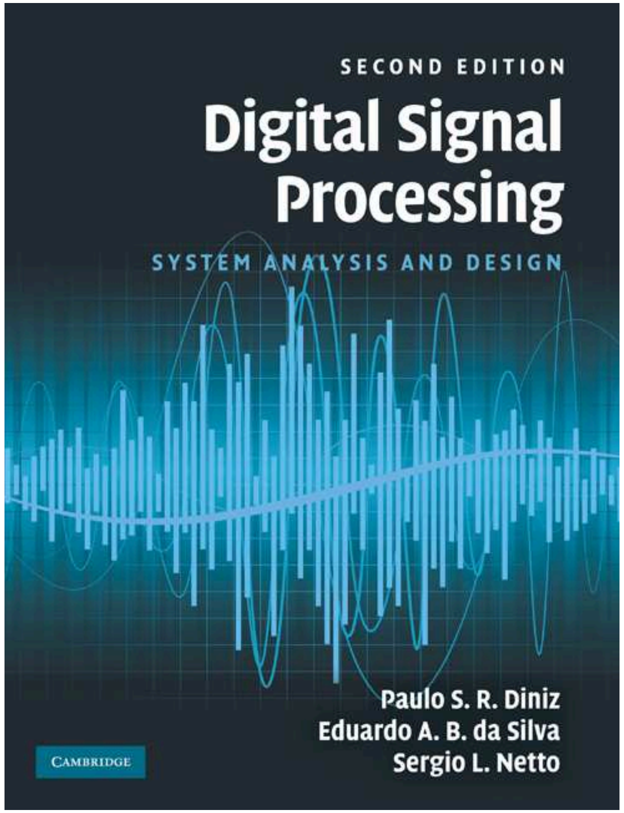
Digital Filters



Paulo S. R. Diniz

Eduardo A. B. da Silva

Sergio L. Netto

diniz,eduardo,sergioln@lps.ufrj.br

September 2010

Contents

- Basic structures of nonrecursive digital filters
  - Direct form
  - Cascade form
  - Linear-phase form
- Basic structures of recursive digital filters
  - Direct forms
  - Cascade form
  - Parallel form
- Digital network analysis
- State-space description

Contents

Introduction

- In the previous chapters, we studied different ways of describing discrete-time systems that are linear and time invariant. It was verified that the z transform greatly simplifies the analysis of discrete-time systems, especially those initially described by a difference equation.
- In this chapter, we study several structures used to realize a given transfer function associated with a specific difference equation through the use of the z transform.
- The transfer functions considered here will be of the polynomial form (nonrecursive filters) and of the rational-polynomial form (recursive filters).
- In the nonrecursive case we emphasize the existence of the important subclass of linear-phase filters.

Introduction

- Then, we introduce some tools to calculate the digital network transfer function as well as to analyze its internal behavior.
- We also discuss some properties of generic digital filter structures associated with practical discrete-time systems.
- The chapter also introduces a number of useful building blocks often utilized in practical applications.
- A Do-it-yourself section is included in order to enlighten the reader on how to start from the concepts and generate some possible realizations for a given transfer function.

Basic structures of nonrecursive digital filters

- Nonrecursive filters are characterized by a difference equation in the form

$$y(n) = \sum_{l=0}^{M} b_l x(n-l) \qquad (1)$$

where the b_l coefficients are directly related to the system impulse response, that is, $b_l = h(l)$.

- Due to the finite length of their impulse responses, nonrecursive filters are also referred to as finite-duration impulse-response (FIR) filters.

- We can rewrite equation (1) as

$$y(n) = \sum_{l=0}^{M} h(l) x(n-l) \qquad (2)$$

Basic structures of nonrecursive digital filters

- Applying the z transform to equation (2), we end up with the input-output relationship

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{l=0}^{M} b_l z^{-l} = \sum_{l=0}^{M} h(l) z^{-l} \tag{3}$$

- In practical terms, equation (3) can be implemented in several distinct forms, using as basic elements the delay, the multiplier, and the adder blocks.
- These basic elements of digital filters and their corresponding standard symbols are depicted in Figure 1.
- An alternative way of representing such elements is the so-called signal flowgraph shown in Figure 2. These two sets of symbolisms representing the delay, multiplier, and adder elements, are used throughout this book interchangeably.

Basic structures of nonrecursive digital filters

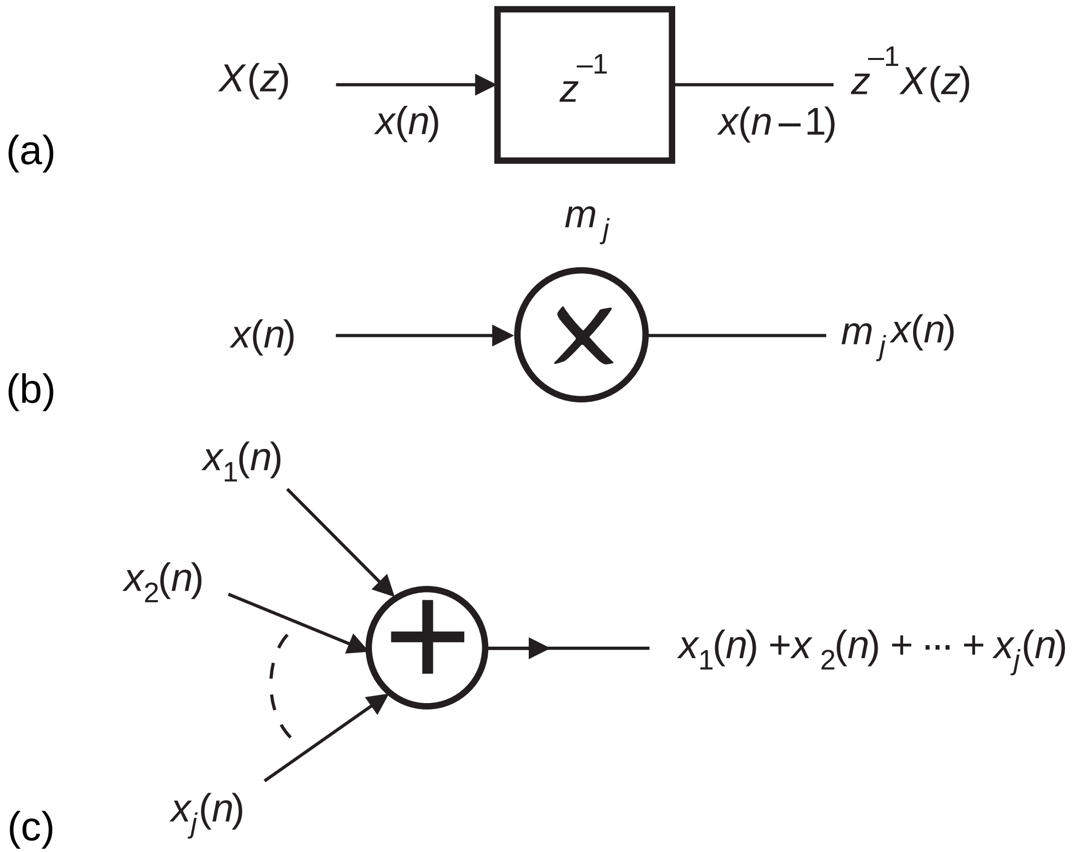


Figure 1: Classic representation of basic elements of digital filters: (a) delay; (b) multiplier; (c) adder.

Basic structures of nonrecursive digital filters

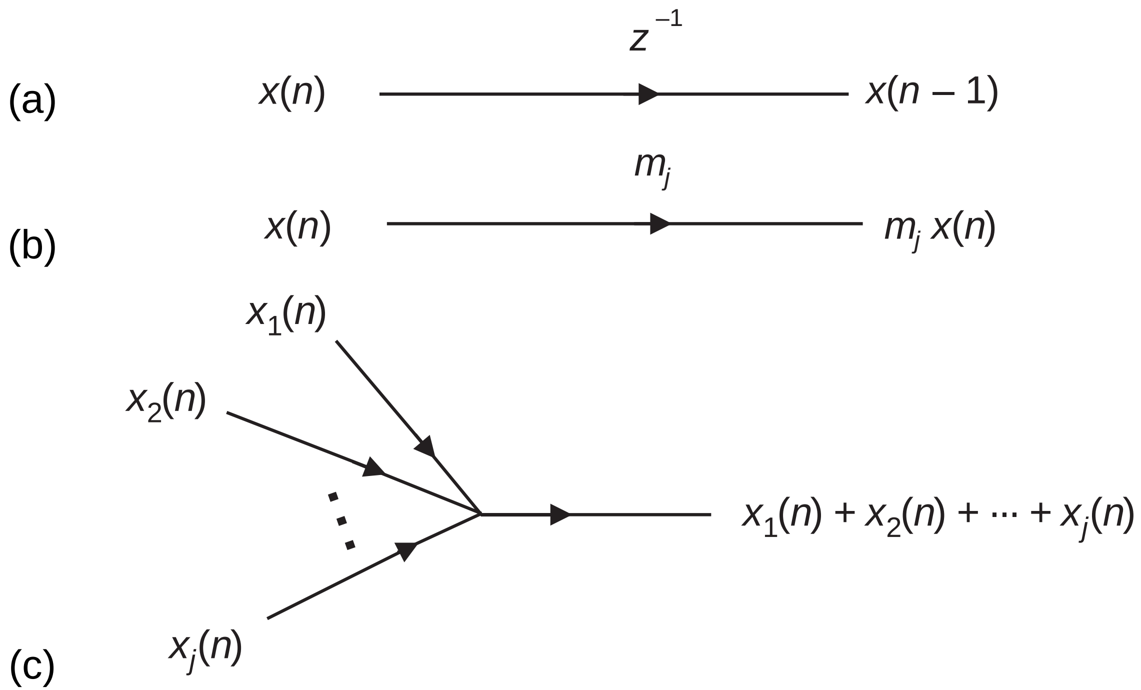


Figure 2: Signal-flowgraph representation of basic elements of digital filters: (a) delay; (b) multiplier; (c) adder.

Direct form

- The simplest realization of an FIR digital filter is derived from equation (3). The resulting structure, seen in Figure 3, is called the direct-form realization, as the multiplier coefficients are obtained directly from the filter transfer function.
- Such a structure is also referred to as the canonic direct form, where for a canonic form we understand any structure that realizes a given transfer function with the minimum number of delays, multipliers, and adders.
- More specifically, a structure that utilizes the minimum number of delays is said to be canonic with respect to the delay element, and so on.

Direct form

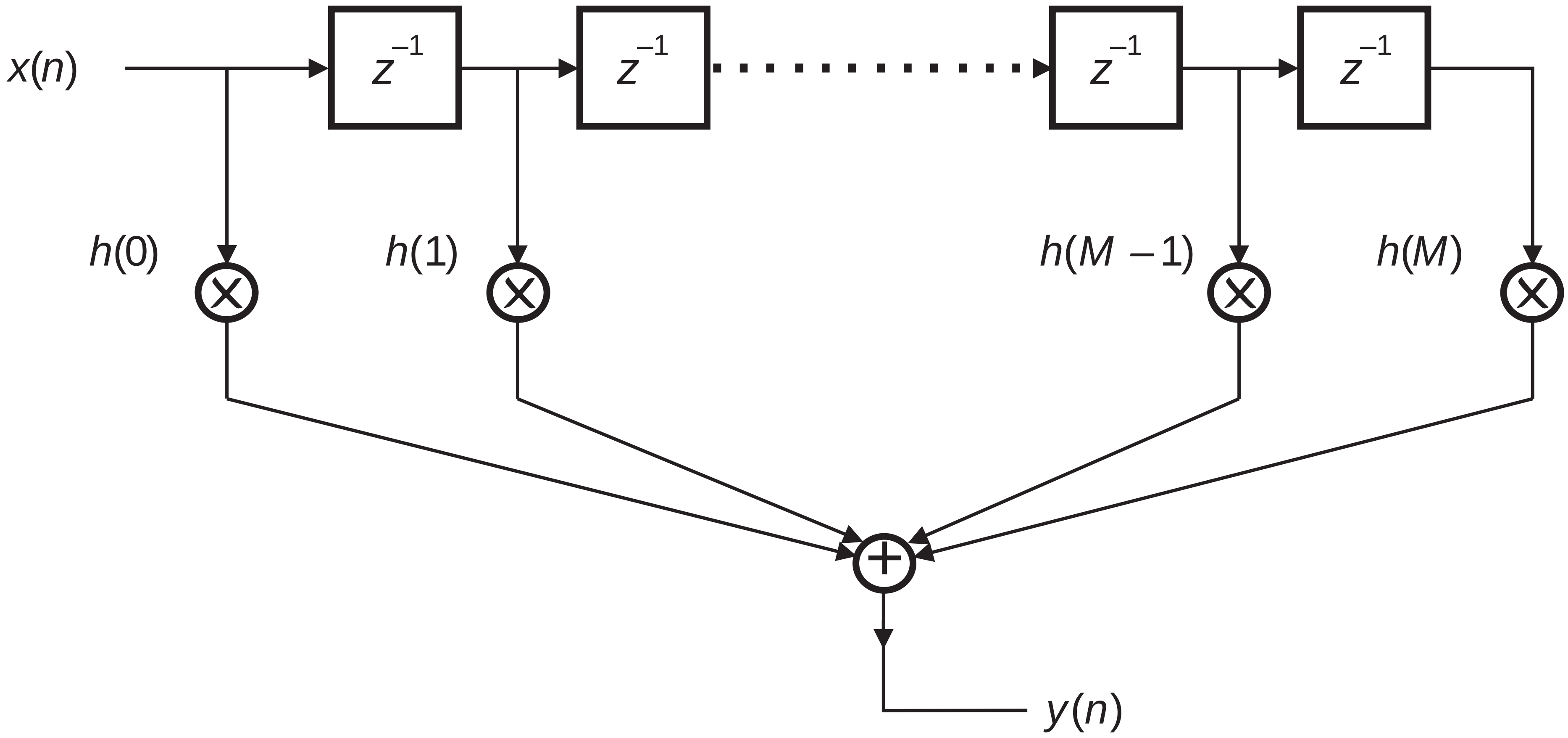


Figure 3: Direct form for FIR digital filters.

Direct form

- An alternative canonic direct form for equation (3) can be derived by expressing $H(z)$ as

$$\begin{aligned} H(z) &= \sum_{l=0}^{M} h(l) z^{-l} \\ &= \Big(h(0)+z^{-1}\Big(h(1)+z^{-1}\Big(h(2)+\cdots+z^{-1}\Big(h(M-1)+z^{-1}h(M)\Big)\cdots\Big)\Big)\Big) \end{aligned} \tag{4}$$

- The implementation of this form is shown in Figure 4.

Direct form

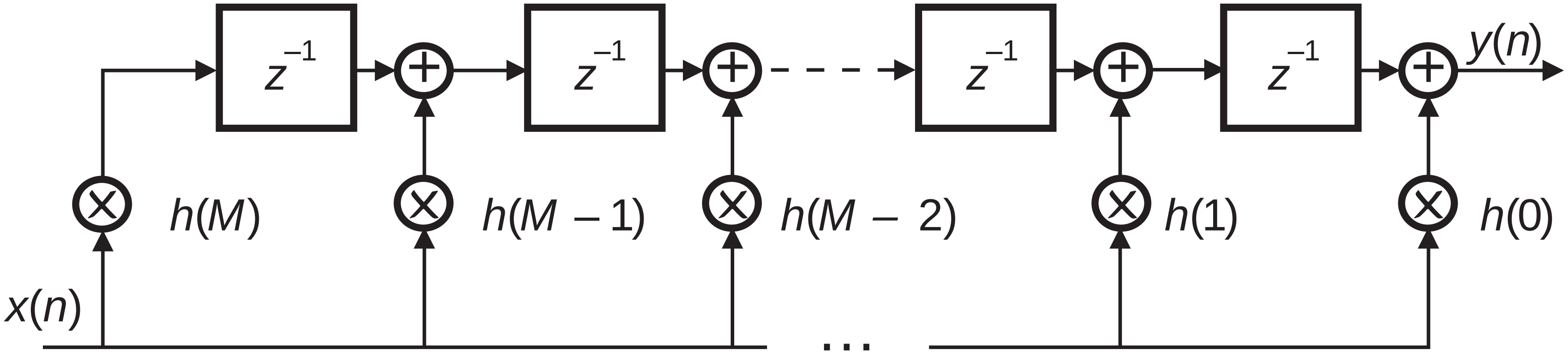


Figure 4: Alternative direct form for FIR digital filters.

Cascade form

- Equation (3) can be realized through a series of equivalent structures. However, the coefficients of such distinct realizations may not be explicitly the filter impulse response or the corresponding transfer function.
- An important example of such a realization is the so-called cascade form which consists of a series of second-order FIR filters connected in cascade, thus the name of the resulting structure, as seen in Figure 5.
- The transfer function associated with such a realization is of the form

$$H(z) = \prod_{k=1}^{N} (\gamma_{0k} + \gamma_{1k} z^{-1} + \gamma_{2k} z^{-2}) \tag{5}$$

where if M is the filter order, then $N = M/2$ when M is even, and $N = (M+1)/2$ when M is odd. In the latter case, one of the γ_{2k} becomes zero.

