Numerator coefficients | Denominator coefficients |
|---|---|
| $b_0 = \phantom{-}0.028\,207\,76$ | $a_0 = \phantom{-}1.000\,000\,00$ |
| $b_1 = -0.001\,494\,75$ | $a_1 = -3.028\,484\,73$ |
| $b_2 = \phantom{-}0.031\,747\,58$ | $a_2 = \phantom{-}4.567\,772\,20$ |
| $b_3 = \phantom{-}0.031\,747\,58$ | $a_3 = -3.900\,153\,49$ |
| $b_4 = -0.001\,494\,75$ | $a_4 = \phantom{-}1.896\,641\,38$ |
| $b_5 = \phantom{-}0.028\,207\,76$ | $a_5 = -0.418\,854\,19$ |

# Example 11.7 - Solution

- For the general direct-form structure described by

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{138}$$

it is easy to find that the sensitivities as defined in equation (125) with respect to the numerator and denominator coefficients are given by

$$_I S_{b_i}^{H(z)}(z) = \frac{z^{-i}}{A(z)}; \quad _I S_{a_i}^{H(z)}(z) = -\frac{z^{-i} H(z)}{A(z)} \tag{139}$$

respectively.

- The magnitude of these functions for the designed fifth-order elliptic filter are seen in Figure 23.

# Example 11.7 - Solution

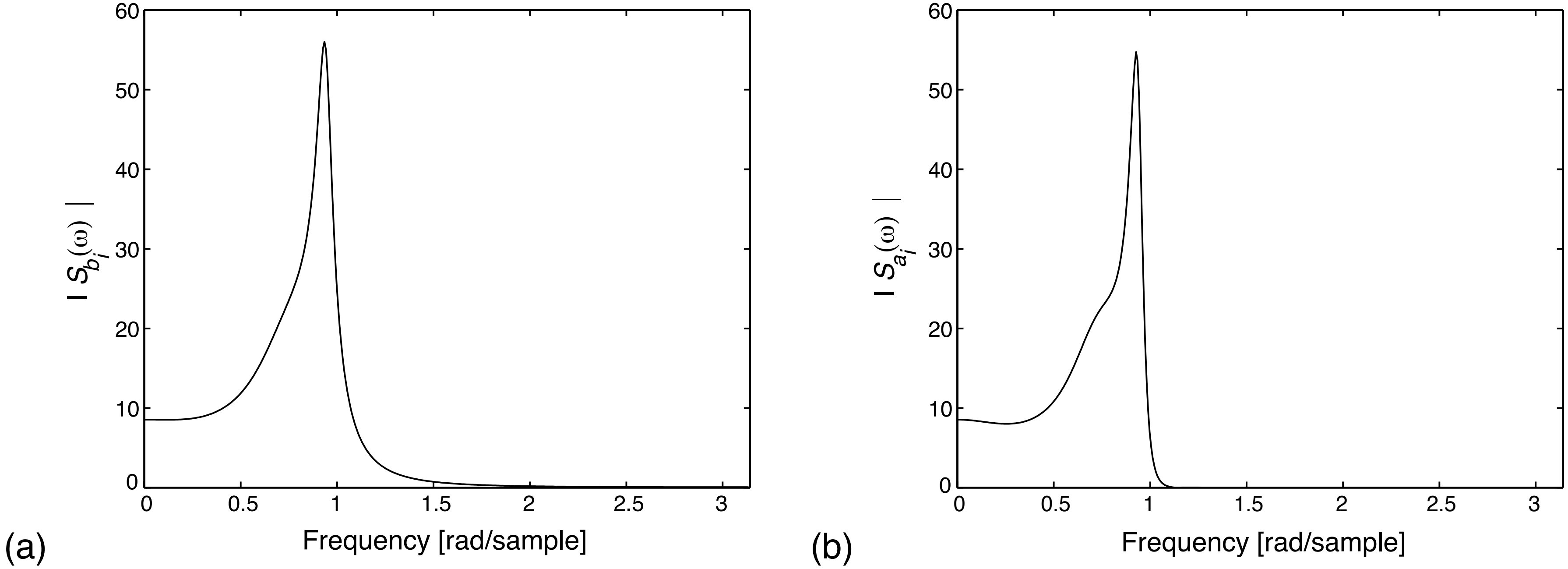


Figure 23: Magnitudes of the sensitivity functions of $H(z)$ with respect to: (a) numerator coefficients $b_i$; (b) denominator coefficients $a_i$.

# Example 11.7 - Solution

- The figure of merit $S(e^{j\omega})$, as given in equation (136), for the general direct-form realization can be written as

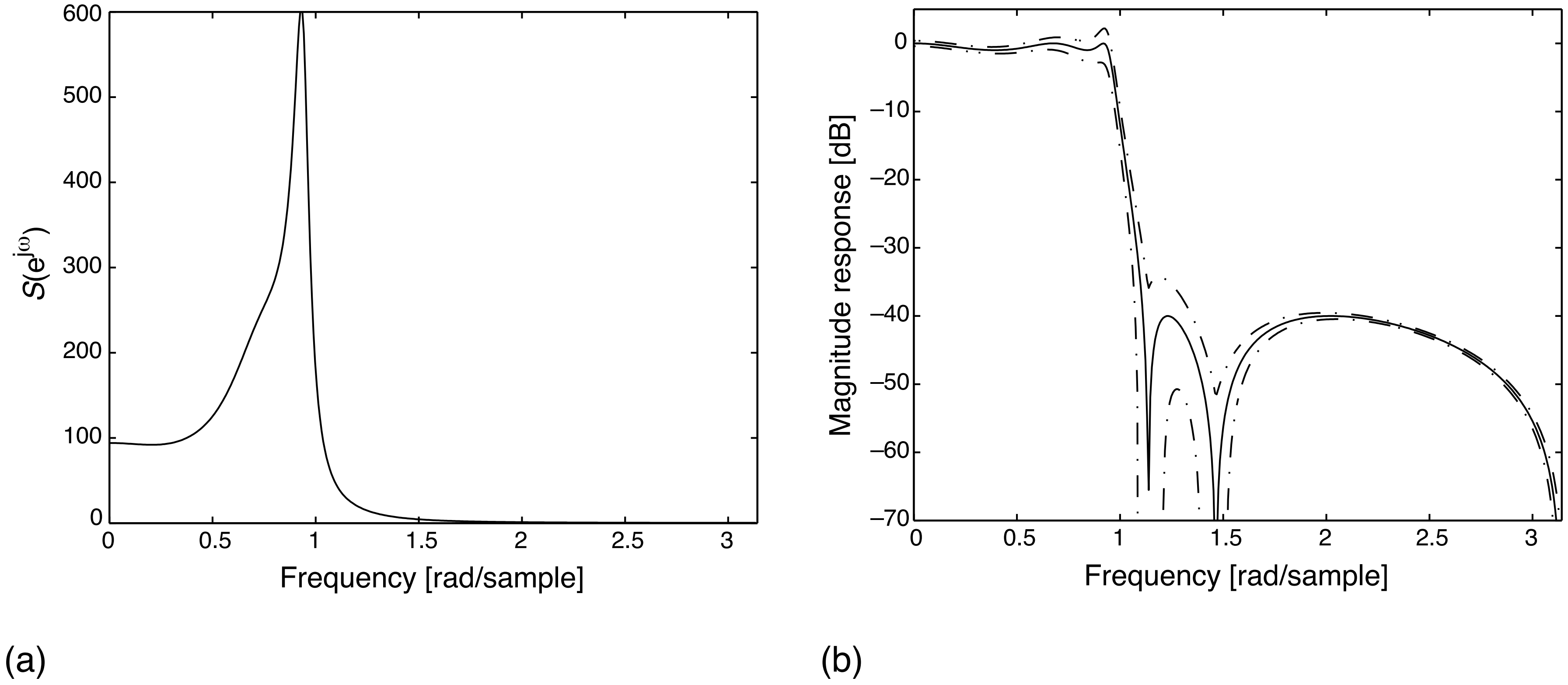
$$S(e^{j\omega}) = \frac{(N+1) + N|H(z)|}{|A(z)|} \tag{140}$$

- For this example, the function is depicted in Figure 24a.
- We can then estimate the variation of the ideal magnitude response using the approximation

$$\Delta|H(e^{j\omega})| \approx \Delta m_i S(e^{j\omega}) \tag{141}$$

- For a fixed-point 11-bit rounding quantization, including the sign bit, max$\{\Delta m_i\} = 2^{-11}$.
- In this case, Figure 24b depicts the ideal magnitude response of a fifth-order elliptic filter, satisfying the specifications in equation (137), along with the corresponding worst-case margins due to the coefficient quantization.

# Example 11.7 - Solution



(a)

(b)

Figure 24: Finite-precision analysis: (a) sensitivity measurement $S(e^{j\omega})$; (b) worst-case variation of $|\mathrm{H}(e^{j\omega})|$ with an 11-bit fixed-point quantization.

# Deterministic sensitivity criterion

- It is worth noting that the sensitivity measurement given by equation (136) is also useful as a figure of merit if one uses the so-called pseudo-floating-point representation, that consists of implementing multiplication between a signal and a coefficient with small magnitude in the following form, as depicted in Figure 25

$$[x \times m_i]_Q = [(x \times m_i \times 2^L) \times 2^{-L}]_Q \tag{142}$$

where $L$ is the exponent of $m_i$ when represented in floating point.

- Note that in the pseudo-floating-point scheme, all operations are actually performed using fixed-point arithmetic.

## Deterministic sensitivity criterion

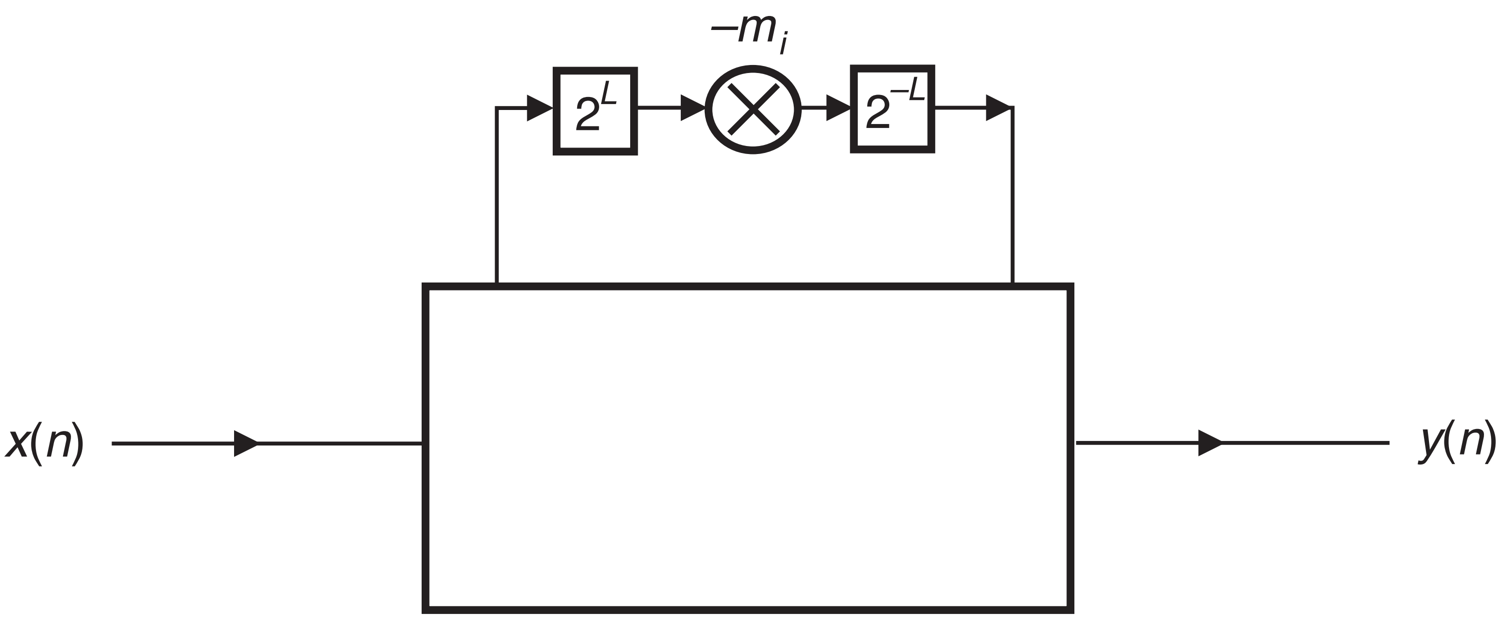


Figure 25: Implementation of a multiplication in a pseudo-floating-point representation.

# Statistical forecast of the wordlength

- In the previous subsection we computed the worst case variation in the frequency response of a digital filter with the quantization of coefficients. By worst case we meant the supposition that quantization made all the coefficients to vary the maximum possible amount, and in the worst direction.
- However, since it is unlikely that all coefficients will undergo a worst-case quantization error, and their quantization effects will accumulate in the worst possible way with respect to the resulting frequency response.
- Thus, it is useful to perform a more realistic, statistical analysis of the deviation in the frequency response.

# Statistical forecast of the wordlength

- In this subsection we perform a statistical forecast of the wordlength necessary for a filter to satisfy a given specification.
- Suppose we have designed a digital filter with frequency response $H(e^{j\omega})$, and that the ideal magnitude response is $H_d(e^{j\omega})$, with a tolerance given by $\rho(\omega)$.
- When the filter coefficients are quantized, we can express the resulting magnitude response as

$$\left|H_Q(e^{j\omega})\right| = \left|H(e^{j\omega})\right| + \Delta\left|H(e^{j\omega})\right| \tag{143}$$

# Statistical forecast of the wordlength

- Obviously, for a meaningful design, $|H_Q(e^{j\omega})|$ must not deviate from $H_d(e^{j\omega})$ by more than a frequency-dependent tolerance $\rho(\omega)$, that is

$$\left|\left(\left|H_Q(e^{j\omega})\right| - H_d(e^{j\omega})\right)\right| = \left|\left|H(e^{j\omega})\right| + \Delta\left|H(e^{j\omega})\right| - H_d(e^{j\omega})\right| \leq \rho(\omega) \tag{144}$$

  or, more strictly,

$$\left|\Delta\left|H(e^{j\omega})\right|\right| \leq \rho(\omega) - \left|\left|H(e^{j\omega})\right| - H_d(e^{j\omega})\right| \tag{145}$$

- The variation in the magnitude response of the digital filter due to the variations in the multiplier coefficients $m_i$ can be approximated by

$$\Delta\left|H(e^{j\omega})\right| \approx \sum_{i=1}^{K} \frac{\partial\left|H(e^{j\omega})\right|}{\partial m_i}\Delta m_i \tag{146}$$

# Statistical forecast of the wordlength

- If we consider that:
  - the multiplier coefficients are rounded;
  - the quantization errors are statistically independent;
  - all $\Delta m_i$ are uniformly distributed;

then the variance of the error in each coefficient, based on equation (54), is given by

$$\sigma^2_{\Delta m_i} = \sigma^2_{\Delta m} = \frac{2^{-2b}}{12}, \quad \text{for} \quad i = 1, 2, \ldots, K \tag{147}$$

where $b$ is the number of bits not including the sign bit.

# Statistical forecast of the wordlength

- With the assumptions above, the mean of $\Delta|H(e^{j\omega})|$ is zero and its variance is given by

$$\sigma^2_{\Delta|H(e^{j\omega})|} \approx \sigma^2_{\Delta m} \sum_{i=1}^{K} \left( \frac{\partial\, |H(e^{j\omega})|}{\partial m_i} \right)^2 = \sigma^2_{\Delta m} S^2(e^{j\omega}) \qquad (148)$$

where $S^2(e^{j\omega})$ is given by equations (125) and (136).

- If we further assume that $\Delta\,|H(e^{j\omega})|$ is Gaussian, we can estimate the probability of $\Delta\,|H(e^{j\omega})|$ being less than or equal to $x\sigma_{\Delta|H(e^{j\omega})|}$ by

$$\Pr\left\{ \left|\Delta H(e^{j\omega})\right| \leq x\sigma_{\Delta|H(e^{j\omega})|} \right\} = \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{\sqrt{2}}} e^{-x'^2} dx' \qquad (149)$$

# Statistical forecast of the wordlength

- To guarantee that equation (145) holds with a probability less than or equal to the one given in equation (149), it suffices that

$$x\sigma_{\Delta m}S(e^{j\omega}) \leq \rho(\omega) - \left|\left|H(e^{j\omega})\right| - H_d(e^{j\omega})\right| \tag{150}$$

- Now, assume that the wordlength, including the sign bit, is given by

$$B = I + F + 1 \tag{151}$$

  where $I$ and $F$ are the numbers of bits in the integer and fractional parts, respectively.

- The value of $I$ depends on the required order of magnitude of the coefficient, and $F$ can be estimated from equation (150) to guarantee that equation (145) holds with probability as given in equation (149).

## Statistical forecast of the wordlength

- To satisfy the inequality in (150), the value of $2^{-b}$, from equation (147), should be given by

$$2^{-b} = \sqrt{12} \min_{\omega \in C} \left\{ \left| \frac{\rho(\omega) - ||H(e^{j\omega})| - H_d(e^{j\omega})|}{xS(e^{j\omega})} \right| \right\} \tag{152}$$

where $C$ is the set of frequencies not belonging to the filter transition bands.

- Then, an estimate for $F$ is

$$F \approx b = -\log_2 \left( \sqrt{12} \min_{\omega \in C} \left\{ \left| \frac{\rho(\omega) - ||H(e^{j\omega})| - H_d(e^{j\omega})|}{xS(e^{j\omega})} \right| \right\} \right) \tag{153}$$

# Statistical forecast of the wordlength

- This method for estimating the wordlength is also useful in iterative procedures to design filters with minimum wordlength.

- An alternative procedure that is widely used in practice to evaluate the design of digital filters with finite-coefficient wordlength, is to design the filters with tighter specifications than required, quantize the coefficients, and check if the prescribed specifications are still met.

- Obviously, in this case, the success of the design is highly dependent on the designer’s experience.

# Example 11.8

- Determine the total number of bits required for the filter designed in Example 11.3 to satisfy the following specifications, after coefficient quantization:

$$\left.\begin{aligned} A_\mathrm{p} &= 1.2 \text{ dB} \\ A_\mathrm{r} &= 39 \text{ dB} \\ \omega_\mathrm{p} &= 0.3\pi \text{ rad/sample} \\ \omega_\mathrm{r} &= 0.4\pi \text{ rad/sample} \end{aligned}\right\} \tag{154}$$

## Example 11.8 - Solution

- Using the specifications in equation (154), we determine

$$\delta_p = 1–10^{-A_p/20} = 0.1482 \tag{155}$$

$$\delta_r = 10^{-A_r/20} = 0.0112 \tag{156}$$

and define

$$\rho(\omega) = \begin{cases} \delta_p, & \text{for } 0 \leq \omega \leq 0.3\pi \\ \delta_r, & \text{for } 0.4\pi \leq \omega \leq \pi \end{cases} \tag{157}$$

$$H_d(e^{j\omega}) = \begin{cases} 1, & \text{for } 0 \leq \omega \leq 0.3\pi \\ 0, & \text{for } 0.4\pi \leq \omega \leq \pi \end{cases} \tag{158}$$

# Example 11.8 - Solution

- A reasonable certainty margin is about 90%, yielding, from equation (149),

$$\frac{x}{\sqrt{2}} = \texttt{erfinv}(0.9) = 1.1631 \Rightarrow x = 1.6449 \tag{159}$$

- We use the filter designed in Example 11.3 as $H(e^{j\omega})$, with the corresponding sensitivity function $S(e^{j\omega})$, as given in equation (140) and depicted in Figure 24a.
- Based on these values, we can compute the number of bits $F$ for the fractional part, using equation (153), resulting in $F \approx 12.0993$, which we round to $F = 12$ bits.
- From Table 1, we observe that $I = 3$ bits are necessary to represent the integer part of the filter coefficients which fall in the range $-4$ to $+4$. Therefore, the total number of bits required, including the sign bit, is

$$B = I + F + 1 = 16 \tag{160}$$

# Example 11.8 - Solution

- Table 2 shows the filter coefficients after quantization.

Table 2: Quantized filter coefficients for specifications (154).

| Numerator<br>coefficients | Denominator<br>coefficients |
|---|---|
| $b_0 = \phantom{-}0.028\,320\,31$ | $a_0 = \phantom{-}1.000\,000\,00$ |
| $b_1 = -0.001\,464\,84$ | $a_1 = -3.028\,564\,45$ |
| $b_2 = \phantom{-}0.031\,738\,28$ | $a_2 = \phantom{-}4.567\,871\,09$ |
| $b_3 = \phantom{-}0.031\,738\,28$ | $a_3 = -3.900\,146\,48$ |
| $b_4 = -0.001\,464\,84$ | $a_4 = \phantom{-}1.896\,728\,52$ |
| $b_5 = \phantom{-}0.028\,320\,31$ | $a_5 = -0.418\,945\,31$ |

# Example 11.8 - Solution

- Table 3 includes the resulting passband ripple and stopband attenuation for several values of $F$, from which we can clearly see that using the predicted $F = 12$, the specifications in equation (154) are satisfied, even after quantization of the filter coefficients.

Table 3: Filter characteristics as a function of the number of fractional bits $F$.

| $F$ | $A_p$ [dB] | $A_r$ [dB] |
|---|---|---|
| 15 | 1.0100 | 40.0012 |
| 14 | 1.0188 | 40.0106 |
| 13 | 1.0174 | 40.0107 |
| **12** | **1.1625** | **39.7525** |
| 11 | 1.1689 | 39.7581 |
| 10 | 1.2996 | 39.7650 |
| 9 | 1.2015 | 40.0280 |
| 8 | 2.3785 | 40.2212 |

# Limit cycles

- A serious practical problem that affects the implementation of recursive digital filters is the possible occurrence of parasitic oscillations.
- These oscillations can be classified, according to their origin, as either granular or overflow limit cycles, as presented below.

# Granular limit cycles

- Any stable digital filter, if implemented with idealized infinite-precision arithmetic, should have an asymptotically decreasing response when the input signal becomes zero after a given instant of time $n_0T$.
- However, if the filter is implemented with finite-precision arithmetic, the noise signals generated at the quantizers become highly correlated from sample to sample and from source to source.
- This correlation can cause autonomous oscillations, referred to as granular limit cycles, originating from quantization performed in the least significant signal bits, as indicated in the example that follows.

## Example 11.9

- Suppose the filter of Figure 26 has the following input signal:

$$x(n) = \begin{cases} 0.111, & \text{for } n = 1 \\ 0.000, & \text{for } n \neq 1 \end{cases} \qquad (161)$$

where the numbers are represented in two's complement. Determine the output signal in the case when the quantizer performs rounding, for $n = 1, 2, \ldots, 40$.