Cascade form

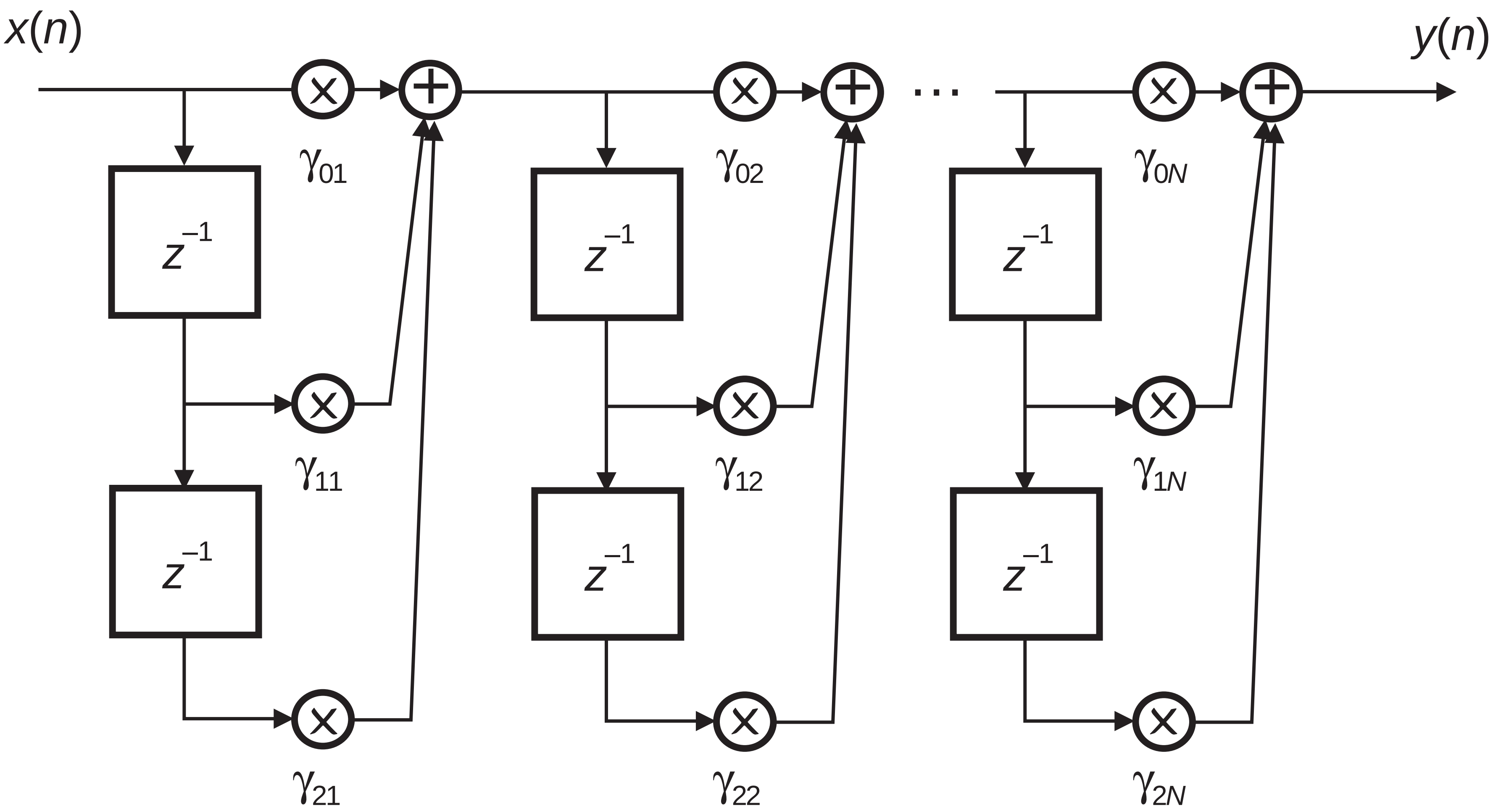


Figure 5: Cascade form for FIR digital filters.

Linear-phase forms

- An important subclass of FIR digital filters is the one that includes linear-phase filters. Such filters are characterized by a constant group delay τ, and therefore they must present a frequency response of the following form:

$$H(e^{j\omega}) = B(\omega)e^{-j\omega\tau + j\phi} \tag{6}$$

where $B(\omega)$ is real, and τ and ϕ are constant.

- Hence, the impulse response $h(n)$ of linear-phase filters satisfies

$$\begin{aligned} h(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} B(\omega)e^{-j\omega\tau + j\phi}e^{j\omega n}d\omega \\ &= \frac{e^{j\phi}}{2\pi}\int_{-\pi}^{\pi} B(\omega)e^{j\omega(n-\tau)}d\omega \end{aligned} \tag{7}$$

Linear-phase forms

- We are considering here filters where the group delay is a multiple of half a sample, that is,

$$\tau = \frac{k}{2}, \quad k \in \mathbb{Z} \tag{8}$$

- Thus, for such cases when 2τ is an integer, equation (8) implies that

$$h(2\tau - n) = \frac{e^{j\phi}}{2\pi}\int_{-\pi}^{\pi} B(\omega)e^{j\omega(2\tau-n-\tau)}d\omega = \frac{e^{j\phi}}{2\pi}\int_{-\pi}^{\pi} B(\omega)e^{j\omega(\tau-n)}d\omega \tag{9}$$

- Since $B(\omega)$ is real,

$$\begin{aligned} h^*(2\tau - n) &= \frac{e^{-j\phi}}{2\pi}\int_{-\pi}^{\pi} B^*(\omega)e^{-j\omega(\tau-n)}d\omega \\ &= \frac{e^{-j\phi}}{2\pi}\int_{-\pi}^{\pi} B(\omega)e^{j\omega(n-\tau)}d\omega \end{aligned} \tag{10}$$

Linear-phase forms

- Then, from equations (7) and (10), in order for a filter to have linear phase with a constant group delay τ, its impulse response must satisfy

$$h(n) = e^{2j\phi} h^*(2\tau - n) \tag{11}$$

- We now proceed to show that linear-phase FIR filters present impulse responses of very particular forms. In fact, equation (11) implies that $h(0) = e^{2j\phi} h^*(2\tau)$.

- Hence, if $h(n)$ is causal and of finite duration, for $0 \leq n \leq M$, we must necessarily have that

$$\tau = \frac{M}{2} \tag{12}$$

and then, equation (11) becomes

$$h(n) = e^{2j\phi} h^*(M - n) \tag{13}$$

which is the general property for the coefficients of a linear-phase FIR filter.

Linear-phase forms

- In the common case where all the filter coefficients are real, then $h(n) = h^*(n)$, and equation (13) implies that $e^{2j\phi}$ must be real. Thus

$$\phi = \frac{k\pi}{2},\ k \in \mathbb{Z} \tag{14}$$

and equation (13) becomes

$$h(n) = (-1)^k h(M-n),\ k \in \mathbb{Z} \tag{15}$$

That is, the filter impulse response must be either symmetric or antisymmetric.

- From equation (6), the frequency response of linear-phase FIR filters with real coefficients becomes

$$H(e^{j\omega}) = B(\omega)e^{-j\omega\frac{M}{2}+j\frac{k\pi}{2}} \tag{16}$$

Linear-phase forms

- For all practical purposes, we only need to consider the cases when $k = 0, 1, 2, 3$, as all other values of k will be equivalent to one of these four cases. Furthermore, as $B(\omega)$ can be either positive or negative, the cases $k = 2$ and $k = 3$ are obtained from cases $k = 0$ and $k = 1$ respectively, by making $B(\omega) \leftarrow -B(\omega)$.

- Therefore, we consider solely the four distinct cases described by equations (13) and (16). They are referred to as follows:
  - Type I: $k = 0$ and M even.
  - Type II: $k = 0$ and M odd.
  - Type III: $k = 1$ and M even.
  - Type IV: $k = 1$ and M odd.

- We now proceed to demonstrate that $h(n) = (-1)^k h(M - n)$ is a sufficient condition for an FIR filter with real coefficients to have a linear phase. The four types above are considered separately.

Linear-phase form - Type I

- Type I: $k = 0$ implies that the filter has symmetric impulse response, that is, $h(M-n) = h(n)$.
- Since the filter order M is even, equation (3) may be rewritten as

$$
\begin{aligned}
H(z) &= \sum_{n=0}^{\frac{M}{2}-1} h(n)z^{-n} + h\left(\frac{M}{2}\right) z^{-\frac{M}{2}} + \sum_{n=\frac{M}{2}+1}^{M} h(n)z^{-n} \\
&= \sum_{n=0}^{\frac{M}{2}-1} h(n)\left[z^{-n} + z^{-(M-n)}\right] + h\left(\frac{M}{2}\right) z^{-\frac{M}{2}} \qquad (17)
\end{aligned}
$$

Linear-phase form - Type I

- Evaluating this equation over the unit circle, that is, using the variable transformation $z \longrightarrow e^{j\omega}$, one obtains

$$
\begin{aligned}
H(e^{j\omega}) &= \sum_{n=0}^{\frac{M}{2}-1} h(n)\left[e^{-j\omega n} + e^{(-j\omega M + j\omega n)}\right] + h\left(\frac{M}{2}\right)e^{-j\omega\frac{M}{2}} \\
&= e^{-j\omega\frac{M}{2}}\left\{h\left(\frac{M}{2}\right) + \sum_{n=0}^{\frac{M}{2}-1} 2h(n)\cos\left[\omega\left(n-\frac{M}{2}\right)\right]\right\} \quad (18)
\end{aligned}
$$

Linear-phase form - Type I

- Substituting n by $(\frac{M}{2} - m)$, we get

$$H(e^{j\omega}) = e^{-j\omega\frac{M}{2}} \left[h\left(\frac{M}{2}\right) + \sum_{m=1}^{\frac{M}{2}} 2h\left(\frac{M}{2} - m\right) \cos(\omega m) \right]$$

$$= e^{-j\omega\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}} a(m) \cos(\omega m) \qquad (19)$$

with $a(0) = h(\frac{M}{2})$ and $a(m) = 2h(\frac{M}{2} - m)$, for $m = 1, 2, \ldots, \frac{M}{2}$.

- Since this equation is in the form of equation (16), this completes the sufficiency proof for Type-I filters.

Linear-phase form - Type II

- Type II: $k = 0$ implies that the filter has a symmetric impulse response, that is, $h(M - n) = h(n)$.

- Since the filter order M is odd, equation (3) may be rewritten as

$$
\begin{aligned}
H(z) &= \sum_{n=0}^{\frac{M-1}{2}} h(n)z^{-n} + \sum_{n=\frac{M+1}{2}}^{M} h(n)z^{-n} \\
&= \sum_{n=0}^{\frac{M-1}{2}} h(n)[z^{-n} + z^{-(M-n)}] \qquad (20)
\end{aligned}
$$

Linear-phase form - Type II

- Evaluating this equation over the unit circle, one obtains

$$
\begin{aligned}
H(e^{j\omega}) &= \sum_{n=0}^{\frac{M-1}{2}} h(n)\left[e^{-j\omega n} + e^{(-j\omega M + j\omega n)}\right] \\
&= e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} h(n)\left[e^{-j\omega(n-\frac{M}{2})} + e^{j\omega(n-\frac{M}{2})}\right] \\
&= e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} 2h(n)\cos\left[\omega\left(n-\frac{M}{2}\right)\right] \qquad (21)
\end{aligned}
$$

Linear-phase form - Type II

- Substituting n with $(\frac{M+1}{2} - m)$

$$H(e^{j\omega}) = e^{-j\omega\frac{M}{2}} \sum_{m=1}^{\frac{M+1}{2}} 2h\left(\frac{M+1}{2} - m\right) \cos\left[\omega\left(m - \frac{1}{2}\right)\right]$$
$$= e^{-j\omega\frac{M}{2}} \sum_{m=1}^{\frac{M+1}{2}} b(m) \cos\left[\omega\left(m - \frac{1}{2}\right)\right] \quad (22)$$

  with $b(m) = 2h(\frac{M+1}{2} - m)$, for $m = 1, 2, \ldots, \frac{M+1}{2}$.

- Since this equation is in the form of equation (16), this completes the sufficiency proof for Type-II filters.

- Notice that at $\omega = \pi$, $H(e^{j\omega}) = 0$, as it consists of a summation of cosine functions evaluated at $\pm\frac{\pi}{2}$, which are obviously null. Therefore, highpass and bandstop filters can not be approximated as Type-II filters.

Linear-phase form - Type III

- Type III: $k = 1$ implies that the filter has an antisymmetric impulse response, that is, $h(M - n) = -h(n)$.
- In this case, $h(\frac{M}{2})$ is necessarily null. Since the filter order M is even, equation (3) may be rewritten as

$$\begin{aligned} H(z) &= \sum_{n=0}^{\frac{M}{2}-1} h(n) z^{-n} + \sum_{n=\frac{M}{2}+1}^{M} h(n) z^{-n} \\ &= \sum_{n=0}^{\frac{M}{2}-1} h(n) \left[z^{-n} - z^{-(M-n)} \right] \end{aligned} \tag{23}$$

Linear-phase form - Type III

- Which, when evaluated over the unit circle, yields

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^{\frac{M}{2}-1} h(n)\left[e^{-j\omega n} - e^{(-j\omega M + j\omega n)}\right] \\ &= e^{-j\omega \frac{M}{2}} \sum_{n=0}^{\frac{M}{2}-1} h(n)\left[e^{-j\omega(n-\frac{M}{2})} - e^{j\omega(n-\frac{M}{2})}\right] \\ &= e^{-j\omega \frac{M}{2}} \sum_{n=0}^{\frac{M}{2}-1} -2jh(n)\sin\left[\omega\left(n-\frac{M}{2}\right)\right] \\ &= e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})} \sum_{n=0}^{\frac{M}{2}-1} -2h(n)\sin\left[\omega\left(n-\frac{M}{2}\right)\right] \end{aligned} \tag{24}$$

Linear-phase form - Type III

- Substituting n by $(\frac{M}{2} - m)$

$$H(e^{j\omega}) = e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})} \sum_{m=1}^{\frac{M}{2}} -2h\left(\frac{M}{2} - m\right) \sin[\omega(-m)]$$
$$= e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})} \sum_{m=1}^{\frac{M}{2}} c(m) \sin(\omega m) \qquad (25)$$

  with $c(m) = 2h(\frac{M}{2} - m)$, for $m = 1, 2, \ldots, \frac{M}{2}$.

- Since this equation is in the form of equation (16), this completes the sufficiency proof for Type-III filters.

- Notice, in this case, that the frequency response becomes null at $\omega = 0$ and at $\omega = \pi$. That makes this type of realization suitable for bandpass filters, differentiators, and Hilbert transformers, these last two due to the phase shift of $\frac{\pi}{2}$.

Linear-phase form - Type IV

- Type IV: $k = 1$ implies that the filter has an antisymmetric impulse response, that is, $h(M - n) = -h(n)$.
- Since the filter order M is odd, equation (3) may be rewritten as

$$
\begin{aligned}
H(z) &= \sum_{n=0}^{\frac{M-1}{2}} h(n) z^{-n} + \sum_{n=\frac{M+1}{2}}^{M} h(n) z^{-n} \\
&= \sum_{n=0}^{\frac{M-1}{2}} h(n) \left[z^{-n} - z^{-(M-n)} \right]
\end{aligned}
\qquad (26)
$$

Linear-phase form - Type IV

- Evaluating this equation over the unit circle

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^{\frac{M-1}{2}} h(n)\left[e^{-j\omega n} - e^{(-j\omega M + j\omega n)}\right] \\ &= e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} h(n)\left[e^{-j\omega(n-\frac{M}{2})} - e^{j\omega(n-\frac{M}{2})}\right] \\ &= e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} -2jh(n)\sin\left[\omega\left(n-\frac{M}{2}\right)\right] \\ &= e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})} \sum_{n=0}^{\frac{M-1}{2}} -2h(n)\sin\left[\omega\left(n-\frac{M}{2}\right)\right] \end{aligned} \tag{27}$$

Linear-phase form - Type IV

- Substituting n by $\left(\frac{M+1}{2} - m\right)$

$$
\begin{aligned}
H(e^{j\omega}) &= e^{-j(\omega\frac{M}{2} - \frac{\pi}{2})} \sum_{m=1}^{\frac{M+1}{2}} -2h\left(\frac{M+1}{2} - m\right) \sin\left[\omega\left(\frac{1}{2} - m\right)\right] \\
&= e^{-j(\omega\frac{M}{2} - \frac{\pi}{2})} \sum_{m=1}^{\frac{M+1}{2}} d(m) \sin\left[\omega\left(m - \frac{1}{2}\right)\right] \qquad (28)
\end{aligned}
$$

with $d(m) = 2h(\frac{M+1}{2} - m)$, for $m = 1, 2, \ldots, \frac{M+1}{2}$.

- Since this equation is in the form of equation (16), this completes the sufficiency proof for Type-IV filters, thus finishing the whole proof.