# Example 11.9

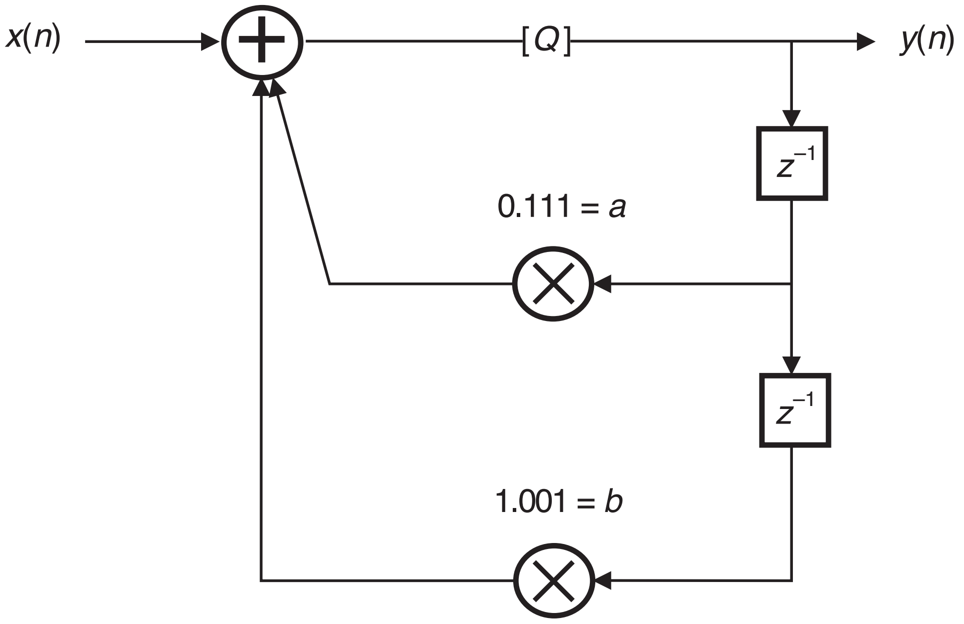


Figure 26: Second-order section with a quantizer.

# Example 11.9 - Solution

- The output in the time domain, assuming that the quantizer rounds the signal, is given in Table 4, where one can easily see that an oscillation is sustained at the output, even after the input becomes zero.
- In many practical applications, where the signal levels in a digital filter can be constant or very low, even for short periods of time, limit cycles are highly undesirable, and should be eliminated or at least have their amplitude bounds strictly limited.

# Example 11.9 - Solution

Table 4: Output signal of network shown in Figure 26.

| $n$ | $y(n)$ |
|---|---|
| 1 | 0.111 |
| 2 | $Q(0.110\,001 + 0.000\,000) = 0.110$ |
| 3 | $Q(0.101\,010 + 1.001\,111) = 1.111$ |
| 4 | $Q(1.111\,001 + 1.010\,110) = 1.010$ |
| 5 | $Q(1.010\,110 + 0.000\,111) = 1.100$ |
| 6 | $Q(1.100\,100 + 0.101\,010) = 0.010$ |
| 7 | $Q(0.001\,110 + 0.011\,100) = 0.101$ |
| 8 | $Q(0.100\,011 + 1.110\,010) = 0.011$ |
| 9 | $Q(0.010\,101 + 1.011\,101) = 1.110$ |
| 10 | $Q(1.110\,010 + 1.101\,011) = 1.100$ |
| 11 | $Q(1.100\,100 + 0.001\,110) = 1.110$ |
| 12 | $Q(1.110\,010 + 0.011\,100) = 0.010$ |
| 13 | $Q(0.001\,110 + 0.001\,110) = 0.100$ |
| 14 | $Q(0.011\,100 + 1.110\,010) = 0.010$ |
| 15 | $Q(0.001\,110 + 1.100\,100) = 1.110$ |
| 16 | $Q(1.110\,010 + 1.110\,010) = 1.100$ |
| 17 | $Q(1.100\,100 + 0.001\,110) = 1.110$ |
| 18 | $Q(1.110\,010 + 0.011\,100) = 0.010$ |
| 19 | $Q(0.001\,110 + 0.001\,110) = 0.100$ |
| 20 | $Q(0.011\,100 + 1.110\,010) = 0.010$ |
| 21 | $Q(0.001\,110 + 1.100\,100) = 1.110$ |
| 22 | $Q(1.110\,010 + 1.110\,010) = 1.100$ |
| $\vdots$ | $\vdots$ |

# Overflow limit cycles

- Overflow limit cycles can occur when the magnitudes of the internal signals exceed the available register range.
- In order to avoid the increase of the signal wordlength in recursive digital filters, overflow nonlinearities must be applied to the signal.
- Such nonlinearities influence the most significant bits of the signal, possibly causing severe distortion.
- An overflow can give rise to self-sustained, high-amplitude oscillations, widely known as overflow limit cycles.
- Overflow can occur in any structure in the presence of an input signal, and input-signal scaling is crucial to reduce the probability of overflow to an acceptable level.

# Example 11.10

- Consider the filter of Figure 27 with $a = 0.9606$ and $b = 0.9849$, where the overflow nonlinearity employed is the two's complement with 3-bit quantization (see Figure 27).

- Its analytic expression is given by

$$Q(x) = \frac{1}{4}[(\lceil 4x - 0.5 \rceil + 4) \mod 8] - 1 \tag{162}$$

where $\lceil x \rceil$ means the smallest integer larger than or equal to $x$.

- Determine the output signal of such a filter for zero input, given the initial conditions $y(-2) = 0.50$ and $y(-1) = -1.00$.

# Example 11.10

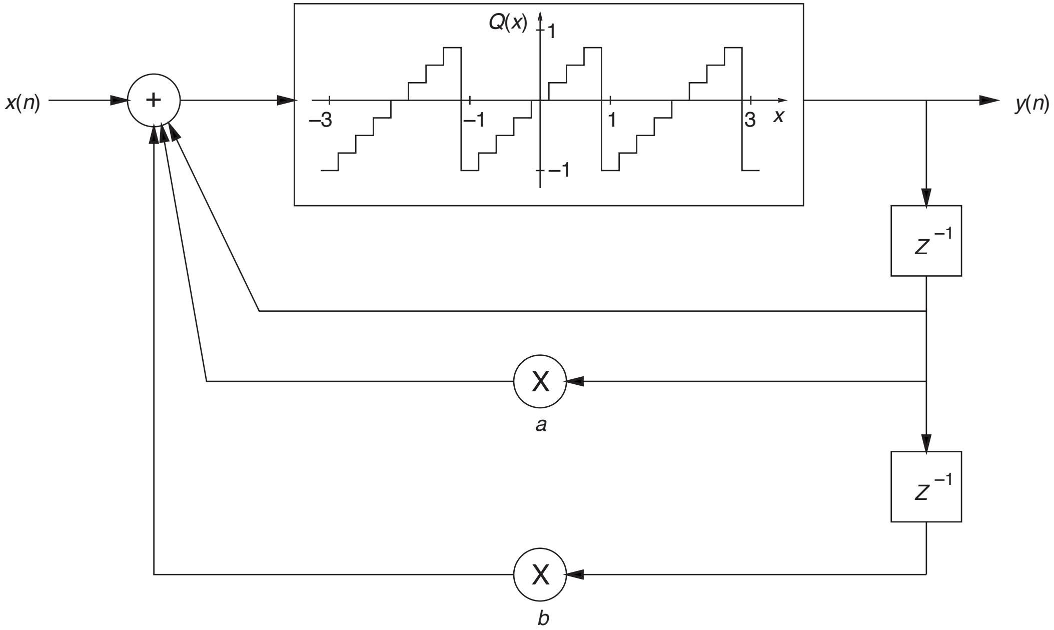


Figure 27: Second-order section with an overflow quantizer.

## Example 11.10 - Solution

- With $a = 0.9606$, $b = -0.9849$, $y(-2) = 0.50$ and $y(-1) = -1.00$, we have that

$$\begin{aligned} y(0) &= Q[1.9606(-1.00) - 0.9849(0.50)] &&= Q[-2.4530] = -0.50 \\ y(1) &= Q[1.9606(-0.50) - 0.9849(-1.00)] &&= Q[0.0046] \quad = 0.00 \\ y(2) &= Q[1.9606(0.00) - 0.9849(-0.50)] &&= Q[0.4924] \quad = 0.50 \\ y(3) &= Q[1.9606(0.50) - 0.9849(0.00)] &&= Q[0.9803] \quad = -1.00 \\ &\vdots \end{aligned} \tag{163}$$

- Since $y(2) = y(-2)$ and $y(3) = y(-1)$, we have that, although there is no excitation, the output signal in nonzero and periodic with a period of 4, thus indicating the existence of overflow limit cycles.

## Overflow limit cycles

- A digital filter structure is considered free from overflow limit cycles if the error introduced in the filter after an overflow decreases with time in such a way that the output of the nonlinear filter (including the quantizers) converges to the output of the ideal linear filter.

- In practice a quantizer incorporates nonlinearities corresponding to both granular quantization and overflow.

- Figure 28 illustrates a digital filter using a quantizer that implements rounding as the granular quantization and saturation arithmetic as the overflow nonlinearity.

- Note that although this overflow nonlinearity is different from the one depicted in Figure 27, both are classified as overflow.

# Overflow limit cycles

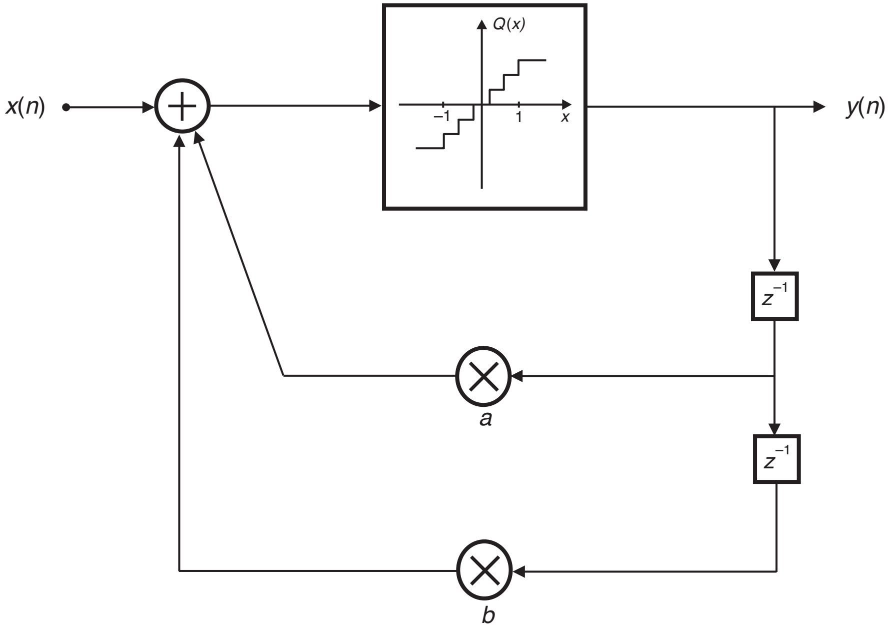


Figure 28: Second-order section with a rounding and saturating quantizer.

# Elimination of zero-input limit cycles

- A general IIR filter can be depicted as in Figure 29a, where the linear $N$-port network consists of interconnections of multipliers and adders.

- In a recursive filter implemented with fixed-point arithmetic, each internal loop contains a quantizer.

- Assuming that the quantizers are placed at the delay inputs (the state variables), as shown in Figure 29b, we can describe the digital filter, including the quantizers, using the following state-space formulation:

$$\begin{aligned} \mathbf{x}(n+1) &= [\mathbf{A}\mathbf{x}(n) + \mathbf{b}u(n)]_Q \\ y(n) &= \mathbf{c}^T\mathbf{x}(n) + du(n) \end{aligned} \tag{164}$$

  where $[x]_Q$ indicates the quantized value of $x$, **A** is the state matrix, **b** is the input vector, **c** is the output vector, and $d$ represents the direct connection between the input and output of the filter.