- Notice that $H(e^{j\omega}) = 0$, at $\omega = 0$. Hence, lowpass filters can not be approximated as Type-IV filters, although they are still suitable for differentiators and Hilbert transformers, like filters having the Type-III form.

Linear-phase forms

- Typical impulse responses of the four cases of linear-phase FIR digital filters are depicted in Figure 6.
- The properties of all four cases are summarized in Table 1.

Table 1: Characteristics of linear-phase FIR filters: order, impulse response, frequency response, phase response, and group delay.

Type	M	$h(n)$	$H(e^{j\omega})$	$\Theta(\omega)$	τ
I	Even	Symmetric	$e^{-j\omega\frac{M}{2}}\sum_{m=0}^{\frac{M}{2}} a(m)\cos(\omega m)$ $a(0)=h\left(\frac{M}{2}\right); a(m)=2h\left(\frac{M}{2}-m\right)$	$-\omega\frac{M}{2}$	$\frac{M}{2}$
II	Odd	Symmetric	$e^{-j\omega\frac{M}{2}}\sum_{m=1}^{\frac{M+1}{2}} b(m)\cos\left[\omega\left(m-\frac{1}{2}\right)\right]$ $b(m)=2h\left(\frac{M+1}{2}-m\right)$	$-\omega\frac{M}{2}$	$\frac{M}{2}$
III	Even	Antisymmetric	$e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})}\sum_{m=1}^{\frac{M}{2}} c(m)\sin(\omega m)$ $c(m)=2h\left(\frac{M}{2}-m\right)$	$-\omega\frac{M}{2}+\frac{\pi}{2}$	$\frac{M}{2}$
IV	Odd	Antisymmetric	$e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})}\sum_{m=1}^{\frac{M+1}{2}} d(m)\sin\left[\omega\left(m-\frac{1}{2}\right)\right]$ $d(m)=2h\left(\frac{M+1}{2}-m\right)$	$-\omega\frac{M}{2}+\frac{\pi}{2}$	$\frac{M}{2}$

Linear-phase forms

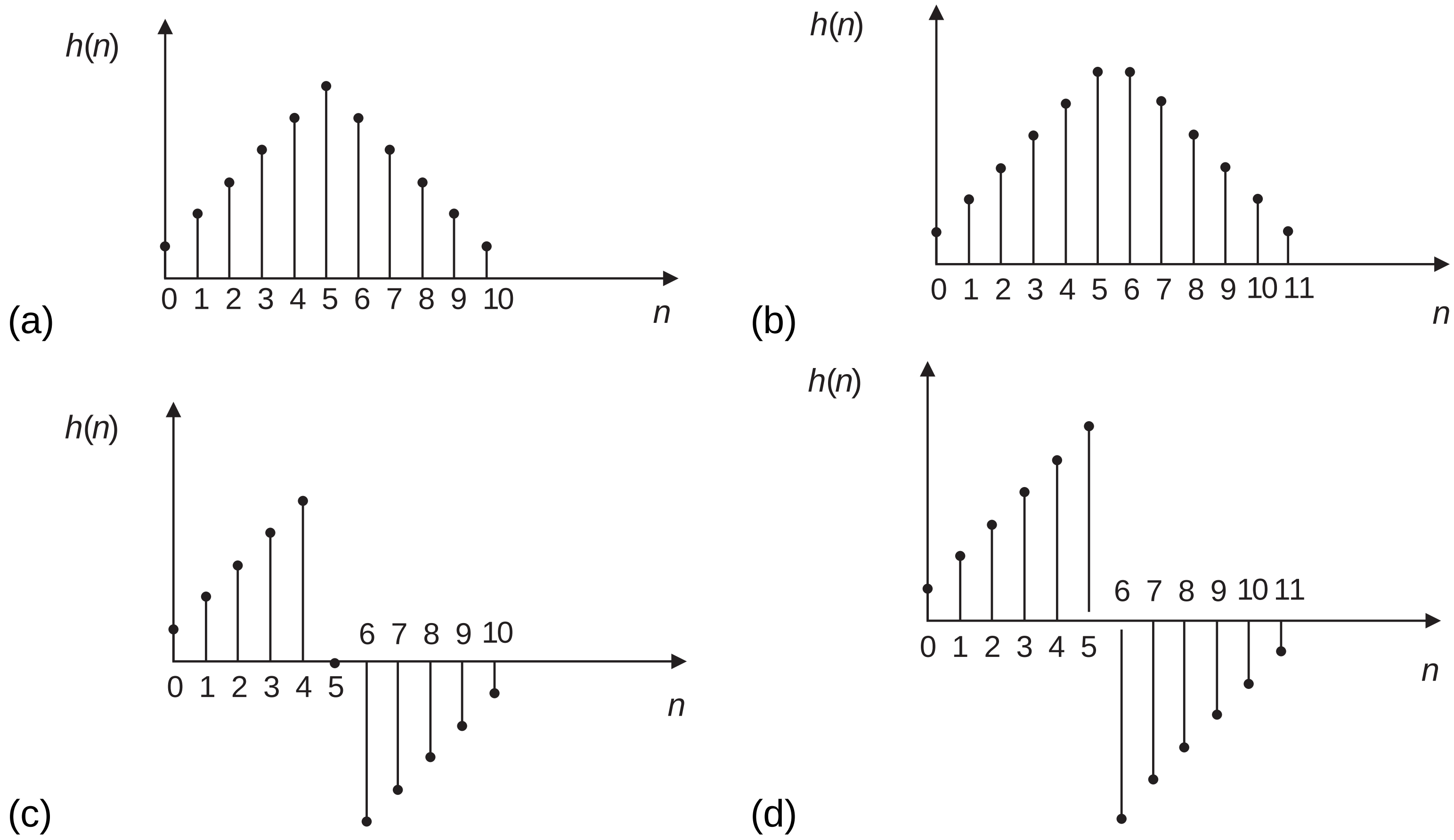


Figure 6: Example of impulse responses of linear-phase FIR digital filters: (a) Type I; (b) Type II; (c) Type III; (d) Type IV.

Linear-phase forms

- One can derive important properties of linear-phase FIR filters by representing equations (17), (20), (23), and (26) in a single framework as

$$H(z) = z^{-\frac{M}{2}} \sum_{n=0}^{K} h(n) \left(z^{\frac{M}{2}-n} \pm z^{-\left(\frac{M}{2}-n\right)} \right) \tag{29}$$

where $K = \frac{M}{2}$, if M is even; or $K = \frac{M-1}{2}$, if M is odd.

- From equation (29), it is easy to observe that, if z_γ is a zero of $H(z)$, so is z_γ^{-1}. This implies that all zeros of $H(z)$ occur in reciprocal pairs.

Linear-phase forms

- Considering that if the coefficients $h(n)$ are real, all complex zeros occur in conjugate pairs, and then one can infer that the zeros of $H(z)$ must satisfy:
  - All complex zeros which are not on the unit circle occur in conjugate and reciprocal quadruples. In other words, if z_γ is complex, then z_γ^{-1}, z_γ^*, and $(z_\gamma^{-1})^*$ are also zeros of $H(z)$.
  - There can be any given number of zeros over the unit circle, in conjugate pairs, since in this case we automatically have that $z_\gamma^{-1} = z_\gamma^*$.
  - All real zeros outside the unit circle occur in reciprocal pairs.
  - There can be any given number of zeros at $z = z_\gamma = \pm 1$, since in this case we necessarily have that $z_\gamma^{-1} = \pm 1$.
- A typical zero plot for a linear-phase lowpass FIR filter is shown in Figure 7.

Linear-phase forms

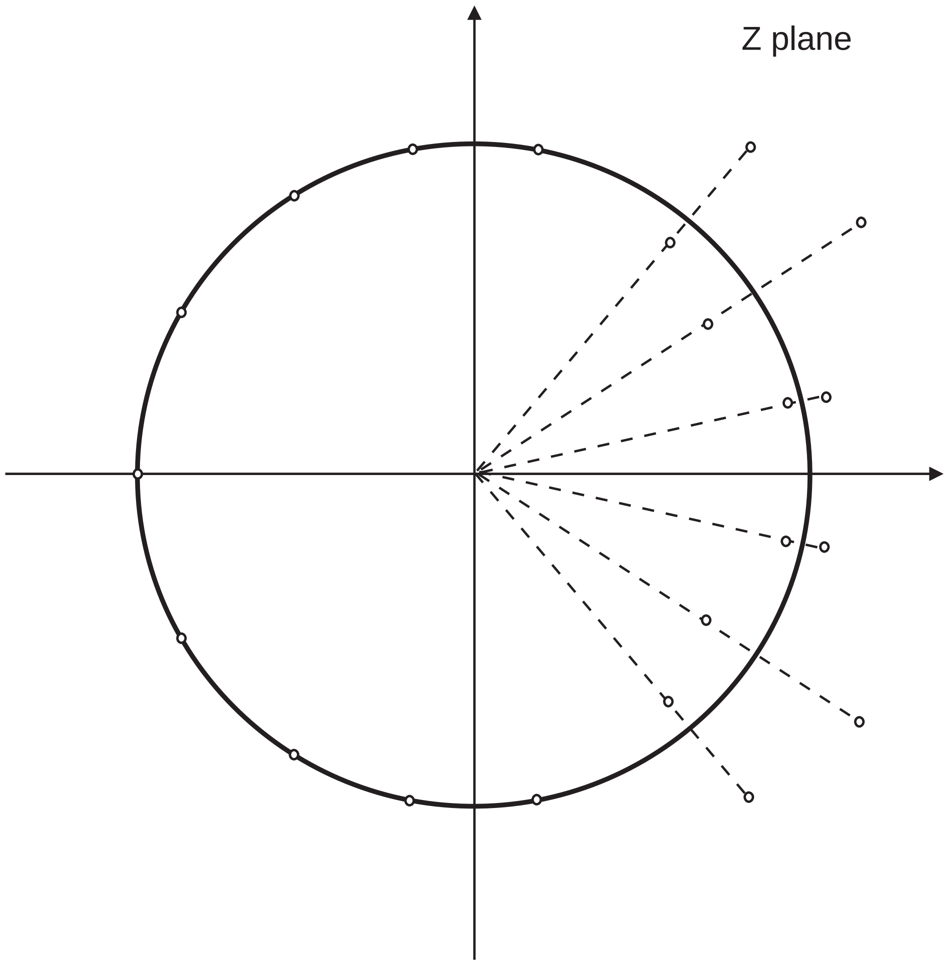


Figure 7: Typical zero plot of a linear-phase FIR digital filter.

Linear-phase forms

- An interesting property of linear-phase FIR digital filters is that they can be realized with efficient structures that exploit their symmetric or antisymmetric impulse-response characteristics.
- In fact, when M is even, these efficient structures require $\frac{M}{2} + 1$ multiplications, while when M is odd, only $\frac{M+1}{2}$ multiplications are necessary.
- Figure 8 depicts two of these efficient structures for linear-phase FIR filters when the impulse response is symmetric.

Linear-phase forms

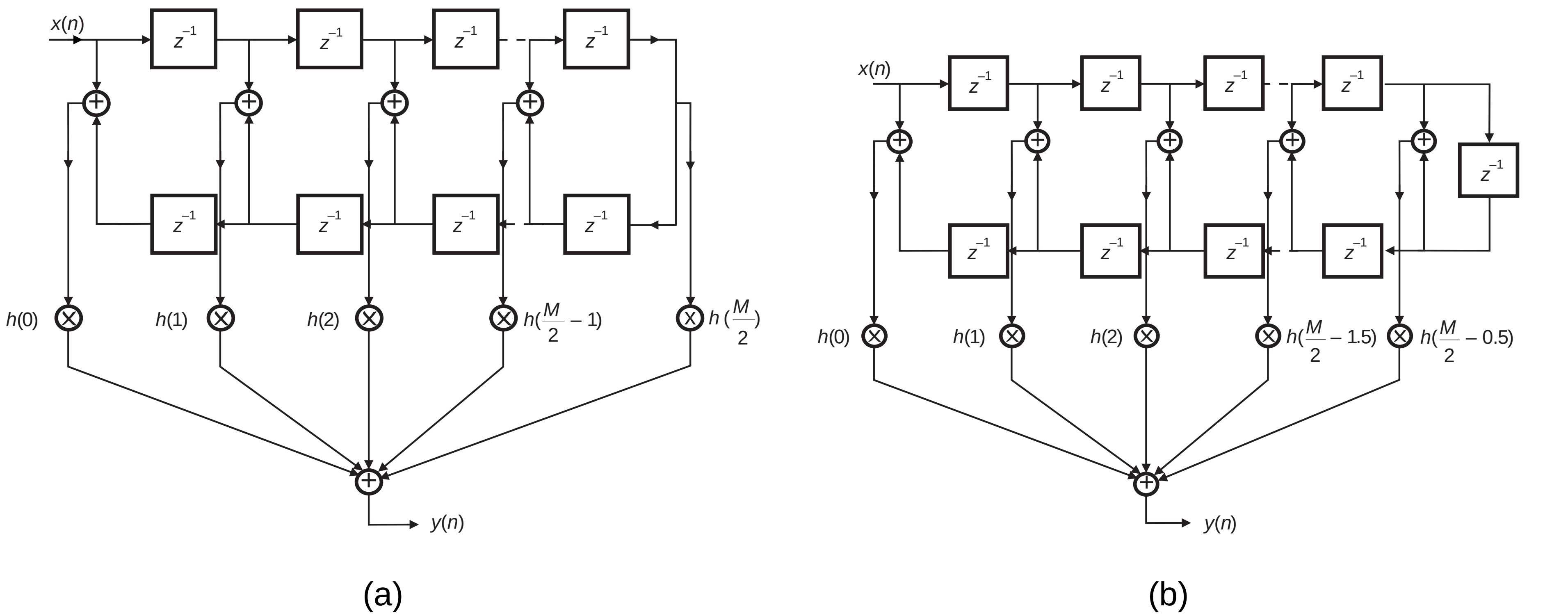


Figure 8: Realizations of linear-phase filters with symmetric impulse response: (a) even order; (b) odd order.

Basic structures of recursive digital filters

- The transfer function of a recursive filter is given by

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{i=1}^{N} a_i z^{-i}} \tag{30}$$

- Since, in most cases, such transfer functions give rise to filters with impulse responses having infinite durations, recursive filters are also referred to as infinite-duration impulse-response (IIR) filters.

- It is important to notice, however, that in the cases where $D(z)$ divides $N(z)$, the filter $H(z)$ turns out to have a finite-duration impulse response, and is actually an FIR filter.

Basic structures of recursive digital filters

- We can consider that $H(z)$ as above results from the cascading of two separate filters of transfer functions $N(z)$ and $\frac{1}{D(z)}$.
- The $N(z)$ polynomial can be realized with the FIR direct form, as shown in the previous section.
- The realization of $\frac{1}{D(z)}$ can be performed as depicted in Figure 9, where the FIR filter shown will be an $(N-1)$th-order filter with transfer function

$$D'(z) = z(1 - D(z)) = -z \sum_{i=1}^{N} a_i z^{-i} \tag{31}$$

which can be realized as in Figure 3. The direct form for realizing $\frac{1}{D(z)}$ is then shown in Figure 10.

Direct form

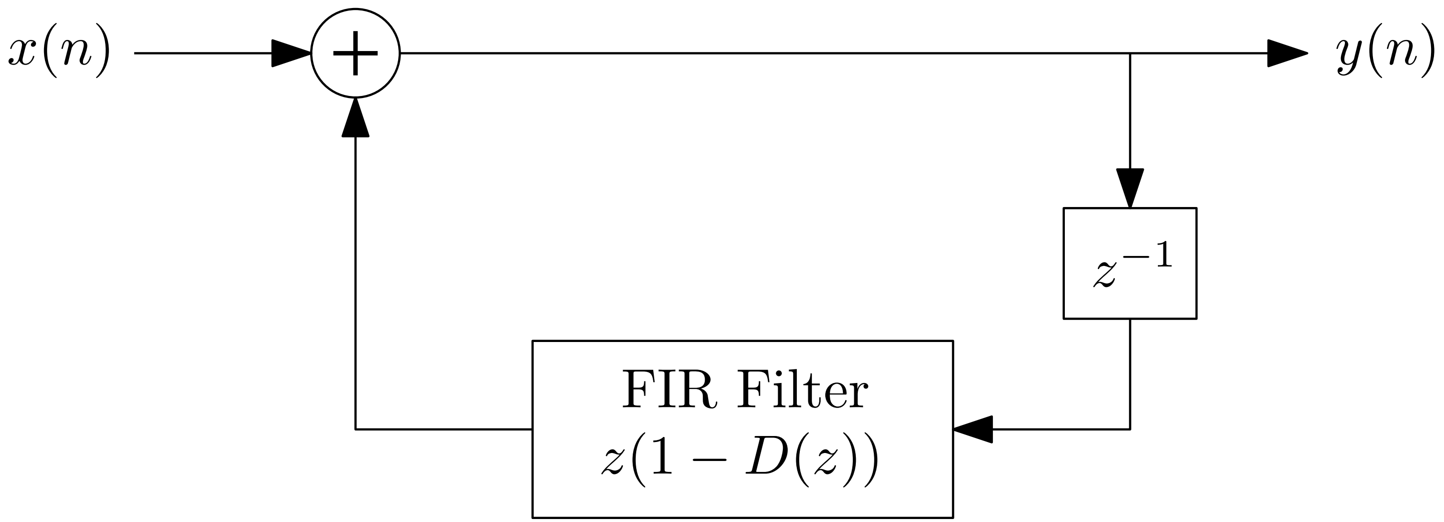


Figure 9: Block diagram realization of $\frac{1}{D(z)}$.

Direct form

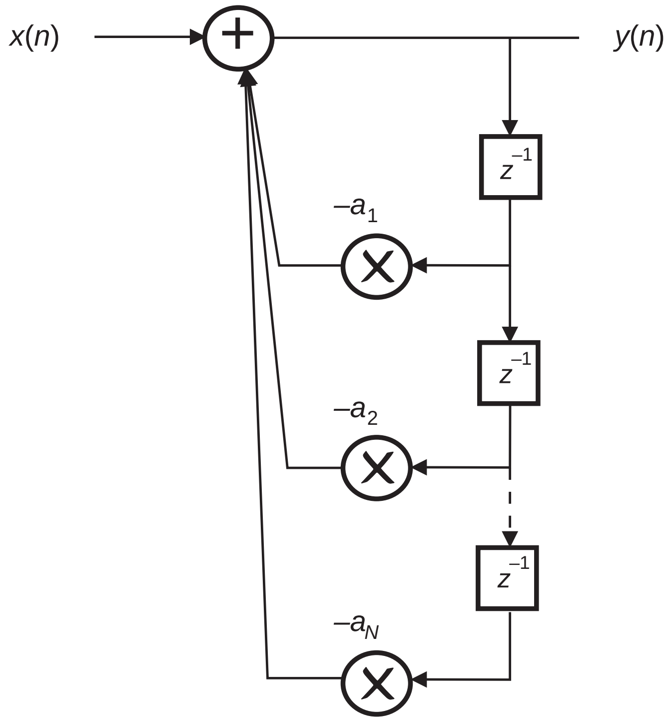


Figure 10: Detailed realization of $\frac{1}{D(z)}$.

Direct form

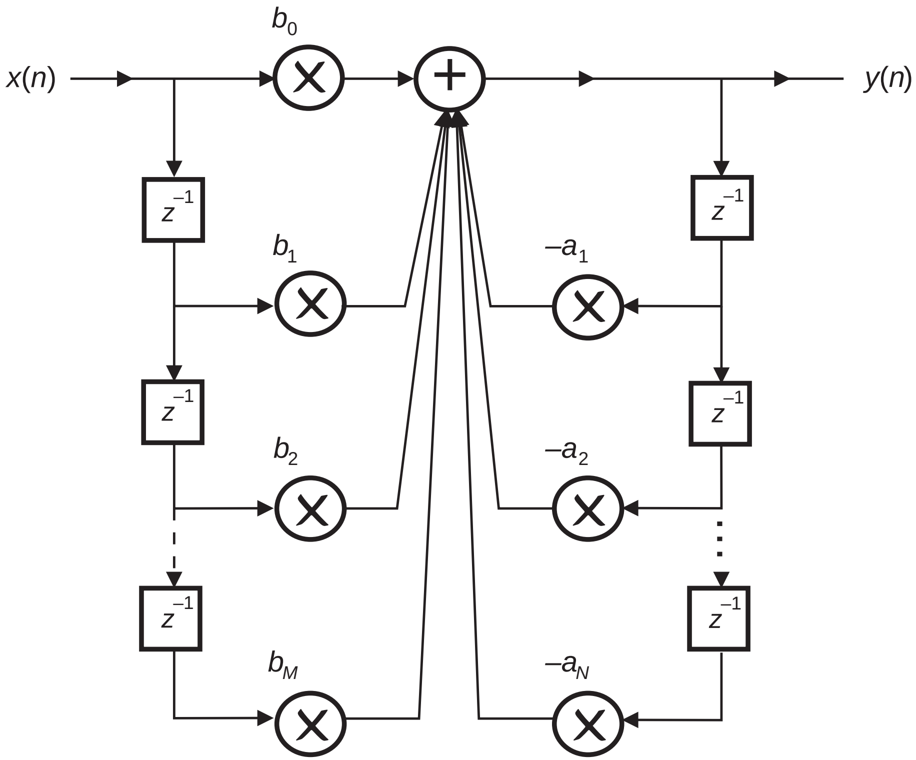


Figure 11: Noncanonic IIR direct-form realization.

Direct form

- The complete realization of $H(z)$, as a cascade of $N(z)$ and $\frac{1}{D(z)}$, is shown in Figure 11. Such a structure is not canonic with respect to the delays, since for a (M, N)th-order filter this realization requires $(N + M)$ delays.

- Clearly, in the general case we can change the order in which we cascade the two separate filters, that is, $H(z)$ can be realized as $N(z) \times \frac{1}{D(z)}$ or $\frac{1}{D(z)} \times N(z)$.

- In the second option, all delays employed start from the same node, which allows us to eliminate the consequent redundant delays. In that manner, the resulting structure, usually referred to as the Type 1 canonic direct form, is the one depicted in Figure 12, for the special case when $N = M$.

- The majority of IIR filter transfer functions used in practice present a numerator degree, M, smaller than or equal to the denominator degree, N. In general, one can consider that $M = N$. In the case where $M < N$, we just make the coefficients $b_{M+1}, b_{M+2}, \ldots, b_N$ in Figures 12 and 13 equal to zero.

Direct form

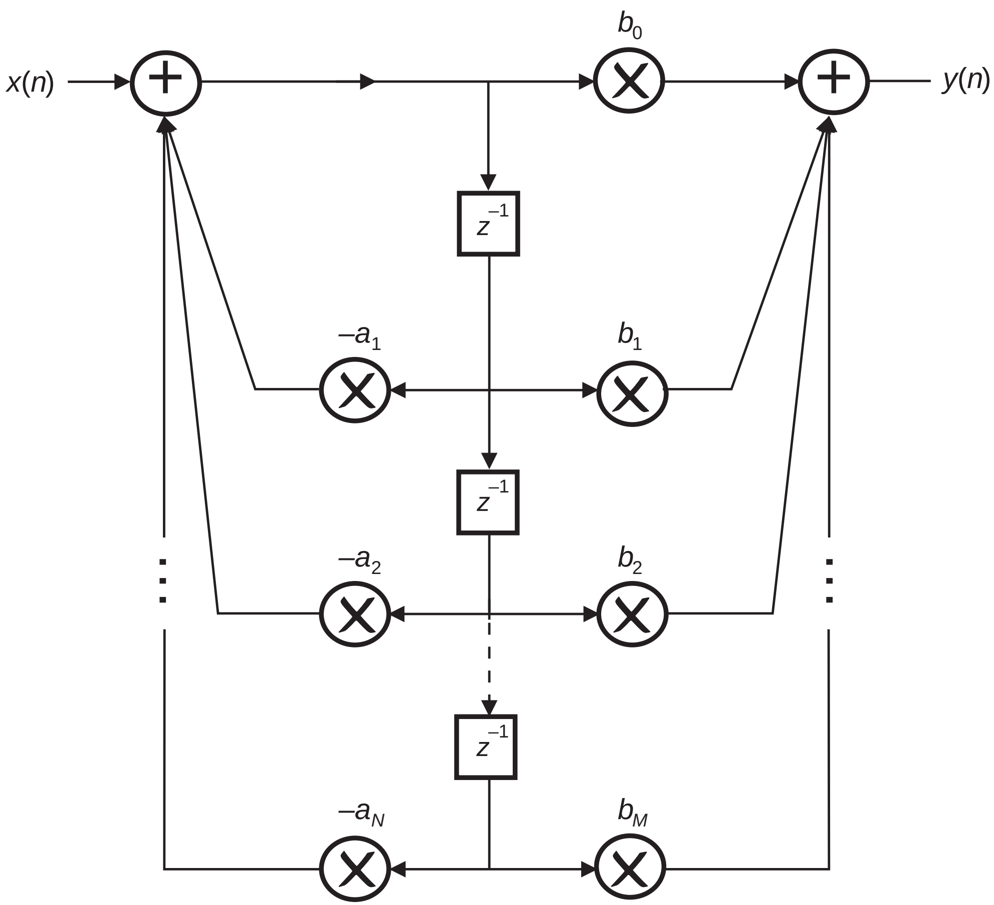


Figure 12: Type 1 canonic direct form for IIR filters.

Direct form

- An alternative structure, the so-called Type 2 canonic direct form, is shown in Figure 13. Such a realization is generated from the nonrecursive form in Figure 4.

Direct form

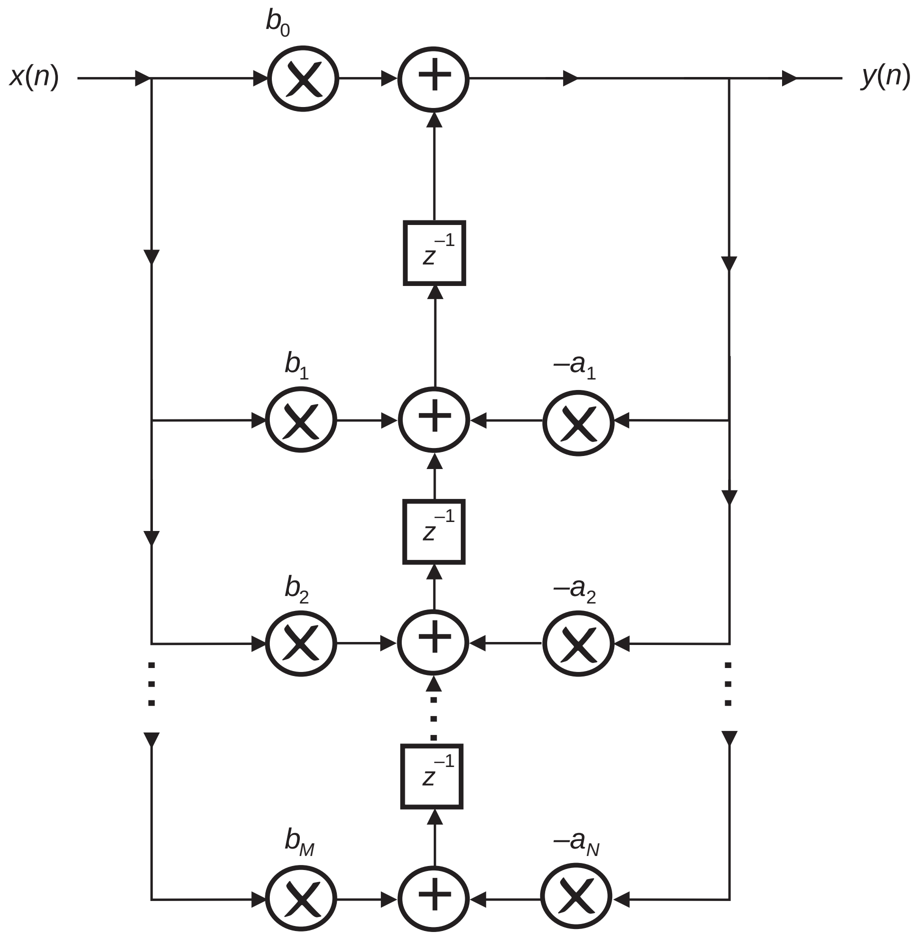


Figure 13: Type 2 canonic direct form for IIR filters.

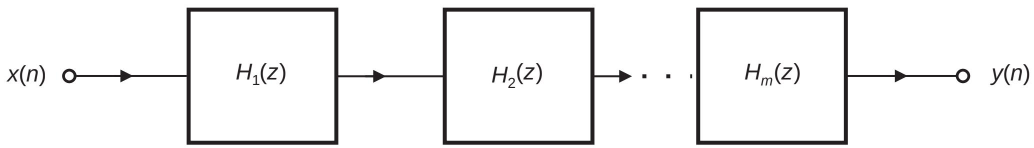
Cascade form

- In the same way as their FIR counterparts, the IIR digital filters present a large variety of possible alternative realizations. An important one, referred to as the cascade realization, is depicted in Figure 14a, where the basic blocks represent simple transfer functions of orders 2 or 1.

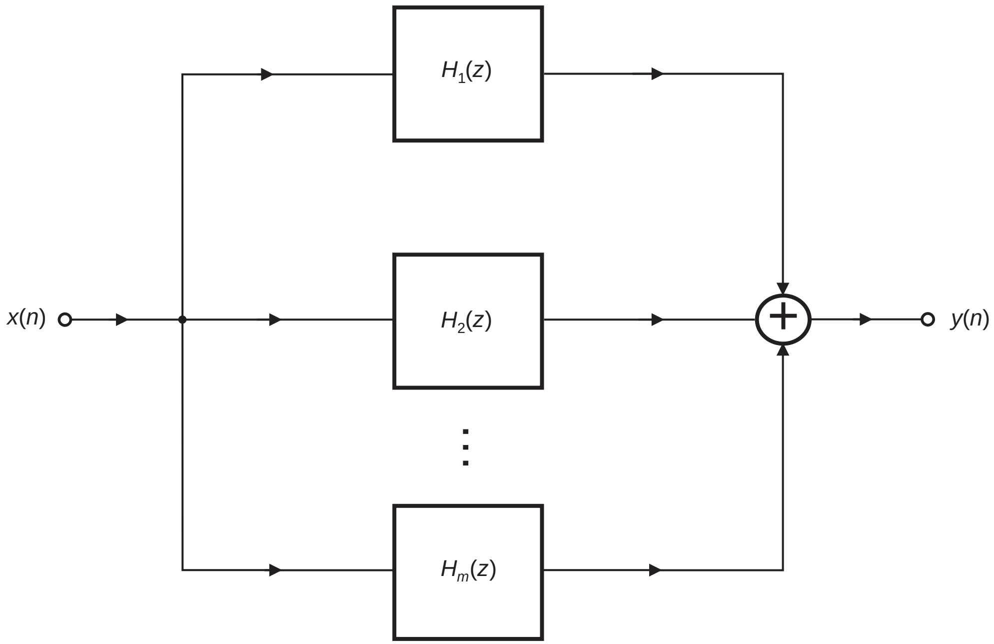
- The cascade form, based on second-order blocks, is associated to the following transfer function decomposition:

$$
\begin{aligned}
H(z) &= \prod_{k=1}^{m} \frac{\gamma_{0k} + \gamma_{1k} z^{-1} + \gamma_{2k} z^{-2}}{1 + m_{1k} z^{-1} + m_{2k} z^{-2}} \\
&= \prod_{k=1}^{m} \frac{\gamma_{0k} z^{2} + \gamma_{1k} z + \gamma_{2k}}{z^{2} + m_{1k} z + m_{2k}} \\
&= H_0 \prod_{k=1}^{m} \frac{z^{2} + \gamma'_{1k} z + \gamma'_{2k}}{z^{2} + m_{1k} z + m_{2k}}
\end{aligned} \tag{32}
$$

Cascade form



(a)



(b)

Figure 14: Block diagrams of: (a) cascade form; (b) parallel form.

Parallel form

- Another important realization for recursive digital filters is the parallel form represented in Figure 14b.

- Using second-order blocks, which are the most commonly used in practice, the parallel realization corresponds to the following transfer function decomposition:

$$
\begin{aligned}
H(z) &= \sum_{k=1}^{m} \frac{\gamma_{0k}^{p} z^2 + \gamma_{1k}^{p} z + \gamma_{2k}^{p}}{z^2 + m_{1k} z + m_{2k}} \\
&= h_0 + \sum_{k=1}^{m} \frac{\gamma_{1k}^{p'} z + \gamma_{2k}^{p'}}{z^2 + m_{1k} z + m_{2k}} \\
&= h_0' + \sum_{k=1}^{m} \frac{\gamma_{0k}^{p''} z^2 + \gamma_{1k}^{p''} z}{z^2 + m_{1k} z + m_{2k}} \qquad (33)
\end{aligned}
$$

also known as the partial-fraction decomposition.

Parallel form

- This equation indicates three alternative forms of the parallel realization, where the last two are canonic with respect to the number of multiplier elements.
- It should be mentioned that each second-order block in the cascade and parallel forms can be realized by any of the existing distinct structures, as, for instance, one of the direct forms shown in Figure 15.