# Elimination of zero-input limit cycles

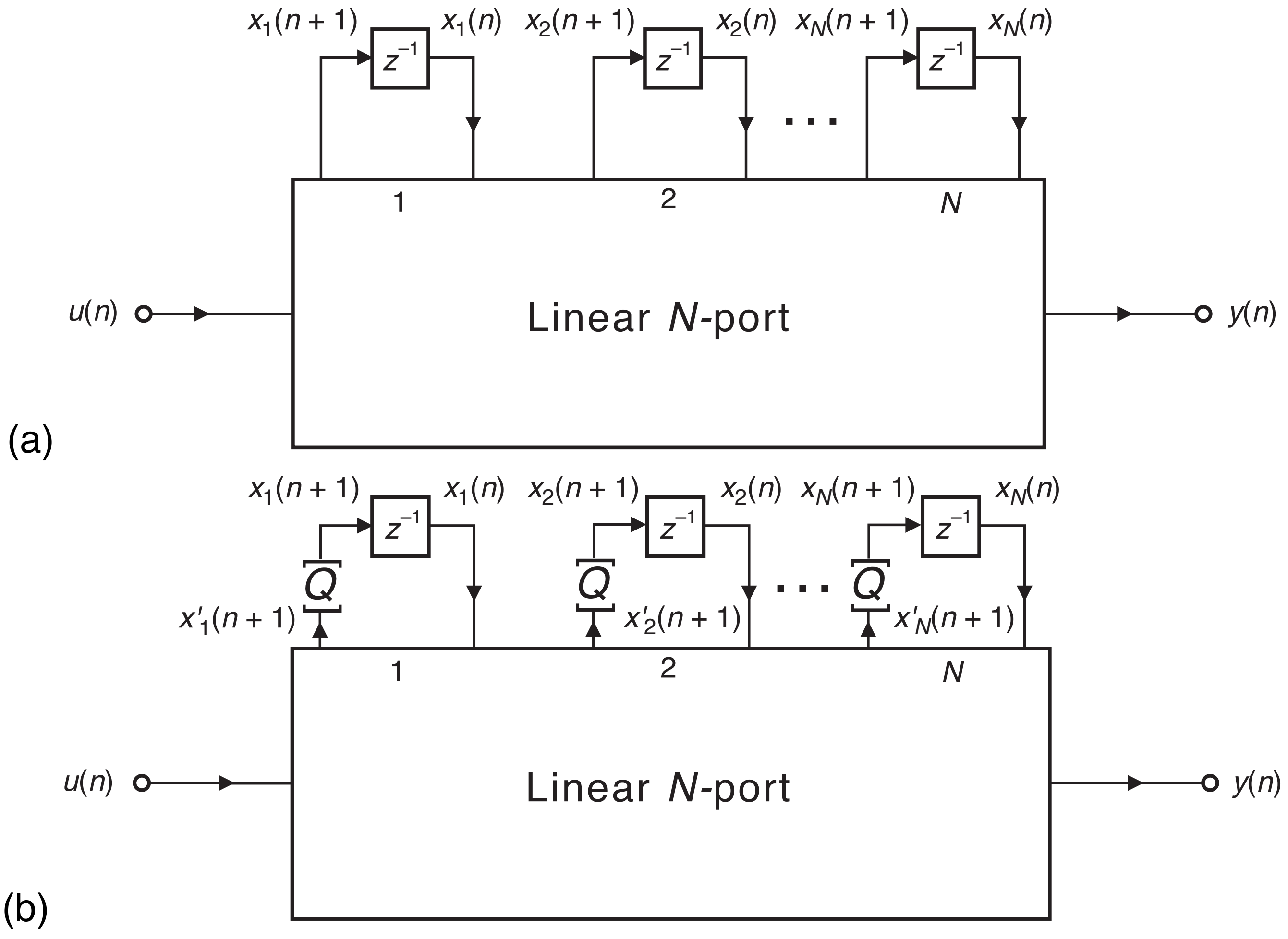


Figure 29: Digital filter networks: (a) ideal; (b) with quantizers at the state variables.

# Elimination of zero-input limit cycles

- In order to analyze zero-input limit cycles, it is sufficient to consider the recursive part of the state equation given by

$$\mathbf{x}(k+1) = [\mathbf{A}\mathbf{x}(k)]_Q = [\mathbf{x}'(k+1)]_Q \tag{165}$$

where the quantization operations $[\cdot]_Q$ are nonlinear operations such as truncation, rounding, or overflow.

## Elimination of zero-input limit cycles

- The basis for the elimination of nonlinear oscillations is given by Theorem 11.1 below.
- **Theorem:** If a stable digital filter has a state matrix **A** and, for any $N \times 1$ vector **x** there exists a diagonal positive-definite matrix **G**, such that

$$\hat{\mathbf{x}}^{\mathsf{T}}(\mathbf{G} - \mathbf{A}^{\mathsf{T}}\mathbf{G}\mathbf{A})\hat{\mathbf{x}} \geq 0 \tag{166}$$

  then the granular zero-input limit cycles can be eliminated if the quantization is performed through magnitude truncation.

- **Proof:** Consider a non-negative pseudo-energy Lyapunov function given by

$$p(\mathbf{x}(n)) = \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \tag{167}$$

# Elimination of zero-input limit cycles

- The energy variation in a single iteration can be defined as

$$\begin{aligned}
\Delta p(n+1) &= p(\mathbf{x}(n+1)) - p(\mathbf{x}(n)) \\
&= \mathbf{x}^{\mathsf{T}}(n+1)\mathbf{G}\mathbf{x}(n+1) - \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \\
&= [\mathbf{x}'^{\mathsf{T}}(n+1)]_{Q}\mathbf{G}[\mathbf{x}'(n+1)]_{Q} - \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \\
&= [\mathbf{A}\mathbf{x}(n)]_{Q}^{\mathsf{T}}\mathbf{G}[\mathbf{A}\mathbf{x}(n)]_{Q} - \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \\
&= [\mathbf{A}\mathbf{x}(n)]^{\mathsf{T}}\mathbf{G}[\mathbf{A}\mathbf{x}(n)] - \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \\
&\quad - \sum_{i=1}^{N}(x_i'^2(n+1) - x_i^2(n+1))g_i \\
&= \mathbf{x}^{\mathsf{T}}(n)[\mathbf{A}^{\mathsf{T}}\mathbf{G}\mathbf{A} - \mathbf{G}]\mathbf{x}(n) - \sum_{i=1}^{N}(x_i'^2(n+1) - x_i^2(n+1))g_i
\end{aligned} \tag{168}$$

where $g_i$ are the diagonal elements of **G**.

# Elimination of zero-input limit cycles

- If quantization is performed through magnitude truncation, then the errors due to granular quantization and overflow are such that

$$|x_i(n+1)| \leq |x'_i(n+1)| \tag{169}$$

  for all $i$ and $n$. Therefore, if equation (166) holds, from equation (168), we have that

$$\Delta p(n+1) \leq 0 \tag{170}$$

- If a digital filter is implemented with finite-precision arithmetic, within a finite number of samples after the input signal becomes zero, the output signal will become either a periodic oscillation or zero.

# Elimination of zero-input limit cycles

- Periodic oscillations, with nonzero amplitude, can not be sustained if $\Delta p(n+1) \leq 0$, as shown above.
- Therefore, equations (166) and (169) are sufficient conditions to guarantee the elimination of granular zero-input limit cycles on a recursive digital filter.
- Note that the condition given in equation (166) is equivalent to requiring that **F** be positive semidefinite, where

$$\mathbf{F} = \left(\mathbf{G} - \mathbf{A}^{T}\mathbf{G}\mathbf{A}\right) \tag{171}$$

- It is worth observing that for any stable state matrix **A**, its eigenvalues are inside the unit circle, and there will always be a positive-definite and symmetric matrix **G** such that **F** is symmetric and positive semidefinite.

# Elimination of zero-input limit cycles

- However, if **G** is not diagonal, the quantization process required to eliminate zero-input limit cycles is extremely complicated, as the quantization operation in each quantizer is coupled to the others.
- On the other hand, if there is a matrix **G** which is diagonal and positive definite such that **F** is positive semidefinite, then zero-input limit cycles can be eliminated by simple magnitude truncation.
- In the following theorem, we will state more specific conditions regarding the elimination of zero-input limit cycles in second-order systems.

## Elimination of zero-input limit cycles

- **Theorem:** Given a $2 \times 2$ stable state matrix **A**, there is a diagonal positive-definite matrix **G**, such that **F** is positive semidefinite, if and only if

$$a_{12}a_{21} \geq 0 \tag{172}$$

or

$$\left.\begin{array}{l} a_{12}a_{21} < 0 \\ |a_{11} - a_{22}| + \det(\mathbf{A}) \leq 1 \end{array}\right\} \tag{173}$$

## Elimination of zero-input limit cycles

- **Proof:** Let $\mathbf{G} = (\mathbf{T}^{-1})^2$ be a diagonal positive-definite matrix, such that $\mathbf{T}$ is a diagonal nonsingular matrix.
- Therefore, we can write $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{T}^{-1}\mathbf{T}^{-1} - \mathbf{A}^{\mathsf{T}}\mathbf{T}^{-1}\mathbf{T}^{-1}\mathbf{A} \tag{174}$$

and then

$$\begin{aligned}\mathbf{T}^{\mathsf{T}}\mathbf{F}\mathbf{T} &= \mathbf{T}^{\mathsf{T}}\mathbf{T}^{-1}\mathbf{T}^{-1}\mathbf{T} - \mathbf{T}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{T}^{-1}\mathbf{T}^{-1}\mathbf{A}\mathbf{T} \\ &= \mathbf{I} - (\mathbf{T}^{-1}\mathbf{A}\mathbf{T})^{\mathsf{T}}(\mathbf{T}^{-1}\mathbf{A}\mathbf{T}) \\ &= \mathbf{I} - \mathbf{M} \end{aligned} \tag{175}$$

with $\mathbf{M} = (\mathbf{T}^{-1}\mathbf{A}\mathbf{T})^{\mathsf{T}}(\mathbf{T}^{-1}\mathbf{A}\mathbf{T})$, as $\mathbf{T}^{\mathsf{T}} = \mathbf{T}$.

- Since the matrix $(\mathbf{I} - \mathbf{M})$ is symmetric and real, its eigenvalues are real.

# Elimination of zero-input limit cycles

- This matrix is then positive semidefinite if and only if its eigenvalues are non-negative, or, equivalently, if and only if its trace and determinant are non-negative. We then have that

$$\det\{\mathbf{I}-\mathbf{M}\} = 1 + \det\{\mathbf{M}\} - \mathrm{tr}\{\mathbf{M}\} = 1 + (\det\{\mathbf{A}\})^2 - \mathrm{tr}\{\mathbf{M}\} \tag{176}$$

$$\mathrm{tr}\{\mathbf{I}-\mathbf{M}\} = 2 - \mathrm{tr}\{\mathbf{M}\} \tag{177}$$

- For a stable digital filter, it is easy to verify that $\det\{\mathbf{A}\} < 1$, and then

$$\mathrm{tr}\{\mathbf{I}-\mathbf{M}\} > \det\{\mathbf{I}-\mathbf{M}\} \tag{178}$$

- Hence, the condition $\det\{\mathbf{I}-\mathbf{M}\} \geq 0$ is necessary and sufficient to guarantee that $(\mathbf{I}-\mathbf{M})$ is positive semidefinite.

# Elimination of zero-input limit cycles

- From the definition of **M**, and using $\alpha = t_{22}/t_{11}$, then

$$\det\{\mathbf{I} - \mathbf{M}\} = 1 + (\det\{\mathbf{A}\})^2 - \left(a_{11}^2 + \alpha^2 a_{12}^2 + \frac{a_{21}^2}{\alpha^2} + a_{22}^2\right) \tag{179}$$

- By calculating the maximum of the equation above with respect to $\alpha$, we get an optimal $\alpha^\star$ such that

$$(\alpha^\star)^2 = \left|\frac{a_{21}}{a_{12}}\right| \tag{180}$$

and then

$$\begin{aligned} \det{}^\star\{\mathbf{I} - \mathbf{M}\} &= 1 + (\det\{\mathbf{A}\})^2 - (a_{11}^2 + 2|a_{12}a_{21}| + a_{22}^2) \\ &= (1 + \det\{\mathbf{A}\})^2 - (\mathrm{tr}\{\mathbf{A}\})^2 + 2(a_{12}a_{21} - |a_{12}a_{21}|) \end{aligned} \tag{181}$$

where $\det^\star$ denotes the maximum value of the respective determinant.

# Elimination of zero-input limit cycles

- We now analyze two separate cases to guarantee that $\det^{\star}\{\mathbf{I}-\mathbf{M}\} \geq 0$.
  - If

$$a_{12}a_{21} \geq 0 \tag{182}$$

then

$$\begin{aligned} \det{}^{\star}\{\mathbf{I}-\mathbf{M}\} &= (1+\det\{\mathbf{A}\})^2 - (\mathrm{tr}\{\mathbf{A}\})^2 \\ &= (1+\alpha_2)^2 - (-\alpha_1)^2 \\ &= (1+\alpha_1+\alpha_2)(1-\alpha_1+\alpha_2) \end{aligned} \tag{183}$$

where $\alpha_1 = -\mathrm{tr}\{\mathbf{A}\}$ and $\alpha_2 = \det\{\mathbf{A}\}$ are the filter denominator coefficients. It can be verified that, for a stable filter, $(1+\alpha_1+\alpha_2)(1-\alpha_1+\alpha_2) > 0$, and then equation (182) implies that $(\mathbf{I}-\mathbf{M})$ is positive definite.

# Elimination of zero-input limit cycles

- (cont.)
  - If

$$a_{12}a_{21} < 0 \tag{184}$$

then

$$\begin{aligned}\det{}^{\star}\{\mathbf{I}-\mathbf{M}\} &= 1 + (\det\{\mathbf{A}\})^2 - (a_{11}^2 - 2a_{12}a_{21} + a_{22}^2) \\ &= (1-\det\{\mathbf{A}\})^2 - (a_{11} - a_{22})^2 \end{aligned} \tag{185}$$

This equation is greater than or equal to zero, if and only if

$$|a_{11} - a_{22}| + \det\{\mathbf{A}\} \leq 1 \tag{186}$$

## Elimination of zero-input limit cycles

- Therefore, either equation (182) or equations (184) and (186) are the necessary and sufficient conditions for the existence of a diagonal matrix

$$\mathbf{T} = \text{diag}\{t_{11}\ t_{22}\} \tag{187}$$

with

$$\frac{t_{22}}{t_{11}} = \sqrt{\left|\frac{a_{21}}{a_{12}}\right|} \tag{188}$$

such that **F** is positive semidefinite.

## Elimination of zero-input limit cycles

- It is worth observing that the above theorem gives the conditions for the matrix **F** to be positive semidefinite for second-order sections.
- In the example below, we illustrate the limit-cycle elimination process by showing, without resorting to Theorem 11.2, that a given second-order structure is free from zero-input limit cycles.
- The reader is encouraged to apply the theorem to show the same result.

# Example 11.11

- Examine the possibility of eliminating limit cycles in the network of Figure 30.

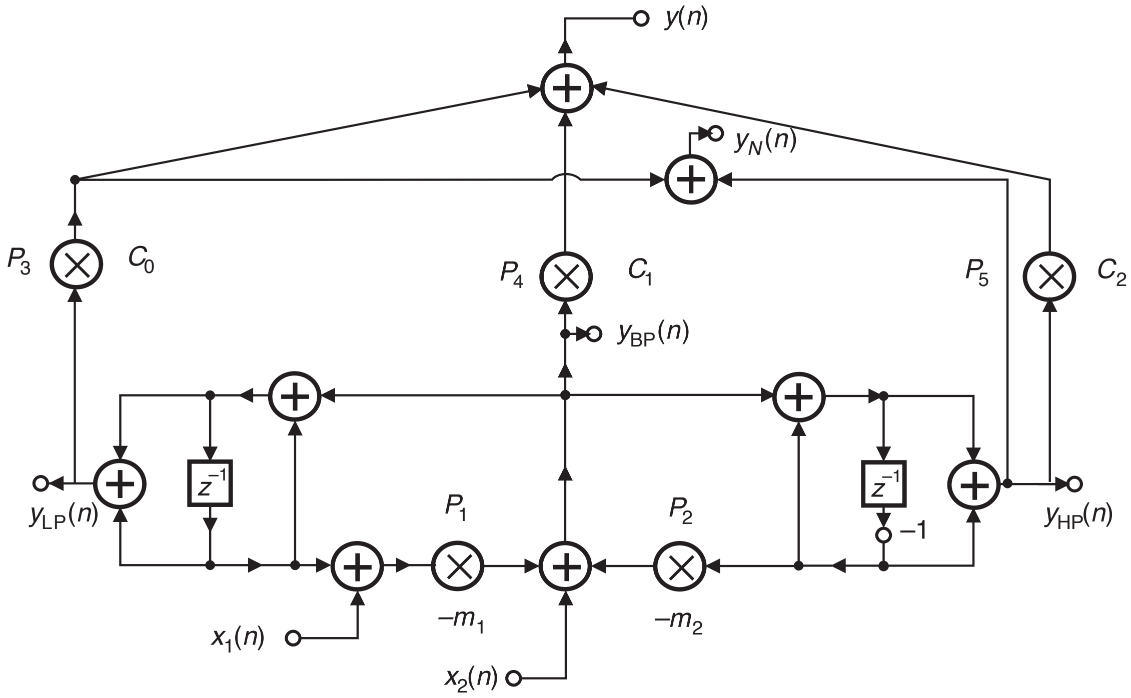


Figure 30: General-purpose network.

# Example 11.11 - Solution

- The structure in Figure 30 realizes lowpass, bandpass, and highpass transfer functions simultaneously (with subscripts LP, BP, and HP, respectively).
- The structure also realizes a transfer function with zeros on the unit circle, using the minimum number of multipliers.
- The characteristic polynomial of the structure is given by

$$D(z) = z^2 + (m_1 - m_2)z + m_1 + m_2 - 1 \tag{189}$$

- In order to guarantee stability, the multiplier coefficients $m_1$ and $m_2$ should fall in the range

$$\left.\begin{aligned} m_1 &> 0 \\ m_2 &> 0 \\ m_1 + m_2 &< 2 \end{aligned}\right\} \tag{190}$$

# Example 11.11 - Solution

- Figure 31 depicts the recursive part of the structure in Figure 30, including the quantizers.

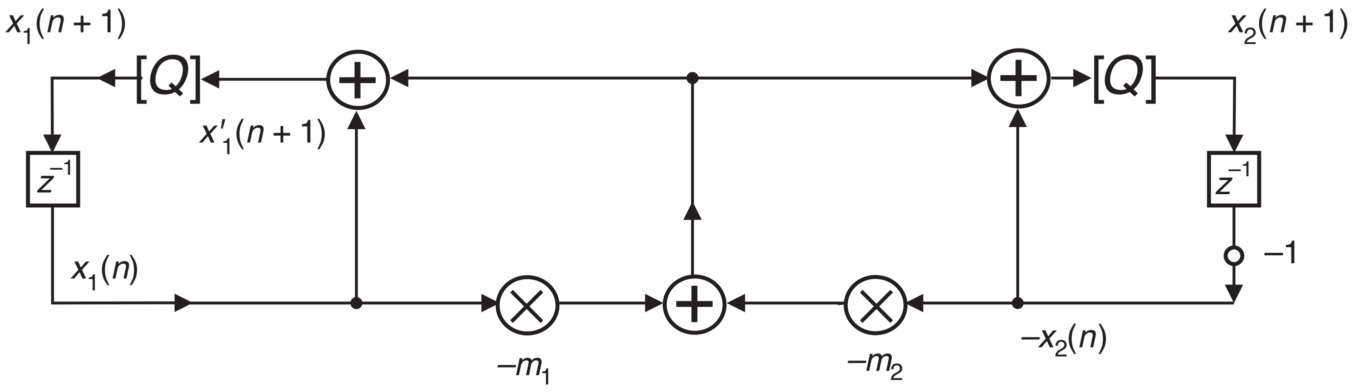


Figure 31: Recursive part of the network in Figure 30.

# Example 11.11 - Solution

- The zero-input state-space equation for the structure in Figure 31 is

$$\mathbf{x}'(n+1) = \begin{bmatrix} x_1'(n+1) \\ x_2'(n+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} \tag{191}$$

with

$$\mathbf{A} = \begin{bmatrix} (1-m_1) & m_2 \\ -m_1 & (m_2-1) \end{bmatrix} \tag{192}$$

- By applying quantization to $\mathbf{x}'(n+1)$, we find

$$\mathbf{x}(n+1) = [\mathbf{x}'(n+1)]_Q = [\mathbf{A}\mathbf{x}(n)]_Q \tag{193}$$

# Example 11.11 - Solution

- A quadratic positive-definite function can be defined as

$$p(\mathbf{x}(n)) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{G}\mathbf{x}(n) = \frac{x_1^2}{m_2} + \frac{x_2^2}{m_1} \tag{194}$$

with

$$\mathbf{G} = \begin{bmatrix} \frac{1}{m_2} & 0 \\ 0 & \frac{1}{m_1} \end{bmatrix} \tag{195}$$

which is positive definite, since from equation (190), $m_1 > 0$ and $m_2 > 0$.

# Example 11.11 - Solution

- An auxiliary energy increment is then

$$\begin{aligned}\Delta p_0(n+1) &= p(\mathbf{x}'(n+1)) - p(\mathbf{x}(n)) \\ &= \mathbf{x}'^{\mathsf{T}}(n+1)\mathbf{G}\mathbf{x}'(n+1) - \mathbf{x}^{\mathsf{T}}(n)\mathbf{G}\mathbf{x}(n) \\ &= \mathbf{x}^{\mathsf{T}}(n)[\mathbf{A}^{\mathsf{T}}\mathbf{G}\mathbf{A} - \mathbf{G}]\mathbf{x}(n) \\ &= (m_1 + m_2 - 2)\left(x_1(n)\sqrt{\frac{m_1}{m_2}} - x_2(n)\sqrt{\frac{m_2}{m_1}}\right)^2 \qquad (196)\end{aligned}$$

## Example 11.11 - Solution

- Since from equation (190), $m_1 + m_2 < 2$, then

$$\left.\begin{array}{ll} \Delta p_0(n+1) = 0, & \text{for } x_1(n) = x_2(n)\dfrac{m_2}{m_1} \\ \Delta p_0(n+1) < 0, & \text{for } x_1(n) \neq x_2(n)\dfrac{m_2}{m_1} \end{array}\right\} \tag{197}$$

- Now, if magnitude truncation is applied to quantize the state variables, then $p(\mathbf{x}(n)) \leq p(\mathbf{x}'(n))$, which implies that

$$\Delta p(\mathbf{x}(n)) = p(\mathbf{x}(n+1)) - p(\mathbf{x}(n)) \leq 0 \tag{198}$$

  and then $p(\mathbf{x}(n))$ is a Lyapunov function.