Parallel form

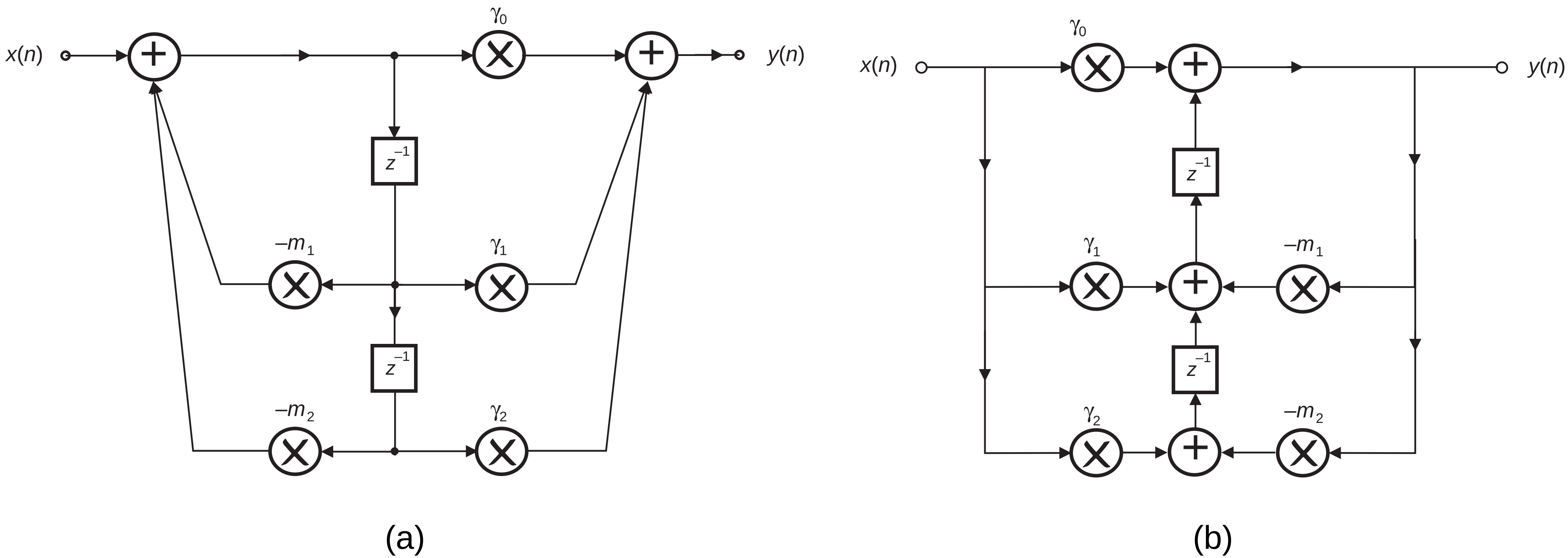


Figure 15: Realizations of second-order blocks: (a) Type 1 direct form; (b) Type 2 direct form.

Parallel form

- As it will be seen in future chapters, all these digital filter realizations present different properties when one considers practical finite-precision implementations, that is, the quantization of the coefficients and the finite precision of the arithmetic operations, such as additions and multiplications.

- In fact, the analysis of the finite-precision effects in the distinct realizations is a fundamental step in the overall process of designing any digital filter, as will be discussed in detail in Chapter 11.

Example 4.1

- Describe the digital filter implementation of the transfer function

$$H(z) = \frac{16z^2(z+1)}{(4z^2 - 2z + 1)(4z + 3)} \tag{34}$$

using

  - A cascade realization.
  - A parallel realization.

Example 4.1 - Solution

- One cascade realization is obtained by describing the original transfer function as a product of second- and first-order building blocks as follows

$$H(z) = \left(\frac{1}{1 - \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}} \right) \left(\frac{1 + z^{-1}}{1 + \frac{3}{4}z^{-1}} \right) \tag{35}$$

- Each section of the decomposed transfer function is implemented using the Type 1 canonic direct-form structure as illustrated in Figure 16.

Example 4.1 - Solution

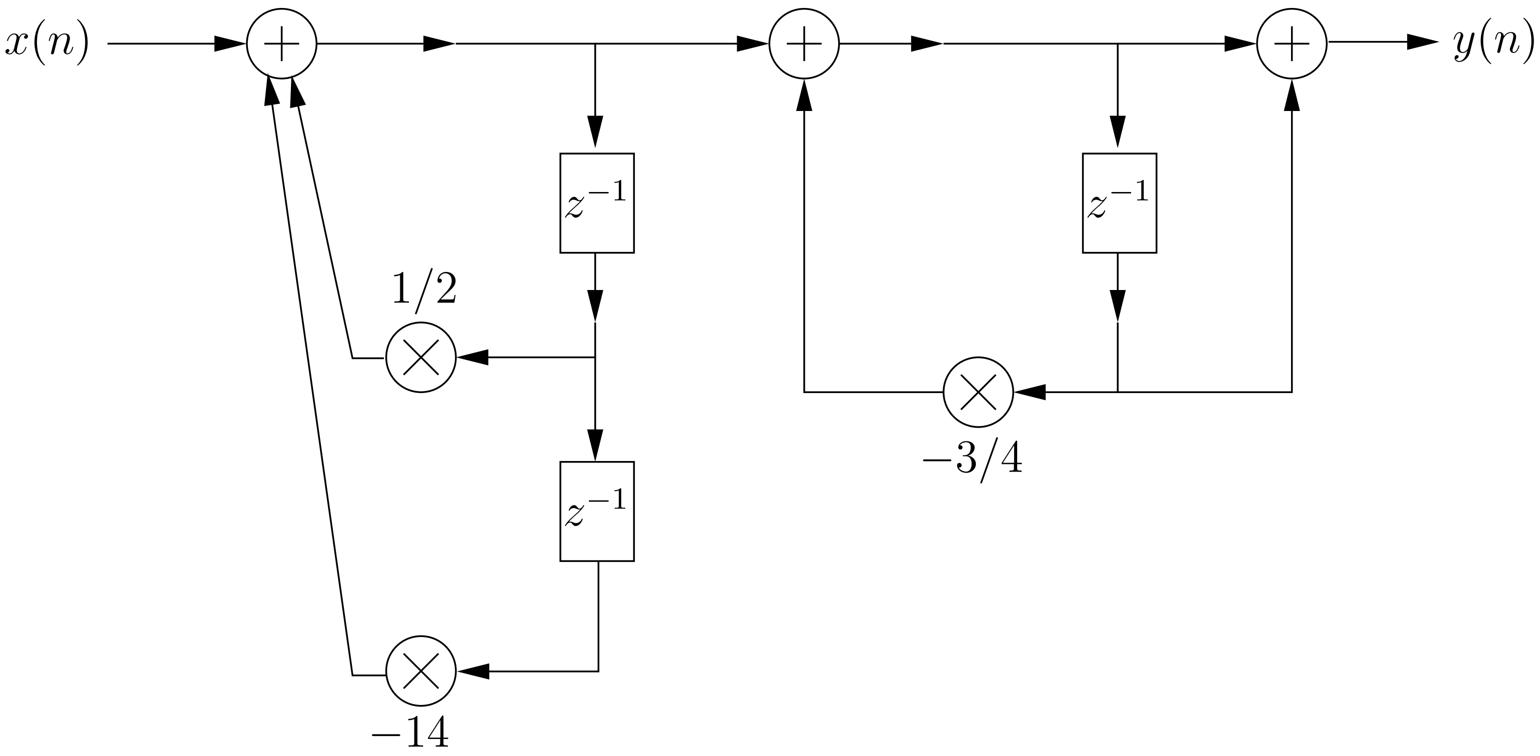


Figure 16: Cascade implementation of $\mathrm{H}(z)$ as given in equation (35).

Example 4.1 - Solution

- For the parallel form, let us write the original transfer function in a more convenient form as follows

$$
\begin{aligned}
H(z) &= \frac{z^2(z+1)}{\left(z^2 - \frac{1}{2}z + \frac{1}{4}\right)\left(z + \frac{3}{4}\right)} \\
&= \frac{z^2(z+1)}{\left(z - \frac{1}{4} - \mathrm{j}\frac{\sqrt{3}}{4}\right)\left(z - \frac{1}{4} + \mathrm{j}\frac{\sqrt{3}}{4}\right)\left(z + \frac{3}{4}\right)}
\end{aligned}
\tag{36}
$$

Example 4.1 - Solution

- Next, we decompose $H(z)$ as a summation of first-order complex sections as

$$H(z) = r_1 + \frac{r_2}{z - p_2} + \frac{r_2^*}{z - p_2^*} + \frac{r_3}{z + p_3} \tag{37}$$

where r_1 is the value of $H(z)$ at $z \to \infty$ and r_i is the residue associated to the pole p_i, for $i = 2, 3$, such that

$$\left.\begin{aligned} p_2 &= \tfrac{1}{4} + \mathrm{j}\tfrac{\sqrt{3}}{4} \\ p_3 &= \tfrac{3}{4} \\ r_1 &= 1 \\ r_2 &= \tfrac{6}{19} + \mathrm{j}\tfrac{5}{3\sqrt{19}} \\ r_3 &= \tfrac{9}{76} \end{aligned}\right\} \tag{38}$$

Example 4.1 - Solution

- Given these values, the complex first-order sections are properly grouped to form second-order sections with real coefficients, and the constant r_1 is grouped with the real coefficient first-order section, resulting in the following decomposition for $H(z)$:

$$H(z) = \frac{1 + \frac{66}{76}z^{-1}}{1 + \frac{3}{4}z^{-1}} + \frac{\frac{12}{19}z^{-1} - \frac{11}{38}z^{-2}}{1 - \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}} \tag{39}$$

- We can then implement each section using the Type 1 canonic direct-form structure leading to the realization shown in Figure 17.

Example 4.1 - Solution

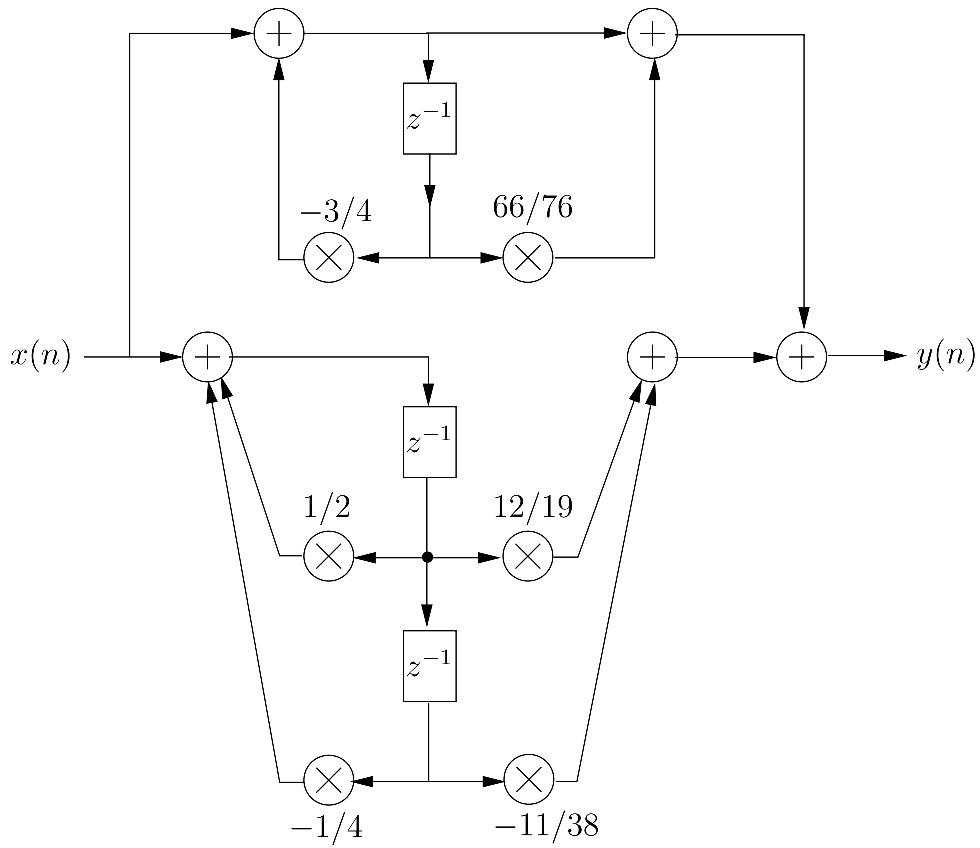


Figure 17: Parallel implementation of $\mathrm{H}(z)$ as given in equation (39).

Digital network analysis

- The signal-flowgraph representation greatly simplifies the analysis of digital networks composed of delays, multipliers, and adder elements.
- In practice, the analysis of such devices is implemented by first numbering all nodes of the graph of interest.
- Then, one determines the relationship between the output signal of each node with respect to the output signals of all other nodes.
- The connections between two nodes, referred to as branches, consist of combinations of delays and/or multipliers.
- Branches that inject external signals into the graph are called source branches. Such branches have a transmission coefficient equal to 1.

Digital network analysis

- Following this framework, we can describe the output signal of each node as a combination of the signals of all other nodes and possibly an external signal, that is

$$Y_j(z) = X_j(z) + \sum_{k=1}^{N} \left(a_{kj} Y_k(z) + z^{-1} b_{kj} Y_k(z) \right) \tag{40}$$

for $j = 1, 2, \ldots, N$, where N is the number of nodes, a_{kj} and $z^{-1} b_{kj}$ are the transmission coefficients of the branch connecting node k to node j, $Y_j(z)$ is the z transform of the output signal of node j, and $X_j(z)$ is the z transform of the external signal injected in node j.

Digital network analysis

- We can express equation (40) in a more compact form as

$$\mathbf{y}(z) = \mathbf{x}(z) + \mathbf{A}^{\mathsf{T}}\mathbf{y}(z) + \mathbf{B}^{\mathsf{T}}\mathbf{y}(z)z^{-1} \tag{41}$$

  where $\mathbf{y}(z)$ is the output signal $\mathrm{N} \times 1$ vector and $\mathbf{x}(z)$ is the external input signal $\mathrm{N} \times 1$ vector for all nodes in the given graph.

- Also, $\mathbf{A}^{\mathsf{T}}$ is an $\mathrm{N} \times \mathrm{N}$ matrix formed by the multiplier coefficients of the delayless branches of the circuit, while $\mathbf{B}^{\mathsf{T}}$ is the $\mathrm{N} \times \mathrm{N}$ matrix of multiplier coefficients of the branches with a delay.

Example 4.2

- Describe the digital filter seen in Figure 18 using the compact representation given in equation (41).

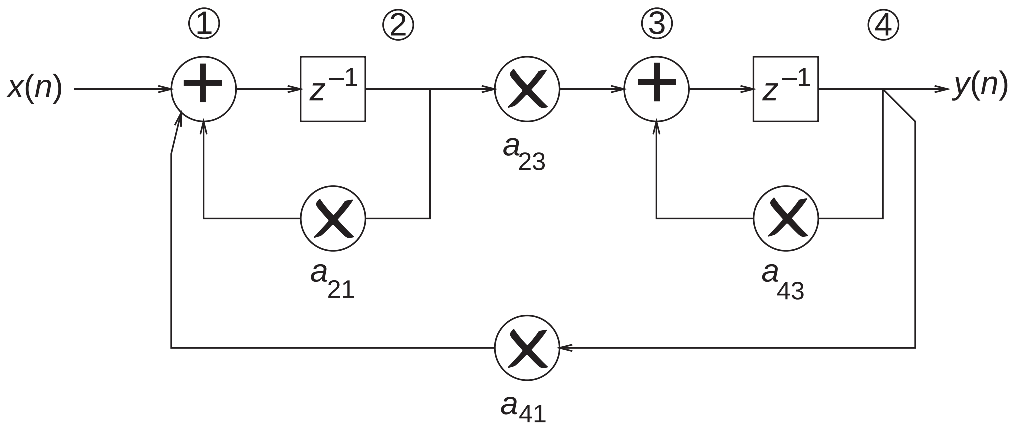


Figure 18: Second-order digital filter.

Example 4.2 - Solution

- In order to carry out the description of the filter as in equations (40) and (41), it is more convenient to represent it in the signal-flowgraph form, as in Figure 19.

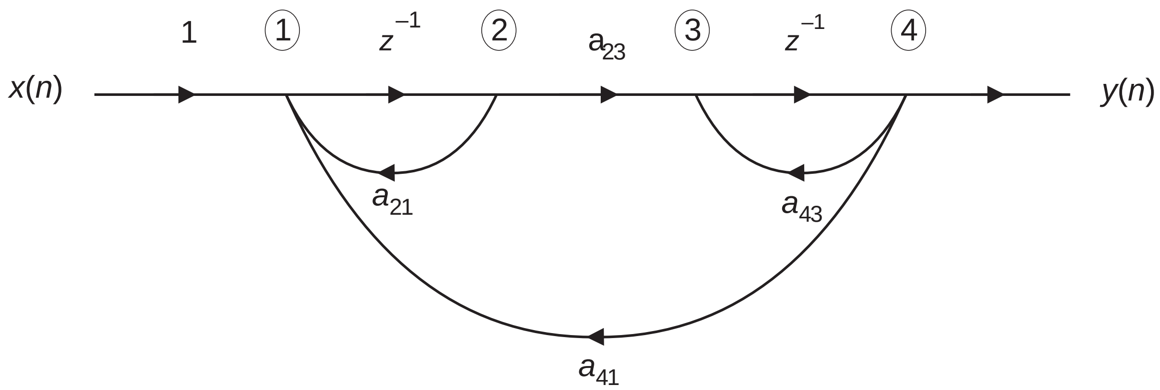


Figure 19: Signal-flowgraph representation of a digital filter.