- Overall, when no quantization is applied in the structure of Figure 30, no self-sustained oscillations occur if the stability conditions of equation (190) are satisfied.

## Example 11.11 - Solution

- If, however, quantization is applied to the structure, as shown in Figure 31, oscillations may occur.

- Using magnitude truncation, then $|x_i(n)| \leq |x'_i(n)|$, and under these circumstances, $p(\mathbf{x}(n))$ decreases during the subsequent iterations, and eventually the oscillations disappear, with

$$\mathbf{x}(n) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (199)$$

being the only possible equilibrium point.

## Elimination of constant-input limit cycles

- As seen above, the sufficient conditions for the elimination of zero-input limit cycles are well established.
- However, if the system input is a nonzero constant, limit cycles may still occur.
- It is worth observing that the response of a stable linear system to a constant input signal should also be a constant signal.
- In the associated literature, a theorem is presented that establishes how constant-input limit cycles can also be eliminated in digital filters in which zero-input limit cycles are eliminated. This theorem is as follows.

## Elimination of constant-input limit cycles

- **Theorem:** Assume that the general digital filter in Figure 29b does not sustain zero-input limit cycles and that

$$\left.\begin{aligned} \mathbf{x}(n+1) &= [\mathbf{A}\mathbf{x}(n) + \mathbf{B}u(n)]_Q \\ y(n) &= \mathbf{c}^T\mathbf{x}(n) + du(n) \end{aligned}\right\} \tag{200}$$

- Constant-input limit cycles can also be eliminated, by modifying the structure in Figure 29b, as shown in Figure 32, where

$$\mathbf{p} = [p_1\; p_2 \cdots p_n]^T = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \tag{201}$$

and $\mathbf{p}u_0$ must be representable in the machine wordlength, where $u_0$ is a constant input signal.

# Elimination of constant-input limit cycles

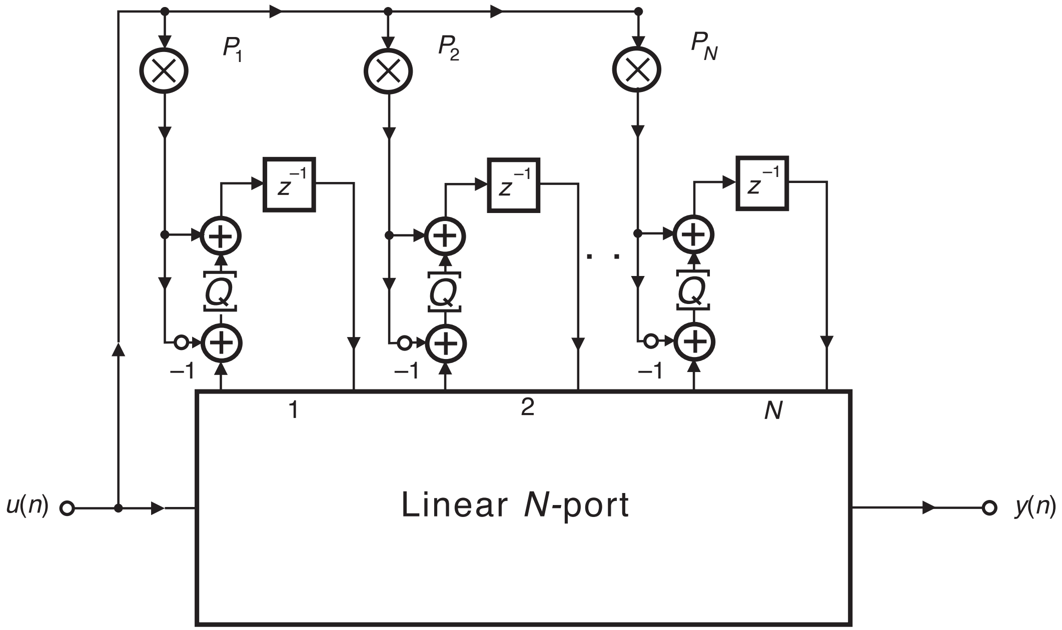


Figure 32: Modified N th-order network for the elimination of constant-input limit cycles.

# Elimination of constant-input limit cycles

- **Proof:** Since the structure of Figure 29b is free from zero-input limit cycles, the autonomous system

$$\mathbf{x}(n+1) = [\mathbf{A}\mathbf{x}(n)]_Q \tag{202}$$

is such that

$$\lim_{n\to\infty} \mathbf{x}(n) = [0\,0\ldots 0]^T \tag{203}$$

- If $\mathbf{p}$ is as given in equation (201), the modified structure of Figure 32 is described by

$$\begin{aligned}\mathbf{x}(n+1) &= [\mathbf{A}\mathbf{x}(n) - \mathbf{p}u_0 + \mathbf{B}u_0]_Q + \mathbf{p}u_0\\ &= [\mathbf{A}\mathbf{x}(n) - \mathbf{I}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}u_0 + (\mathbf{I}-\mathbf{A})(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}u_0]_Q + \mathbf{p}u_0\\ &= [\mathbf{A}(\mathbf{x}(n) - \mathbf{p}u_0)]_Q + \mathbf{p}u_0 \end{aligned} \tag{204}$$

# Elimination of constant-input limit cycles

- Defining

$$\hat{\mathbf{x}}(n) = \mathbf{x}(n) - \mathbf{p}u_0 \tag{205}$$

then, from equation (204), we can write that

$$\hat{\mathbf{x}}(n+1) = [\mathbf{A}\hat{\mathbf{x}}(n)]_Q \tag{206}$$

- Which is the same as equation (202), except for the transformation in the state variable. Hence, as $\mathbf{p}u_0$ is machine representable (that is, $\mathbf{p}u_0$ can be exactly calculated with the available wordlength), equation (206) also represents a stable system free from constant-input limit cycles.

- If the quantization of the structure depicted in Figure 29b is performed using magnitude truncation, the application of the strategy of Theorem 11.3 leads to the so-called controlled rounding method.

## Elimination of constant-input limit cycles

- The constraints imposed by requiring that $\mathbf{p}u_0$ be machine representable reduce the number of structures in which the technique described by Theorem 11.3 applies.
- However, there are a large number of second-order sections and wave digital filter structures in which these requirements are automatically met.
- In fact, a large number of research papers have been published proposing new structures which are free from zero-input limit cycles, and free from constant-input limit cycles.
- However, the analysis procedures for the generation of these structures are not unified and here we have aimed to provide a unified framework leading to a general procedure to generate structures which are free from granular limit cycles.

## Example 11.12

- Show that by placing the input signal at the point denoted by $x_1(n)$, the structure in Figure 30 is free from constant-input limit cycles.

## Example 11.12 - Solution

- The second-order section of Figure 30, with a constant input $x_1(n) = u_0$, can be described by

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \begin{bmatrix} -m_1 \\ -m_1 \end{bmatrix} u_0 \qquad (207)$$

with **p** such that

$$\mathbf{p} = \begin{bmatrix} m_1 & -m_2 \\ m_1 & 2 - m_2 \end{bmatrix}^{-1} \begin{bmatrix} -m_1 \\ -m_1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad (208)$$

- Therefore, $\mathbf{p}u_0$ is clearly machine representable, for any $u_0$, and the constant-input limit cycles can be eliminated, as depicted in Figure 33.

# Example 11.12 - Solution

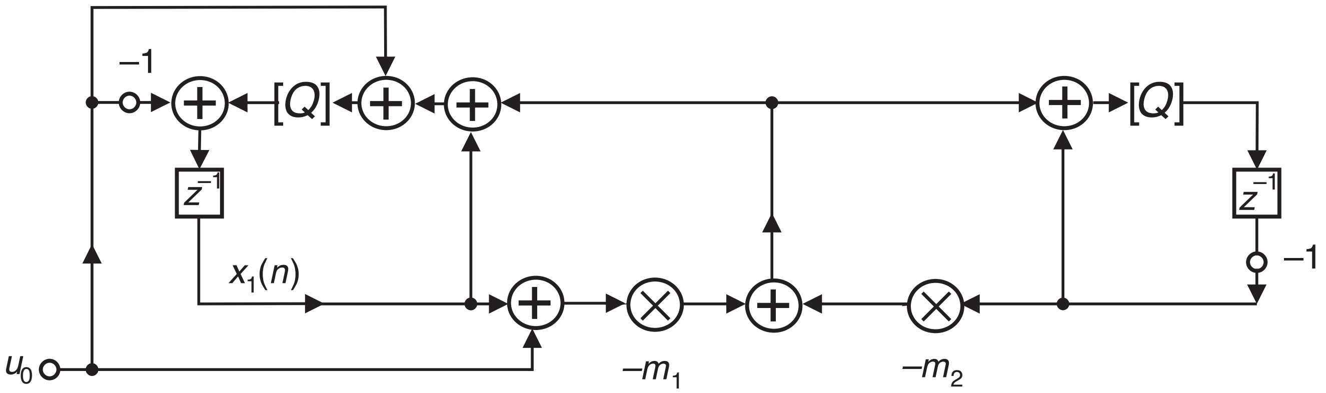


Figure 33: Elimination of constant-input limit cycles in the structure of Figure 30.

# Example 11.13

- For the second-order state-variable realization as given in Figure 4.23 of the book, discuss the elimination of constant-input limit-cycles in a distributed arithmetic implementation as the one in Section 11.4.

# Example 11.13 - Solution

- In a regular implementation, as the one in Figure 11, to eliminate zero-input limit cycles in the state-space realization, the state variables $x_1(n)$ and $x_2(n)$ must be calculated by ALU$_1$ and ALU$_2$ in double precision, and then properly quantized, before being loaded into the shift-registers SR$_2$ and SR$_3$.
- However, it can be shown that no double-precision computation is necessary to avoid zero-input limit cycles when implementing the state-space realization with the distributed arithmetic approach.
- To eliminate constant-input limit cycles, the state-space realization shown in Figure 34 requires that the state variable $x_1(n)$ must be computed by ALU$_1$ in double precision, and then properly quantized to be subtracted from the input signal.

# Example 11.13 - Solution

- To perform this subtraction, register $A$ in Figure 9 must be multiplexed with another register that contains the input signal $x(n)$, in order to guarantee that the signal arriving at the adder, at the appropriate instant of time, is the complemented version of $x(n)$, instead of a signal coming from the memory.

- In such a case, the content of the ROM of ALU$_1$ must be generated as

$$s'_{1j} = a_{11}x_{1j}(n) + a_{12}x_{2j}(n) + a_{11}x_j(n); \text{ for the ROM of the ALU}_1 \quad (209)$$

while ALU$_3$ is filled in the same fashion as given in equation (52) and for ALU$_2$ the content is the same as in equation (51) with $b_2$ replaced by $a_{21}$.

# Example 11.13 - Solution

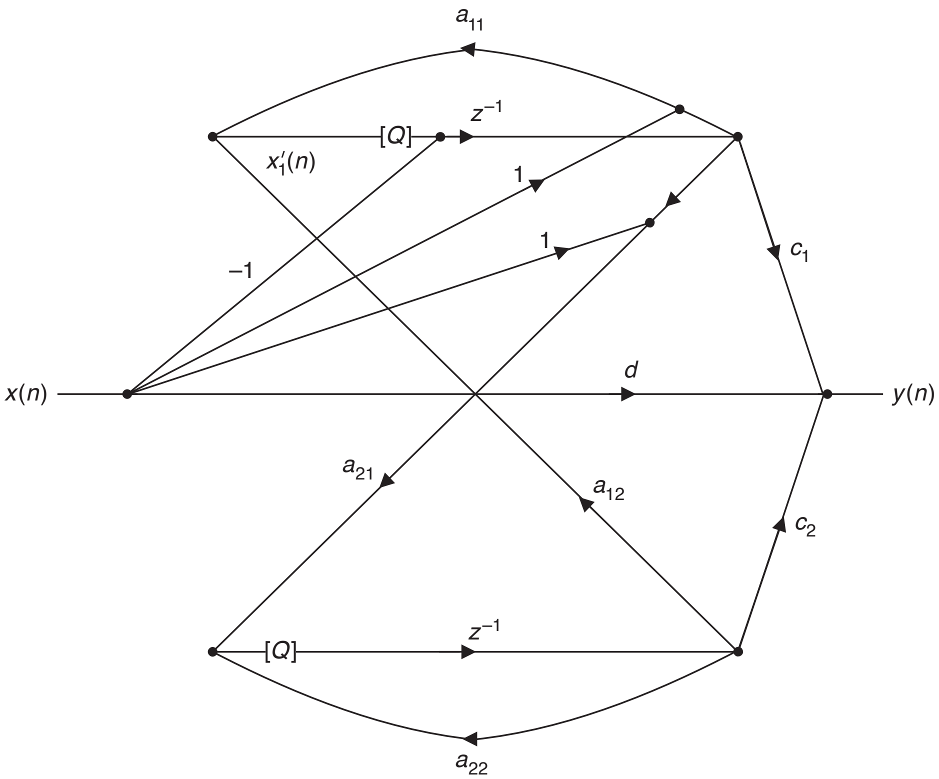


Figure 34: State-space realization immune to constant-input limit cycles.

# Forced-response stability of digital filters with nonlinearities due to overflow

- The stability analysis of the forced response of digital filters that include nonlinearities to control overflow must be performed considering input signals for which in the ideal linear system the overflow level is never reached after a given instant $n_0$.

- In this way, we can verify whether the real system output will recover after an overflow has occurred before instant $n_0$.

- Although the input signals considered are in a particular class of signals, it can be shown that if the real system recovers for these signals, it will also recover after each overflow, for any input signal, if the recovery period is shorter than the time between two consecutive overflows.

- Consider the ideal linear system as depicted in Figure 35a and the real nonlinear system as depicted in Figure 35b.

# Forced-response stability of digital filters with nonlinearities due to overflow

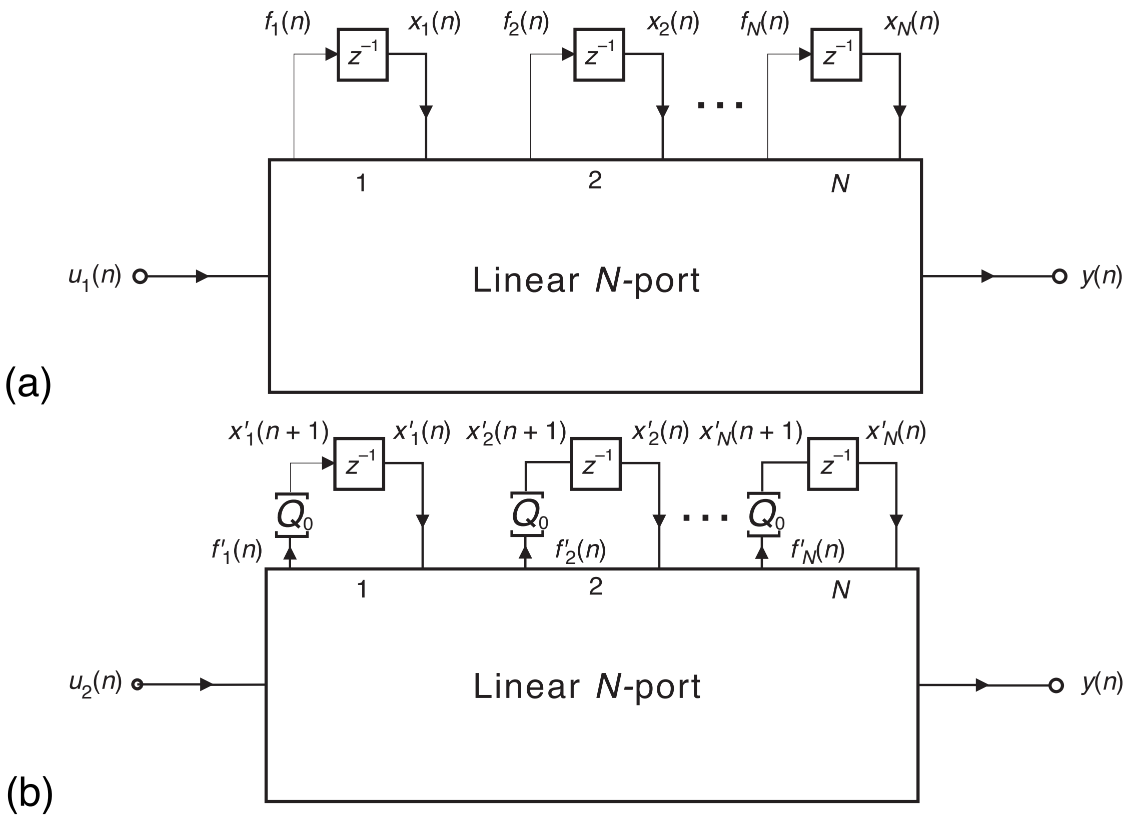


Figure 35: General digital filter networks: (a) ideal; (b) with quantizers at the state variables.

# Forced-response stability of digital filters with nonlinearities due to overflow

- The linear system illustrated in Figure 35a is described by the equations

$$\mathbf{f}(n) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}u_1(n) \tag{210}$$

$$\mathbf{x}(n) = \mathbf{f}(n-1) \tag{211}$$

and the nonlinear system illustrated in Figure 35b is described by the equations

$$\mathbf{f}'(n) = \mathbf{A}\mathbf{x}'(n) + \mathbf{B}u_2(n) \tag{212}$$

$$\mathbf{x}'(n) = [\mathbf{f}'(n-1)]_{Q_0} \tag{213}$$

where $[u]_{Q_0}$ denotes quantization of $u$, in the case where an overflow occurs.

- We assume that the output signal of the nonlinear system is properly scaled so that no oscillation due to overflow occurs if it does not occur at the state variables.

# Forced-response stability of digital filters with nonlinearities due to overflow

- The response of the nonlinear system of Figure 35b is stable if, when $u_1(n) = u_2(n)$, the difference between the outputs of the linear $N$-port system of Figure 35a, $\mathbf{f}(n)$, and the outputs of the linear $N$-port system of Figure 35b, $\mathbf{f}'(n)$, tends to zero as $n \rightarrow \infty$.

- In other words, if we define an error signal $\mathbf{e}(n) = \mathbf{f}'(n) - \mathbf{f}(n)$, then

$$\lim_{n \rightarrow \infty} \mathbf{e}(n) = [0\ 0 \ldots 0]^{\mathsf{T}} \qquad (214)$$

- If the difference between the output signals of the two linear $N$-port systems converges to zero, this implies that the difference between the state variables of both systems will also tend to zero.

# Forced-response stability of digital filters with nonlinearities due to overflow

- This can be deduced from equations (210) and (212), which yield

$$\mathbf{e}(n) = \mathbf{f}'(n) - \mathbf{f}(n) = \mathbf{A}[\mathbf{x}'(n) - \mathbf{x}(n)] = \mathbf{A}\mathbf{e}'(n) \tag{215}$$

  where $\mathbf{e}'(n) = \mathbf{x}'(n) - \mathbf{x}(n)$ is the difference between the state variables of both systems.

- Equation (215) is equivalent to saying that $\mathbf{e}(n)$ and $\mathbf{e}'(n)$ are the output and input signals of a linear $N$-port system described by matrix $\mathbf{A}$, which is the transition matrix of the original system.

- Then, from equation (214), the forced-response stability of the system in Figure 35b is equivalent to the zero-input response of the same system, regardless of the quantization characteristics $[\cdot]_{Q_0}$.

# Forced-response stability of digital filters with nonlinearities due to overflow

- Substituting equations (211) and (213) in equation (215), we have that

$$\mathbf{e}'(n) = [\mathbf{f}'(n-1)]_{Q_0} - \mathbf{f}(n-1) = [\mathbf{e}(n-1) + \mathbf{f}(n-1)]_{Q_0} - \mathbf{f}(n-1) \quad (216)$$

- By defining the time-varying vector $\mathbf{v}(\mathbf{e}(n), n)$ as

$$\mathbf{v}(\mathbf{e}(n), n) = [\mathbf{e}(n) + \mathbf{f}(n)]_{Q_0} - \mathbf{f}(n) \quad (217)$$

equation (216) can be rewritten as

$$\mathbf{e}'(n) = \mathbf{v}(\mathbf{e}(n-1), (n-1)) \quad (218)$$

- The nonlinear system described by equations (215)–(218) is depicted in Figure 36.

# Forced-response stability of digital filters with nonlinearities due to overflow

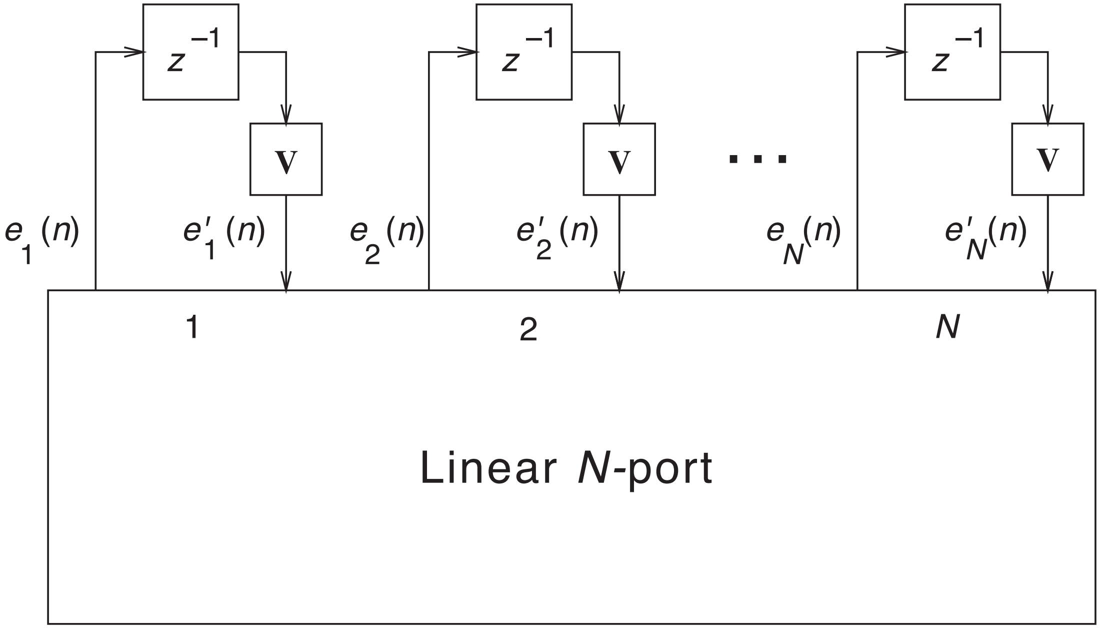


Figure 36: Nonlinear system relating the signals $\mathbf{e}'(n)$ and $\mathbf{e}(n)$.