Example 4.2 - Solution

- Following the procedure described above, one can easily write that

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix} = \begin{bmatrix} X_1(z) \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{21} & 0 & a_{41} \\ 0 & 0 & 0 & 0 \\ 0 & a_{23} & 0 & a_{43} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix}$$

$$+ \quad z^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix} \qquad (42)$$

Digital network analysis

- The z-domain signals $Y_j(z)$ associated with the network nodes are determined as

$$\mathbf{y}(z) = \mathbf{T}^{\mathsf{T}}(z)\mathbf{x}(z) \tag{43}$$

  with

$$\mathbf{T}^{\mathsf{T}}(z) = \left(\mathbf{I} - \mathbf{A}^{\mathsf{T}} - \mathbf{B}^{\mathsf{T}} z^{-1}\right)^{-1} \tag{44}$$

  where $\mathbf{I}$ is the Nth-order identity matrix.

- In the above equation, $\mathbf{T}(z)$ is the so-called transfer matrix, whose entry $T_{ij}(z)$ describes the transfer function from node i to node j, that is

$$T_{ij}(z) = \left.\frac{Y_j(z)}{X_i(z)}\right|_{X_k(z)=0,\, k=1,2,\ldots,N,\, k\neq j} \tag{45}$$

  and then $T_{ij}(z)$ gives the response at node j when the only nonzero input is applied to node i.

Digital network analysis

- If one is interested in the output signal of a particular node when several signals are injected into the network, equation (43) can be used to give

$$Y_j(z) = \sum_{i=1}^{N} T_{ij}(z) X_i(z) \tag{46}$$

- Equation (41) can be expressed in the time domain as

$$\mathbf{y}(n) = \mathbf{x}(n) + \mathbf{A}^T \mathbf{y}(n) + \mathbf{B}^T \mathbf{y}(n-1) \tag{47}$$

- If the above equation is used as a recurrence relation to determine the signal at a particular node given the input signal and initial conditions, there is no guarantee that the signal at a particular node does not depend on the signal at a different node whose output is yet to be determined.

Digital network analysis

- This can be avoided by the use of special node orderings, such as:

  (i) Enumerate the nodes only connected to either source branches or branches with a delay element. Note that for computing the outputs of these nodes, we need only the current values of the external input signals or values of the internal signals at instant $(n-1)$.

  (ii) Enumerate the nodes only connected to source branches, branches with a delay element, or branches connected to the nodes whose outputs were already computed in step (i). For this new group of nodes, their corresponding outputs depend on external signals, signals at instant $(n-1)$, or signals previously determined in step (i).

  (iii) Repeat the procedure above until the outputs of all nodes have been enumerated. The only case in which this is not achievable (notice that at each step, at least the output of one new node should be enumerated), occurs when the given network presents a delayless loop, which is of no practical use.

Example 4.3

- Analyze the network given in the previous example, using the algorithm described above.

Example 4.3 - Solution

- In the given example, the first group consists of nodes 2 and 4, and the second group consists of nodes 1 and 3. If we reorder the nodes 2, 4, 1, and 3 as 1, 2, 3, and 4, respectively, we end up with the network shown in Figure 20, which corresponds to

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ X_3(z) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_{21} & a_{41} & 0 & 0 \\ a_{23} & a_{43} & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix}$$
$$+ z^{-1} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \\ Y_4(z) \end{bmatrix} \tag{48}$$

Example 4.3 - Solution

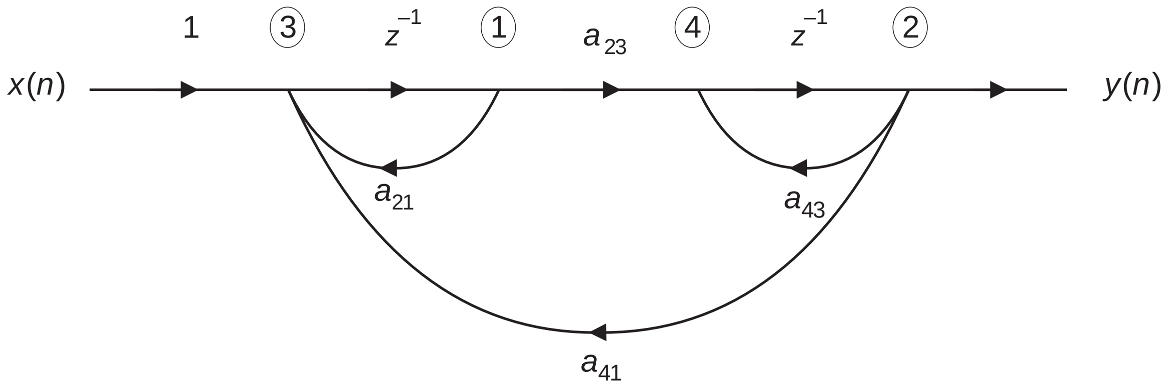


Figure 20: Reordering the nodes in the signal flowgraph.

Digital network analysis

- In general, after reordering, $\mathbf{A}^{\mathsf{T}}$ can be put in the following form:

$$\mathbf{A}^{\mathsf{T}} = \begin{bmatrix} 0 & \ldots & 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ a_{1j} & \ldots & a_{kj} & \ldots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ a_{1N} & \ldots & a_{kN} & \ldots & 0 & 0 \end{bmatrix} \tag{49}$$

- This means that $\mathbf{A}^{\mathsf{T}}$ and $\mathbf{B}^{\mathsf{T}}$ tend to be sparse matrices. Therefore, efficient algorithms can be employed to solve the time- or z-domain analyses described in this section.

State-space description

- An alternative form of representing digital filters is to use what is called the state-space representation.
- In such a description, the outputs of the memory elements (delays) are considered the system states.
- Once all the values of the external and state signals are known, we can determine the future values of the system states (the delay inputs) and the system output signals as follows:

$$\left.\begin{aligned} \mathbf{x}(n+1) &= \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) \\ \mathbf{y}(n) &= \mathbf{C}^{\mathrm{T}}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n) \end{aligned}\right\} \tag{50}$$

where $\mathbf{x}(n)$ is the $N \times 1$ vector of the state variables.

State-space description

- If M is the number of system inputs and M' is the number of system outputs, we have that **A** is $N \times N$, **B** is $N \times M$, **C** is $N \times M'$, and **D** is $M' \times M$.
- In general, we work with single-input and single-output systems. In such cases, **B** is $N \times 1$, **C** is $N \times 1$, and **D** is 1×1, that is, **D** $= d$ is a scalar.
- Note that this representation is essentially different from the one given in equation (47), because in that equation the variables are the outputs of each node, whereas in the state-space approach the variables are just the outputs of the delays.

State-space description

- The impulse response for a system as described in equation (50) is given by

$$h(n) = \begin{cases} d, & \text{for } n = 0 \\ \mathbf{c}^{\mathsf{T}}\mathbf{A}^{n-1}\mathbf{B}, & \text{for } n > 0 \end{cases} \tag{51}$$

- To determine the corresponding transfer function, we first apply the z transform to equation (50), obtaining (note that in this case both $\mathbf{y}(n)$ and $\mathbf{u}(n)$ are scalars)

$$\left.\begin{aligned} z\mathbf{X}(z) &= \mathbf{A}\mathbf{X}(z) + \mathbf{B}U(z) \\ Y(z) &= \mathbf{c}^{\mathsf{T}}\mathbf{X}(z) + dU(z) \end{aligned}\right\} \tag{52}$$

and then

$$H(z) = \frac{Y(z)}{U(z)} = \mathbf{c}^{\mathsf{T}}\left(z\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} + d \tag{53}$$

- From equation (53), it should be noticed that the poles of $H(z)$ are the eigenvalues of $\mathbf{A}$, as the denominator of $H(z)$ will be given by the determinant of $(z\mathbf{I} - \mathbf{A})$.

State-space description

- By applying a linear transformation $\mathbf{T}$ to the state vector such that

$$\mathbf{x}(n) = \mathbf{T}\mathbf{x}'(n) \tag{54}$$

where $\mathbf{T}$ is any $N \times N$ nonsingular matrix, we end up with a system characterized by

$$\mathbf{A}' = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}; \quad \mathbf{B}' = \mathbf{T}^{-1}\mathbf{B}; \quad \mathbf{C}' = \mathbf{T}^{\mathsf{T}}\mathbf{C}; \quad d' = d \tag{55}$$

- Such a system will present the same transfer function as the original system, and consequently the same poles and zeros. The proof of this fact is left to the interested reader as an exercise.

- In fact, the state-space representation leads to the minimum number of equations describing all the internal behavior and the input-output relationship associated to a given network, except for possible cases of insufficient controllability and observability characteristics.

Example 4.4

- Determine the state-space equations of the filters given in Figures 15a and 18.

Example 4.4 - Solution

- Associating a state-space variable $x_i(n)$ to each delay output, the corresponding delay input is represented by $x_i(n+1)$.
- In Figure 15a, let us use $i = 1$ in the upper delay and $i = 2$ in the lower one.
- The state-space description can be determined as:

  (i) The elements of the state-space transition matrix **A** can be obtained by inspection as follows: For each delay input $x_i(n+1)$, locate the direct paths from the states $x_j(n)$, for all j, without crossing any other delay input. In Figure 15a, the coefficients $-m_1$ and $-m_2$ form the only direct paths from the states $x_1(n)$ and $x_2(n)$, respectively, to $x_1(n+1)$. In addition, the relationship $x_1(n) = x_2(n+1)$ defines the only direct path from all states to $x_2(n+1)$.

  (ii) The elements of the input vector **B** represent the direct path between the input signal to each delay input without crossing any other delay input. In Figure 15a, only the delay input $x_1(n+1)$ is directly connected to the input signal with coefficient value of 1.

Example 4.4 - Solution

- (Cont.)

  (iii) The elements of the output vector **C** account for the direct connection between each state and the output node without crossing any delay input. In Figure 15a, the first state $x_1(n)$ has two direct connections with the output signal: One through the multiplier γ_1 and the other across the multipliers $-m_1$ and γ_0. Similarly, the second state $x_2(n)$ has direct connections to the output node through the multiplier γ_2 and through the cascade of $-m_2$ and γ_0.

  (iv) The feedforward coefficient d accounts for the direct connections between the input signal and the output node without crossing any state. In Figure 15a, there is a single direct connection though the multiplier with coefficient γ_0.

Example 4.4 - Solution

- Following the procedure described above, for the filter shown in Figure 15a, we have that

$$\left.\begin{aligned} \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} &= \begin{bmatrix} -m_1 & -m_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n) \\ y(n) &= \begin{bmatrix} (\gamma_1 - m_1\gamma_0) & (\gamma_2 - m_2\gamma_0) \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \gamma_0 u(n) \end{aligned}\right\} \quad (56)$$

Example 4.4 - Solution

- Using the same procedure for the filter in Figure 18, we get

$$\left.\begin{array}{l} x_1(n+1) = a_{21}x_1(n) + a_{41}x_2(n) + u(n) \\ x_2(n+1) = a_{23}x_1(n) + a_{43}x_2(n) \\ y(n) = x_2(n) \end{array}\right\} \quad (57)$$

leading to the following state-space description:

$$\left.\begin{array}{l} \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} a_{21} & a_{41} \\ a_{23} & a_{43} \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(n) \\ y(n) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + 0u(n) \end{array}\right\} \quad (58)$$

- Note that each state-space element is represented by a single coefficient, since there is at most one direct path among the states, input node, and output node.

Basic properties of digital networks

- In this section, we introduce some network properties that are very useful for designing and analyzing digital filters.
- More specifically, we see the following concepts:
  - Tellegen's theorem
  - Reciprocity and interreciprocity
  - Transposition
  - Sensitivity

Tellegen's theorem

- Consider a digital network represented by the corresponding signal flowgraph, in which the signal at node j is y_j, and the signal that reaches node j from node i is denoted by x_{ij}. We can use this notation also to represent a branch that leaves from and arrives at the same node. Such a branch is called a loop.
- Among other things, loops are used to represent source branches entering a node. In fact, every source branch will be represented as a loop having the value of its source. In this case, x_{ii} includes the external signal and any other loop connecting node i to itself.
- Following this framework, for each node of a given graph, we can write that

$$y_j = \sum_{i=1}^{N} x_{ij} \tag{59}$$

where N in this case is the total number of nodes. Consider now the following result.

Tellegen's theorem

- **Telegen's Theorem:** All corresponding signals, (x_{ij}, y_j) and (x'_{ij}, y'_j), of two distinct networks with equal signal-flowgraph representations satisfy

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(y_j x'_{ij} - y'_i x_{ji}) = 0 \tag{60}$$

where both sums include all nodes of both networks.

- **Prrof:** Equation (60) can be rewritten as

$$\sum_{j=1}^{N}\left(y_j \sum_{i=1}^{N} x'_{ij}\right) - \sum_{i=1}^{N}\left(y'_i \sum_{j=1}^{N} x_{ji}\right) = \sum_{j=1}^{N} y_j y'_j - \sum_{i=1}^{N} y'_i y_i = 0 \tag{61}$$

which completes the proof.

Tellegen's theorem

- Tellegen's theorem can be generalized for the frequency domain, since

$$Y_j = \sum_{i=1}^{N} X_{ij} \tag{62}$$

and then

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \left(Y_j X'_{ij} - Y'_i X_{ji}\right) = 0 \tag{63}$$

- Notice that in Tellegen's theorem x_{ij} is actually the sum of all signals departing from node i and arriving at node j. Therefore, in the most general case where the two graphs have different topologies, Tellegen's theorem can still be applied, making the two topologies equal by adding as many nodes and branches with null transmission values as necessary.

Reciprocity

- Consider a particular network in which M of its nodes each have two branches connecting them to the outside world, as depicted in Figure 21.
- The first branch in each of these nodes is a source branch through which an external signal is injected into the node. The second branch makes the signal of each node available as an output signal.
- Naturally, in nodes where there are neither external input nor output signals, one must consider the corresponding branch to have a null transmission value.
- For generality, if a node does not have neither external nor output signals, both corresponding branches are considered to have a null transmission value.

Reciprocity

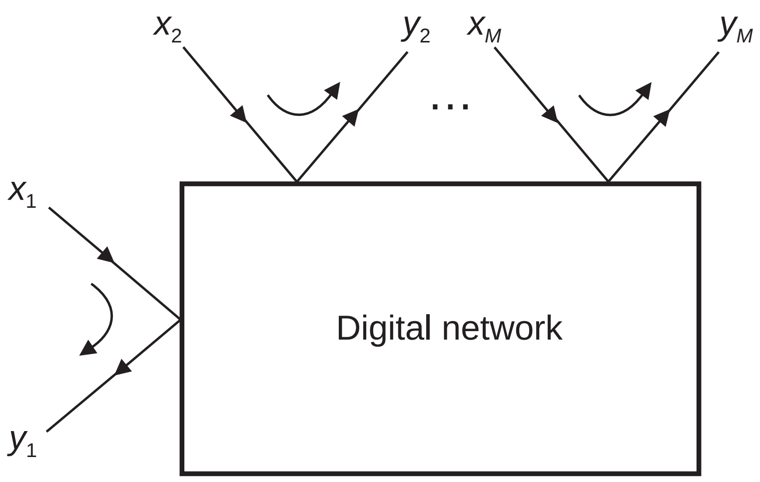


Figure 21: General digital network with M ports.

Reciprocity

- Suppose that we apply a set of signals X_i to this network, and collect as output the signals Y_i.
- Alternatively, we could apply the signals X'_i, and observe the Y'_i signals.
- The particular network is said to be reciprocal if

$$\sum_{i=1}^{M} (X_i Y'_i - X'_i Y_i) = 0 \tag{64}$$

- In such cases, if the M-port network is described by

$$Y_i = \sum_{j=1}^{M} T_{ji} X_j \tag{65}$$

where T_{ji} is the transfer function from port j to port i, equation (64) is equivalent to

$$T_{ij} = T_{ji} \tag{66}$$

Reciprocity

- Proof for this statement is based on the substitution of equation (65) into equation (64):

$$\begin{aligned}\sum_{i=1}^{M}\left(X_i\sum_{j=1}^{M}T_{ji}X'_j - X'_i\sum_{j=1}^{M}T_{ji}X_j\right) &= \sum_{i=1}^{M}\sum_{j=1}^{M}\left(X_iT_{ji}X'_j\right) - \sum_{i=1}^{M}\sum_{j=1}^{M}\left(X'_iT_{ji}X_j\right)\\ &= \sum_{i=1}^{M}\sum_{j=1}^{M}\left(X_iT_{ji}X'_j\right) - \sum_{i=1}^{M}\sum_{j=1}^{M}\left(X'_jT_{ij}X_i\right)\\ &= \sum_{i=1}^{M}\sum_{j=1}^{M}\left(T_{ji}-T_{ij}\right)\left(X_iX'_j\right)\\ &= 0 \end{aligned} \tag{67}$$

and thus

$$T_{ij} = T_{ji} \tag{68}$$

Interreciprocity

- The vast majority of the digital networks associated with digital filters are not reciprocal. However, such a concept is crucial in some cases. Fortunately, there is another related property, called interreciprocity, between two networks which is very common and useful.

- Consider two networks with the same number of nodes, and also consider that X_i and Y_i are respectively input and output signals of the first network. Correspondingly, X_i' and Y_i' represent input and output signals of the second network.

- Such networks are considered interreciprocal if equation (64) holds for (X_i, Y_i) and (X_i', Y_i'), $i = 1, 2, \ldots, M$.

Interreciprocity

- If two networks are described by

$$Y_i = \sum_{j=1}^{M} T_{ji} X_j \tag{69}$$

and

$$Y'_i = \sum_{j=1}^{M} T'_{ji} X'_j \tag{70}$$

it can be easily shown that these two networks are interreciprocal if

$$T_{ji} = T'_{ij} \tag{71}$$

Once again, the proof is left as an exercise for the interested reader.

Transposition

- Given any signal-flowgraph representation of a digital network, we can generate another network by reversing the directions of all branches. In such a procedure, all addition nodes turn into distribution nodes, and vice versa.
- Also, if in the original network the branch from node i to node j is F_{ij} (that is, $X_{ij} = F_{ij} Y_j$), the transpose network will have a branch from node j to node i with transmission F'_{ji} such that

$$F_{ij} = F'_{ji} \tag{72}$$

- Using Tellegen's theorem, one can easily show that the original network and its corresponding transpose network are interreciprocal.

Transposition

- In fact, if we represent X_i as X_{ii} and number the M input-output nodes in Figure 21 as 1 to M, leaving indexes $(M+1)$ to N to represent the internal nodes, by applying Tellegen's theorem to all signals of both networks, one obtains

$$\begin{aligned}
\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i,\ \text{if}\ i<M+1}}^{N} \left(Y_j X'_{ij} - Y'_i X_{ji}\right) + \sum_{i=1}^{M} (Y_i X'_{ii} - Y'_i X_{ii}) \\
= \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i,\ \text{if}\ i<M+1}}^{N} \left(Y_j F'_{ij} Y'_i - Y'_i F_{ji} Y_j\right) + \sum_{i=1}^{M} (Y_i X'_{ii} - Y'_i X_{ii}) \\
= 0 + \sum_{i=1}^{M} (Y_i X'_{ii} - Y'_i X_{ii}) \\
= 0 \qquad (73)
\end{aligned}$$

where all external signals are considered to be injected at the first M nodes.

Transposition

- Naturally, $\sum_{i=1}^{M}(Y_i X'_{ii} - Y'_i X_{ii}) = 0$ is equivalent to interreciprocity (see equation (64) applied to the interreciprocity case), which implies that (equation (71))

$$T_{ij} = T'_{ji} \tag{74}$$

- This is a very important result, because it indicates that a network and its transpose must have the same transfer function.

- For instance, the equivalence of the networks in Figures 3 and 4 can be deduced from the fact that one is the transpose of the other. The same can be said about the networks in Figures 12 and 13.

Sensitivity

- Sensitivity is a measure of the degree of variation of a network's overall transfer function with respect to small fluctuations in the value of one of its elements.

- In the specific case of digital filters, one is often interested in the sensitivity with respect to variations of the multiplier coefficients, that is

$$S_{m_i}^{H(z)} = \frac{\partial H(z)}{\partial m_i} \tag{75}$$

for $i = 1, 2, \ldots, L$, where L is the total number of multipliers in the particular network.

- Using the concept of transposition, we can determine the sensitivity of $H(z)$ with respect to a given coefficient m_i in a very efficient way. To understand how, consider a network, its transpose, and also the original network with a specific coefficient slightly modified, as depicted in Figure 22.

Sensitivity

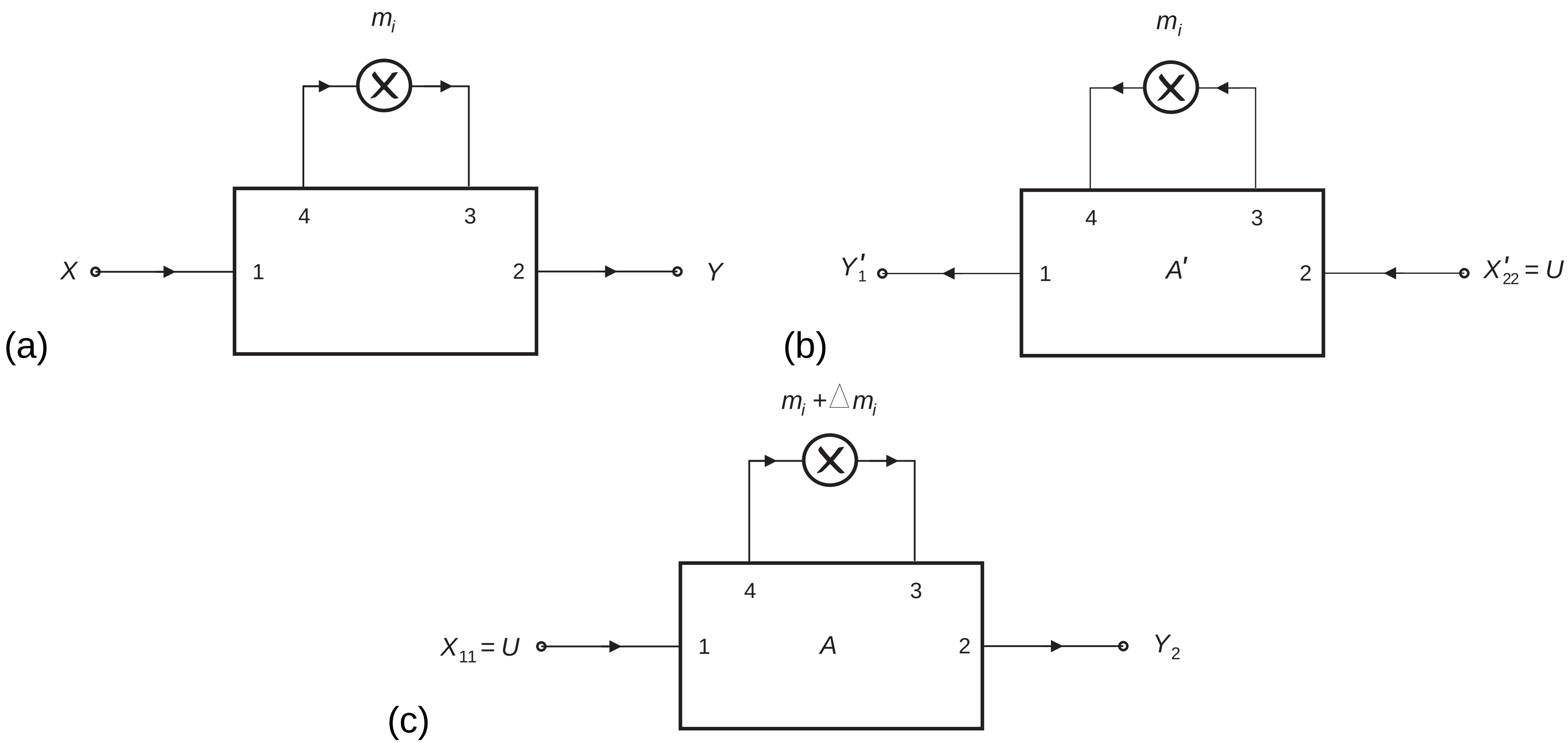


Figure 22: Digital networks: (a) original; (b) transpose; (c) original with modified coefficient.

Sensitivity

- Using Tellegen's theorem on the networks shown in Figures 22b and 22c, one obtains

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N}\left(Y_j X'_{ij} - Y'_i X_{ji}\right) &= \underbrace{\sum_{j=1}^{N}\left(Y_j X'_{1j} - Y'_1 X_{j1}\right)}_{A_1} + \underbrace{\sum_{j=1}^{N}\left(Y_j X'_{2j} - Y'_2 X_{j2}\right)}_{A_2} \\
&\quad + \underbrace{\sum_{j=1}^{N}\left(Y_j X'_{3j} - Y'_3 X_{j3}\right)}_{A_3} + \underbrace{\sum_{i=4}^{N}\sum_{j=1}^{N}\left(Y_j X'_{ij} - Y'_i X_{ji}\right)}_{A_4} \\
&= A_1 + A_2 + A_3 + A_4 \\
&= 0 \qquad (76)
\end{aligned}
$$

where A_1, A_2, A_3, and A_4 are separately determined below.

Sensitivity

- For A_1:

$$\begin{aligned} A_1 &= Y_1 X'_{11} - Y'_1 X_{11} + \sum_{j=2}^{N} \left(Y_j X'_{1j} - Y'_1 X_{j1} \right) \\ &= -UY'_1 + \sum_{j=2}^{N} \left(Y_j F'_{1j} Y'_1 - Y'_1 F_{j1} Y_j \right) \\ &= -UY'_1 \end{aligned} \tag{77}$$

since $F'_{1j} = F_{j1}$, for all j.

Sensitivity

- In addition,

$$\begin{aligned} A_2 &= Y_2 X'_{22} - Y'_2 X_{22} + \sum_{\substack{j=1 \\ j \neq 2}}^{N} \left(Y_j X'_{2j} - Y'_2 X_{j2} \right) \\ &= U Y_2 + \sum_{\substack{j=1 \\ j \neq 2}}^{N} \left(Y_j F'_{2j} Y'_2 - Y'_2 F_{j2} Y_j \right) \\ &= U Y_2 \end{aligned} \tag{78}$$

Sensitivity

- For A_3:

$$A_3 = Y_4 X'_{34} - Y'_3 X_{43} + \sum_{\substack{j=1 \\ j\neq 4}}^{N} \left(Y_j F'_{3j} Y'_3 - Y'_3 F_{j3} Y_j \right)$$
$$= Y_4 m_i Y'_3 - Y'_3 \left(m_i + \Delta m_i \right) Y_4$$
$$= -\Delta m_i Y_4 Y'_3 \qquad (79)$$

- Finally, for A_4:

$$A_4 = \sum_{i=4}^{N} \sum_{j=1}^{N} \left(Y_j X'_{ij} - Y'_i X_{ji} \right)$$
$$= \sum_{i=4}^{N} \sum_{j=1}^{N} \left(Y_j F'_{ij} Y'_i - Y'_i F_{ji} Y_j \right)$$
$$= 0 \qquad (80)$$

Sensitivity

- Hence, one has that

$$-UY'_1 + UY_2 - \Delta m_i Y_4 Y'_3 = 0 \tag{81}$$

and thus

$$U\left(Y_2 - Y'_1\right) = \Delta m_i Y_4 Y'_3 \tag{82}$$

- Defining

$$\Delta H_{12} = \left(H_{12} - H'_{21}\right) = \left(\frac{Y_2}{U} - \frac{Y'_1}{U}\right) \tag{83}$$

one gets, from equation (82), that

$$U^2\left(H_{12} - H'_{21}\right) = U^2 \Delta H_{12} = \Delta m_i Y_4 Y'_3 \tag{84}$$

Sensitivity

- If we now let Δm_i converge to zero, H_{12} tends to H'_{21}, and consequently

$$\frac{\partial H_{12}}{\partial m_i} = \frac{Y_4 Y'_3}{U^2} = H'_{23} H_{14} = H_{32} H_{14} \tag{85}$$

- This equation indicates that the sensitivity of the transfer function of the original network, H_{12}, with respect to variations of one of its coefficients, can be determined based on transfer functions between the system input and the node before the multiplier, H_{14}, and between the multiplier output node and the system output, H_{32}.

Example 4.5

- Determine the sensitivity of $H(z)$ with respect to the coefficients a_{11}, a_{22}, a_{12}, and a_{21} in the network of Figure 23.

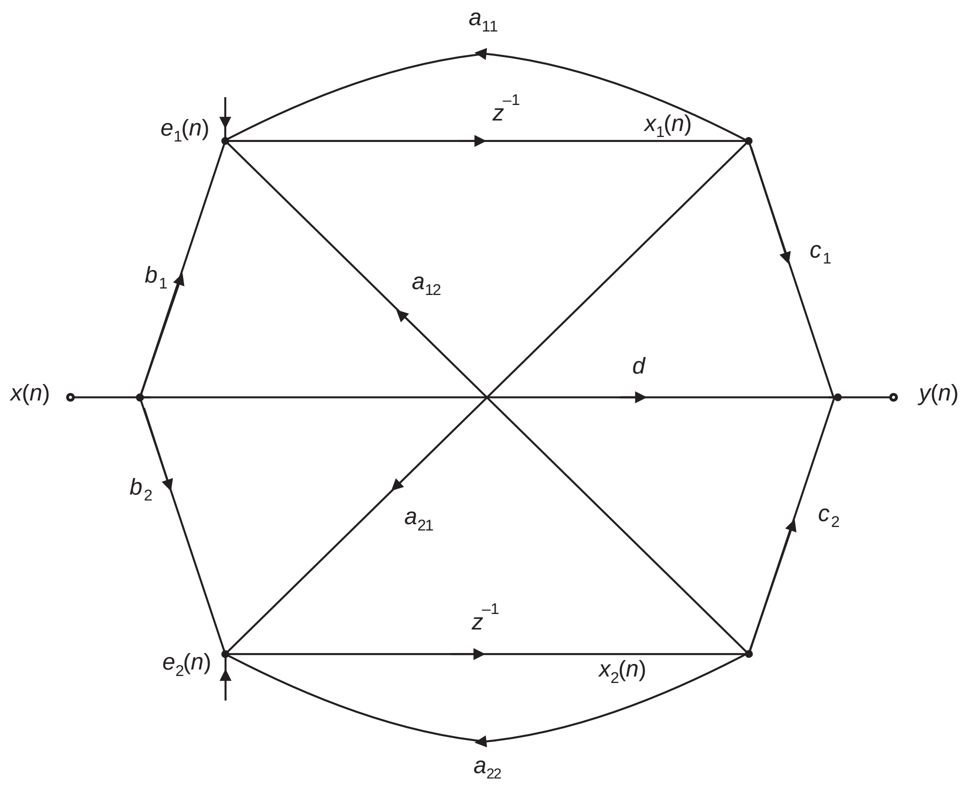


Figure 23: State variable network.

Example 4.5 - Solution

- The state-space description of the network in Figure 23 can be determined by using the procedure described in Example 4.4 for the network in Figure 15a.
- In the present case, entries of the state-space matrices correspond exactly to the multiplier coefficients of the network (due to this fact, the network in Figure 23 is called the state-space structure):

$$\left.\begin{array}{l} \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(n) \\ y(n) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + du(n) \end{array}\right\} \quad (86)$$

Example 4.5 - Solution

- The transfer function of the state-space structure is given by

$$\begin{aligned} H(z) &= \mathbf{C}^T (z\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + d \\ &= \frac{(b_1c_1 + b_2c_2)z + b_1c_2a_{12} + b_2c_1a_{21} - b_1c_1a_{22} - b_2c_2a_{11}}{D(z)} + d \end{aligned} \quad (87)$$

with

$$D(z) = z^2 - (a_{11} + a_{22})z + (a_{11}a_{22} - a_{12}a_{21}) \quad (88)$$

- The transfer functions required for computing the desired sensitivity functions can be obtained as special cases of the general transfer function $H(z)$.

Example 4.5 - Solution

- For instance, the transfer function from the filter input to the state $x_1(n)$ is obtained by setting $c_1 = 1$, $c_2 = 0$, and $d = 0$ in equation (87), leading to

$$F_1(z) = \frac{X_1(z)}{X(z)} = \frac{b_1 z + (b_2 a_{21} - b_1 a_{22})}{D(z)} \tag{89}$$

- The transfer function from the filter input to state $x_2(n)$ is obtained by setting $c_1 = 0$, $c_2 = 1$, and $d = 0$ in equation (87), resulting in

$$F_2(z) = \frac{X_2(z)}{X(z)} = \frac{b_2 z + (b_1 a_{12} - b_2 a_{11})}{D(z)} \tag{90}$$

Example 4.5 - Solution

- Using $b_1 = 1$, $b_2 = 0$, and $d = 0$ in equation (87), one determines the transfer function from state $x_1(n)$ to the filter output, such that

$$G_1(z) = \frac{Y(z)}{E_1(z)} = \frac{c_1 z + (c_2 a_{12} - c_1 a_{22})}{D(z)} \tag{91}$$

- Finally, the transfer function from state $x_2(n)$ to the filter output is

$$G_2(z) = \frac{Y(z)}{E_2(z)} = \frac{c_2 z + (c_1 a_{21} - c_2 a_{11})}{D(z)} \tag{92}$$

which is determined by setting $b_1 = 0$, $b_2 = 1$, and $d = 0$ in equation (87).

Example 4.5 - Solution

- The required sensitivities are then

$$S_{a_{11}}^{H(z)} = F_1(z)G_1(z) \tag{93}$$

$$S_{a_{22}}^{H(z)} = F_2(z)G_2(z) \tag{94}$$

$$S_{a_{12}}^{H(z)} = F_2(z)G_1(z) \tag{95}$$

$$S_{a_{21}}^{H(z)} = F_1(z)G_2(z) \tag{96}$$

Example 4.6

- Given the digital filter structure of Figure 24:

  (a) Generate its transposed realization.