# Forced-response stability of digital filters with nonlinearities due to overflow

- As we saw in Subsection 11.8.3, a system such as the one in Figure 36 is free from zero-input nonlinear oscillations if the nonlinearity $\mathbf{v}(\cdot, n)$ is equivalent to the magnitude truncation, that is

$$|\mathbf{v}(e_i(n), n)| < |e_i(n)|, \quad \text{for } i = 1, 2, \ldots, N \tag{219}$$

- If we assume that the internal signals are such that $|f_i(n)| \leq 1$, for $n > n_0$, then it can be shown that equation (219) remains valid whenever the quantizer $Q_0$ has overflow characteristics within the hatched region of Figure 37.

# Forced-response stability of digital filters with nonlinearities due to overflow

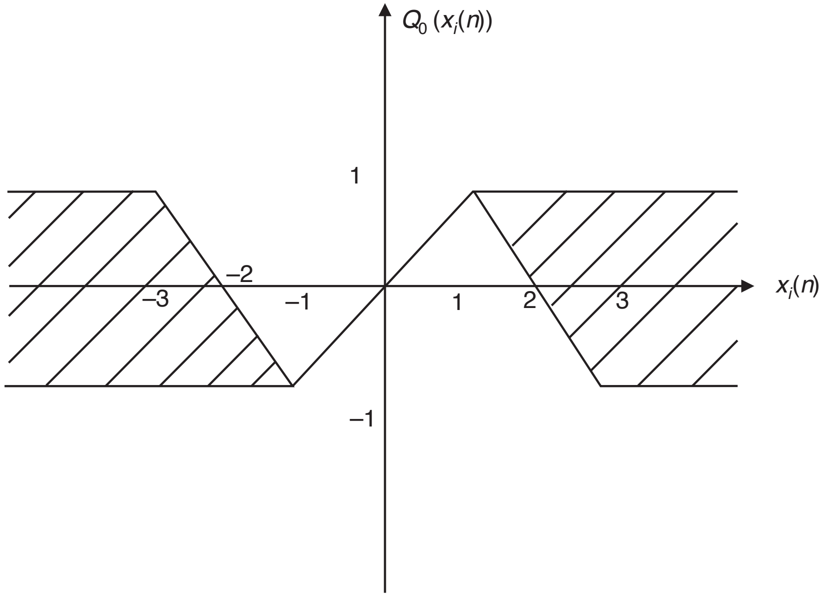


Figure 37: Region for the overflow nonlinearity which guarantees forced-response stability in networks which satisfy Theorem 11.1.

# Forced-response stability of digital filters with nonlinearities due to overflow

- Figure 37 can be interpreted as follows:
  - If $-1 \leq x_i(n) \leq 1$, then there should be no overflow.
  - If $1 \leq x_i(n) \leq 3$, then the overflow nonlinearity should be such that $2 - x_i(n) \leq Q_0(x_i(n)) \leq 1$.
  - If $-3 \leq x_i(n) \leq -1$, then the overflow nonlinearity should be such that $-1 \leq Q_0(x_i(n)) \leq -2 - x_i(n)$.
  - If $x_i(n) \geq 3$ or $x_i(n) \leq -3$, then $-1 \leq Q_0(x_i(n)) \leq 1$.
- It is important to note that the overflow nonlinearity of the saturation type (equation (162)) satisfies the requirements in Figure 37.
- Summarizing the above reasoning, one can state that a digital filter which is free of zero-input limit cycles, according to the condition of equation (166), is also forced-input stable, provided that the overflow nonlinearities are in the hatched regions of Figure 37.

# Do-it-yourself: Finite-precision digital signal processing

- **Experiment 11.1:** Let us play around digital representation in MATLAB. We concentrate our efforts here in the case $-1 < x < 0$, which yields different $(n+1)$-bit standard, one's-complement, two's-complement, and CSD representations.

- The standard sign-magnitude binary representation `xbin` can be obtained making $s_x = 1$ and following the procedure performed in equation (12), such that

```
x = abs(x); xbin = [1 zeros(1,n)];
for i=2:n+1,
   x = 2*x;
   if x >= 1,
      xbin(i) = 1; x = x-1;
   end;
end;
```

## Do-it-yourself: Finite-precision digital signal processing

- The one's-complement `xbin1` representation of `x` can be determined as

  ```
  xbin1 = [1 ~xbin(2:n+1)];
  ```

  where the `~x` operator determines the binary complement of `x` in MATLAB.

- For the two's-complement representation `xbin2`, we must add 1 to the least significant bit of `xbin1`. This can be performed, for instance, by detecting the last 0-bit in `xbin1`, which indicates the final position of the carry-over bit, such that

  ```
  xbin2 = xbin1;
  b = max(find(xbin1 == 0));
  xbin2(b:n+1) = ~xbin2(b:n+1);
  ```

## Do-it-yourself: Finite-precision digital signal processing

- We can then obtain the CSD representation `xCSD` from `xbin2`, following the algorithm described in Subsection 11.2.2:

```
delta = zeros(1,n+2); theta = zeros(1,n+1); xCSD =
theta;
x2aux = [xbin2(1) xbin2 0];
for i = n:-1:1,
   theta(i) = xor(x2aux(i+1),x2aux(i+2));
   delta(i) = and(~delta(i+1),theta(i));
   xCSD(i) = (1-2*x2aux(i))*delta(i);
end;
```

## Do-it-yourself: Finite-precision digital signal processing

- Using $n = 7$ and $x = -0.6875$ with the scripts above, results in the numerical representations seen in Table 5.

Table 5: Numerical 8-digit representations of $x = -0.6875$ in Experiment 11.1.

| Numerical Format | $[x]$ |
|---|---|
| Standard binary | 1.1011000 |
| One's complement | 1.0100111 |
| Two's complement | 1.0101000 |
| CSD | $\bar{1}$0101000 |

# Do-it-yourself: Finite-precision digital signal processing

- **Experiment 11.2:** Consider the digital filter structure depicted in Figure 38, whose state-space description is given by

$$\mathbf{x}(n+1) = \begin{bmatrix} (1-m_1) & -m_2 \\ m_1 & (m_2-1) \end{bmatrix} \mathbf{x}(n) + \begin{bmatrix} (2-m_1-m_2) \\ -(2-m_1-m_2) \end{bmatrix} u(n) \quad (220)$$

$$y(n) = \begin{bmatrix} -m_1 & -m_2 \end{bmatrix} \mathbf{x}(n) + (1-m_1-m_2)u(n) \quad (221)$$

## Do-it-yourself: Finite-precision digital signal processing

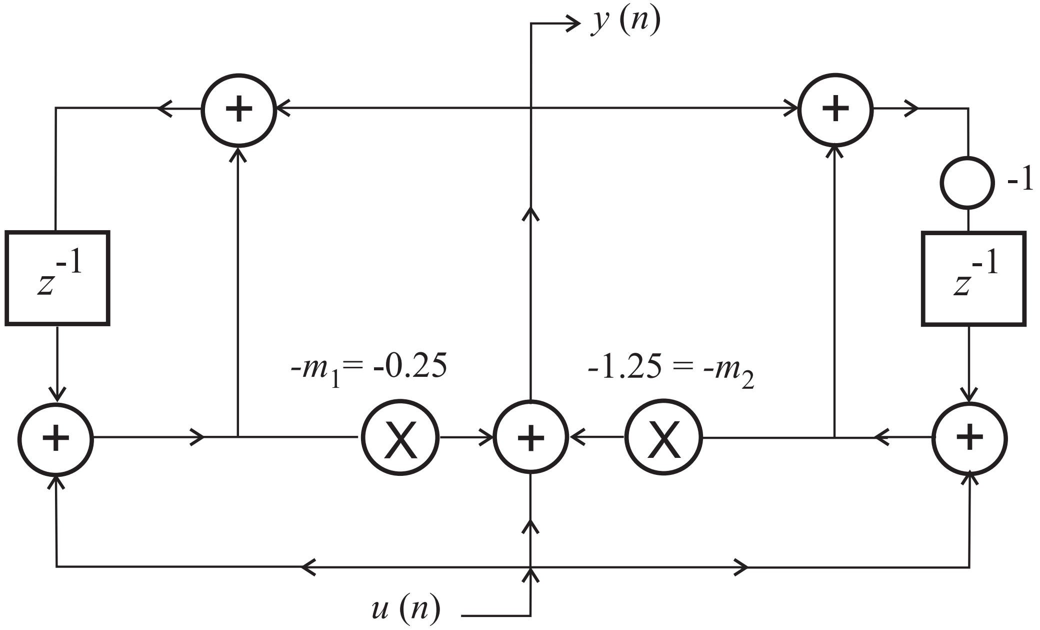


Figure 38: Digital filter structure.

## Do-it-yourself: Finite-precision digital signal processing

- The corresponding transfer function is

$$H(z) = \begin{bmatrix} -m_1 & -m_2 \end{bmatrix} \begin{bmatrix} (z-1+m_1) & m_2 \\ -m_1 & (z-m_2+1) \end{bmatrix}^{-1} \begin{bmatrix} (2-m_1-m_2) \\ -(2-m_1-m_2) \end{bmatrix} + (1-m_1-m_2) \tag{222}$$

which, after some cumbersome algebraic development, becomes

$$H(z) = \frac{N(z)}{D(z)} = -\frac{(m_1+m_2-1)z^2 + (m_1-m_2)z + 1}{z^2 + (m_1-m_2)z + (m_1+m_2-1)} \tag{223}$$

corresponding to an all-pass second-order block.

## Do-it-yourself: Finite-precision digital signal processing

- The transfer function from the filter input to the $-m_1$ multiplier is given by

$$F_1(z) = \frac{z^2 + 2(1 - m_2)z + 1}{z^2 + (m_1 - m_2)z + (m_1 + m_2 - 1)} \tag{224}$$

and the transfer function from the filter input to the $-m_2$ multiplier is

$$F_2(z) = \frac{z^2 + 2(m_1 - 1)z + 1}{z^2 + (m_1 - m_2)z + (m_1 + m_2 - 1)} \tag{225}$$

## Do-it-yourself: Finite-precision digital signal processing

- Determining the $\mathrm{L}_2$ norm for each scaling transfer function in a closed form is quite computationally intensive. Using MATLAB, however, this can be done numerically using few commands, such as:

```
m1 = 0.25; m2 = 1.25;
N1 = [1 2*(1-m2) 1]; N2 = [1 2*(m1-1) 1];
D = [1 (m1-m2) (m1+m2-1)];
np = 1000;
[F1,f] = freqz(N1,D,np); F1_2 =
sqrt((sum(abs(F1).^2))/np);
[F2,f] = freqz(N2,D,np); F2_2 =
sqrt((sum(abs(F2).^2))/np);
```

## Do-it-yourself: Finite-precision digital signal processing

- As a result, `F1_2` and `F2_2` are equal to $1.7332$ and $1.1830$, respectively.
- Then, we should scale the filter input by a factor of

$$\lambda = \frac{1}{\max_{i=1,2} \left[\|F_i(z)\|_2\right]} = \frac{1}{\max\left[1.7332, 1.1830\right]} = 0.5770 \qquad (226)$$

and compensate for this by multiplying the filter output by $g = \frac{1}{\lambda} = 1.7332$.

## Do-it-yourself: Finite-precision digital signal processing

- The transfer functions from the outputs of both multipliers $-m_1$ and $-m_2$ to the filter output are expressed as

$$G_1(z) = G_2(z) = H(z) \tag{227}$$

such that

$$\|G_1(z)\|_2 = \|G_2(z)\|_2 = 1 \tag{228}$$

as $H(z)$ represents an all-pass filter with unit gain.

- Therefore, considering the scaling operation performed as above, the output noise variance is given by

$$\sigma_y^2 = 3g^2\sigma_e^2 + \sigma_e^2 = 10.0120\sigma_e^2 \tag{229}$$

where the factor $3$ accounts for the noise sources in the input scaling, $-m_1$, and $-m_2$ multipliers.