  (b) Derive the structure to compute the sensitivity of the transfer function with respect to multiplier coefficient λ_1.

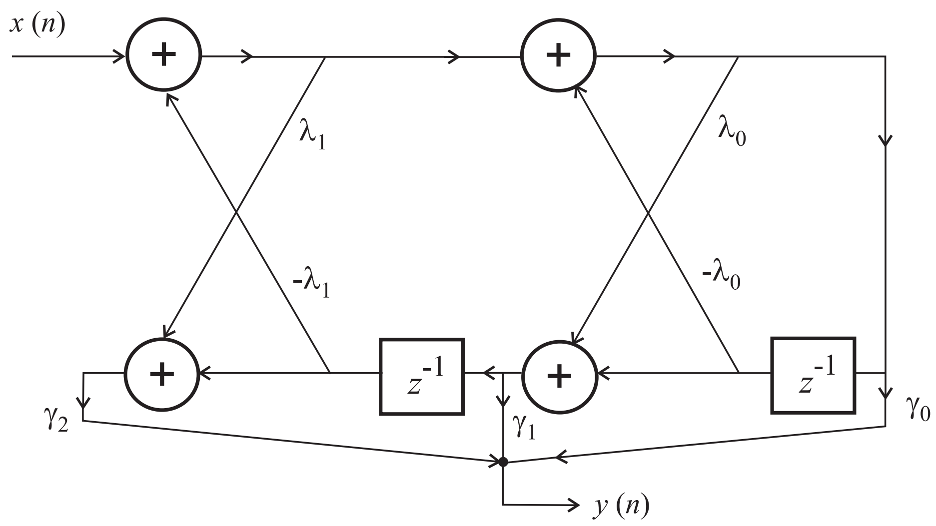


Figure 24: Second-order lattice structure in Example 4.6.

Example 4.6 - Solution

(a) Starting from the structure of Figure 24, we can obtain the transposed lattice structure by changing the branches directions, turning the summations nodes into distribution nodes and vice versa, and redrawing the network so that the input is placed at the right-hand side, as given in Figure 25.

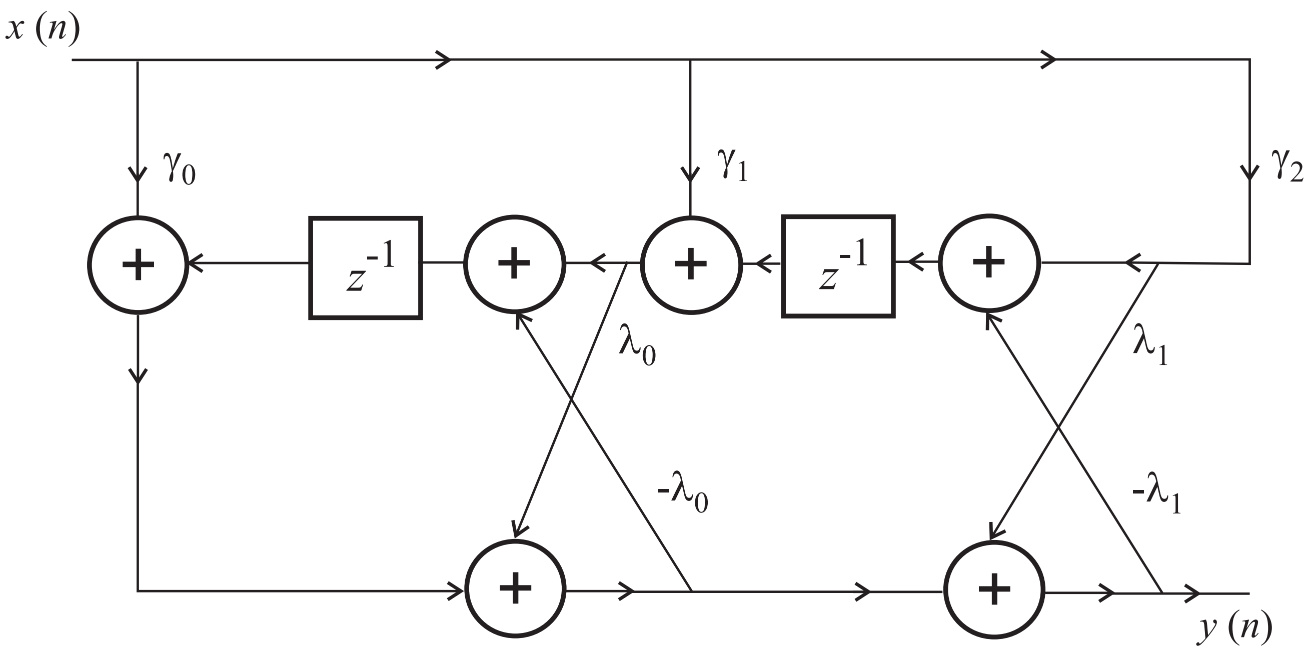


Figure 25: Transposed lattice structure in Example 4.6.

Example 4.6 - Solution

(b) For the sensitivity calculation, we must first note that the multiplier λ_1 appears in the network also as $-\lambda_1$. Actually, if they were two different multipliers λ_1 and $\overline{\lambda}_1$, respectively, we would have that

$$\Delta H(z) = \frac{\partial H(z)}{\partial \lambda_1}\Delta\lambda_1 + \frac{\partial H(z)}{\partial \overline{\lambda}_1}\Delta\overline{\lambda}_1 \tag{97}$$

Since $\overline{\lambda}_1 = -\lambda_1$, we have that $\Delta\overline{\lambda}_1 = -\Delta\lambda_1$, and thus

$$\frac{\Delta H(z)}{\Delta\lambda_1} = \frac{\partial H(z)}{\partial \lambda_1} - \frac{\partial H(z)}{\partial \overline{\lambda}_1} \tag{98}$$

As indicated in Figure 22, from equation (85), if multiplier λ_1 goes from node 4 to 3, and multiplier $-\lambda_1$ goes from node $4'$ do $3'$, the above equation becomes

$$\frac{\Delta H(z)}{\Delta\lambda_1} = H_{14}(z)H_{32}(z) - H_{14'}(z)H_{3'2}(z) \tag{99}$$

Example 4.6 - Solution

- We now build a network to compute the above equation. The upper subnetwork of Figure 26 computes two outputs, one equal to $H_{14}(z)X(z)$ and another equal to $H_{14'}(z)X(z)$.
- We then use the lower subnetwork of Figure 26 to compute $H_{32}(z)$ and $H_{3'2}(z)$.
- The output $H_{14}(z)X(z)$ is input to node 3 and the output $H_{14'}(z)X(z)$ is multiplied by -1 and input to node $3'$ (note the -1 multiplier in Figure 26).
- The output of the network in Figure 26 is then $H_{14}(z)H_{32}X(z) - H_{14'}(z)H_{3'2}X(z)$.
- From equation (99), this implies that its transfer function is equal to $\frac{\partial H(z)}{\partial \lambda_1}$.

Example 4.6 - Solution

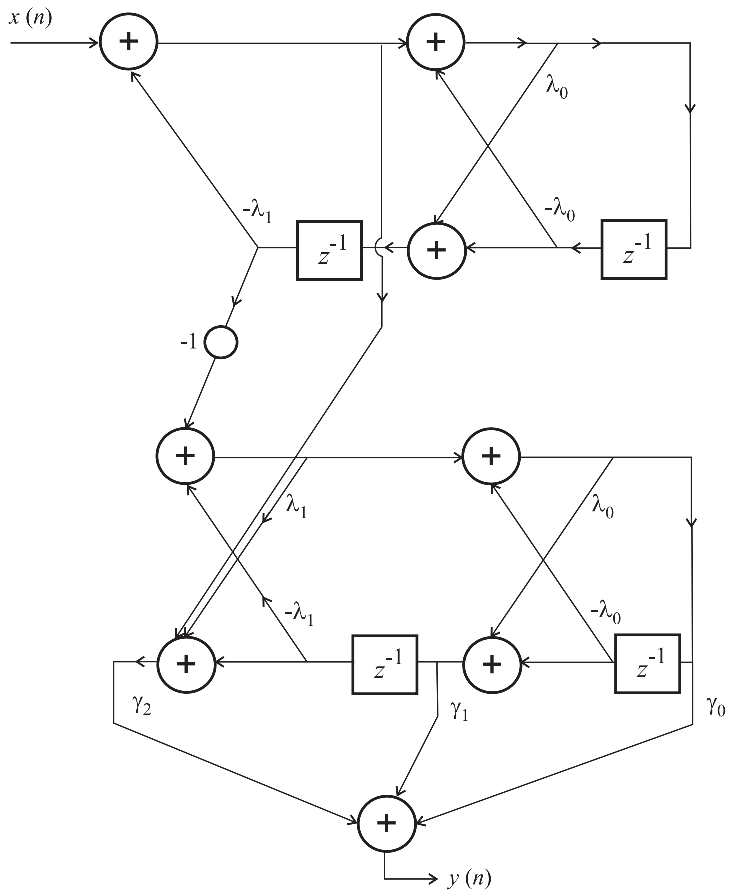


Figure 26: Derivative structure in Example 4.6.

Useful building blocks

- In this section, several building blocks with particularly attractive features are presented and briefly analyzed.
- More specifically, we see:
  - Second-order building blocks
  - Digital oscillators
  - Comb filter

Second-order building blocks

- The typical second-order transfer functions resulting from classical approximation methods are lowpass, bandpass, highpass, lowpass notch, highpass notch, and allpass.
- The transfer functions discussed below are special cases where the numerator polynomial is constrained to have its zeros either on the unit circle, where the zeros are more effective in shaping the magnitude response, or are reciprocals of the poles, as in the allpass case.

Second-order building blocks - Lowpass

- Lowpass filter:

$$H(z) = \frac{(z+1)^2}{z^2 + m_1 z + m_2} \tag{100}$$

- In this transfer function, the zeros are placed at $z = -1$, leading to trivial coefficients in the numerator.

- Typically the magnitude response will be increasing close to $z = 1$, it will reach a maximum value at the frequency corresponding directly to the poles angles, and then it will decrease to reach a zero value at $z = -1$, as illustrated in Figure 27a.

Second-order building blocks - Bandpass

- Bandpass filter:

$$H(z) = \frac{z^2 - 1}{z^2 + m_1 z + m_2} = \frac{(z-1)(z+1)}{z^2 + m_1 z + m_2} \tag{101}$$

- In this case, the zeros are placed at $z = \pm 1$, also leading to trivial coefficients in the numerator.

- Typically the magnitude response will be zero at $z = \pm 1$, and will reach a maximum value at frequency directly related to the poles angles, as depicted in Figure 27b.

Second-order building blocks - Highpass

- Highpass filter:

$$H(z) = \frac{(z-1)^2}{z^2 + m_1 z + m_2} \tag{102}$$

- As can be observed in Figure 27c, the zeros are placed at $z = 1$, so that all numerator coefficients are simple to implement.

- The magnitude response will be decreasing close to $z = -1$, it will reach a maximum value at a frequency directly related to the poles angles, and then it will decrease to reach a zero value at $z = 1$.

Second-order building blocks - Notch

- Notch filter:

$$H(z) = \frac{z^2 + \frac{m_1}{\sqrt{m_2}} z + 1}{z^2 + m_1 z + m_2} \tag{103}$$

- The zeros are placed on the unit circle with angles coinciding with the poles angles, whereas the poles are obviously placed inside the unit circle as in all other building blocks discussed here (this requires that $m_2 < 1$).

- An example is shown in Figure 27d.

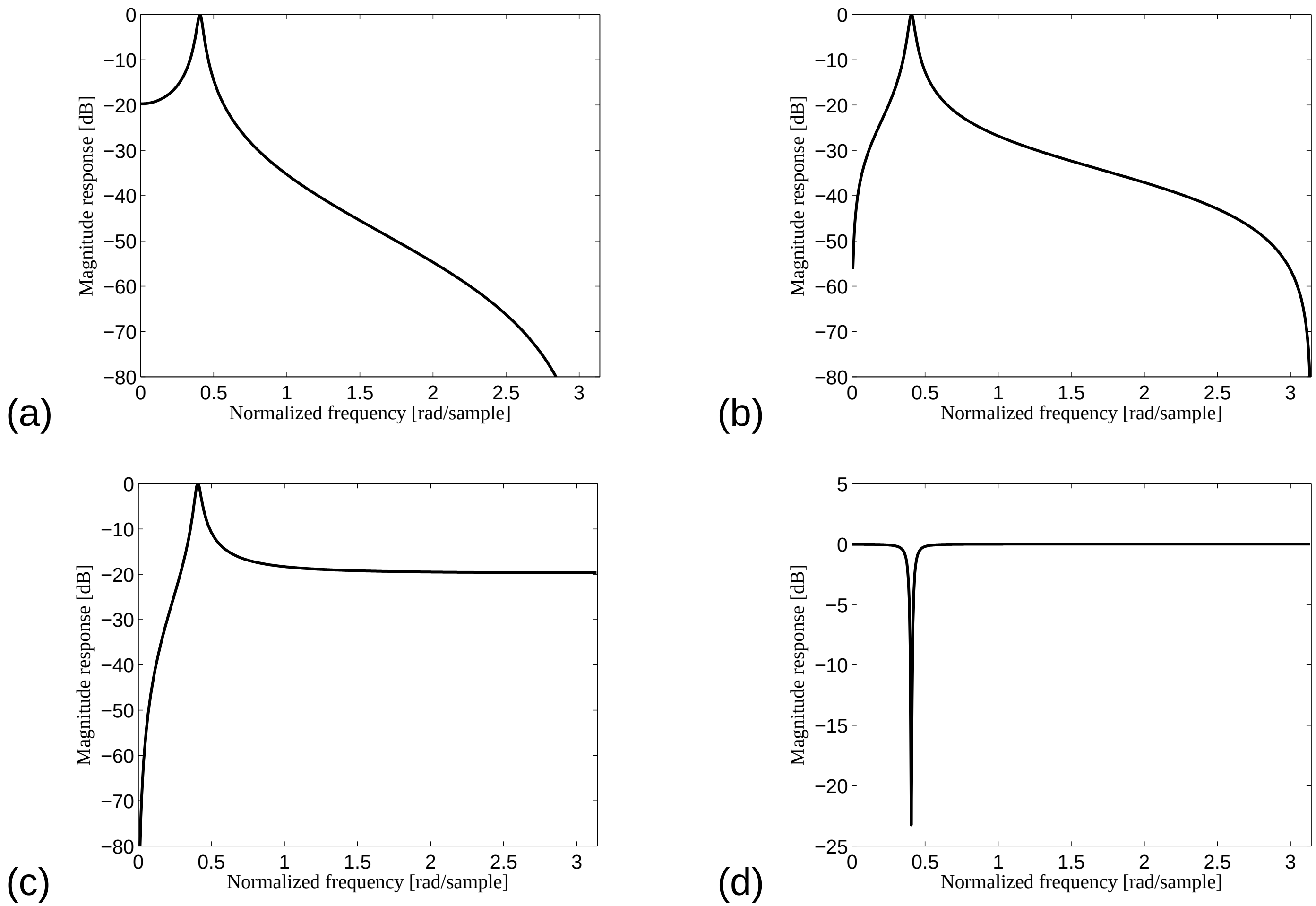


Figure 27: Magnitude responses of normalized, standard, second-order blocks with $m_1 = -1.8$ and $m_2 = 0.96$: (a) lowpass; (b) bandpass; (c) highpass; (d) notch.

Second-order building blocks - Notch

- Lowpass/highpass notch filters:

$$H(z) = \frac{z^2 + m_3 z + 1}{z^2 + m_1 z + m_2} \tag{104}$$

- In the highpass case, the zero at positive frequency is placed on the unit circle with smaller positive angle than the pole positive angle, and in the lowpass case, with larger positive angle than the pole positive angle.

- Figures 28a and 28b show typical magnitude responses of lowpass and highpass notch filters, respectively.

Second-order building blocks - Allpass

- Allpass filters:

$$H(z) = \frac{m_2 z^2 + m_1 z + 1}{z^2 + m_1 z + m_2} \tag{105}$$

- For allpass filters the zeros are reciprocals of the poles, that is, if p_1 and p_1^* are the stable filter poles, then $z_1^* = \frac{1}{p_1}$ and $z_1 = \frac{1}{p_1^*}$ are the zeros.

- Note that from equation (105)

$$H(z) \quad = \quad z^2 \frac{m_2 + m_1 z^{-1} + z^{-2}}{z^2 + m_1 z + m_2} = z^2 \frac{A(z^{-1})}{A(z)} \tag{106}$$

and the magnitude response is

$$|H(e^{j\omega})| \quad = \quad \frac{|A(e^{-j\omega})|}{|A(e^{j\omega})|} = \frac{|A^*(e^{j\omega})|}{|A(e^{j\omega})|} = 1 \tag{107}$$

since m_1 and m_2 are real.

Second-order building blocks - Allpass

- The magnitude and phase responses of an allpass filter are shown in Figures 28c and 28d, respectively.
- Such blocks are usually employed in delay equalizers, since they modify the phase without changing the magnitude.

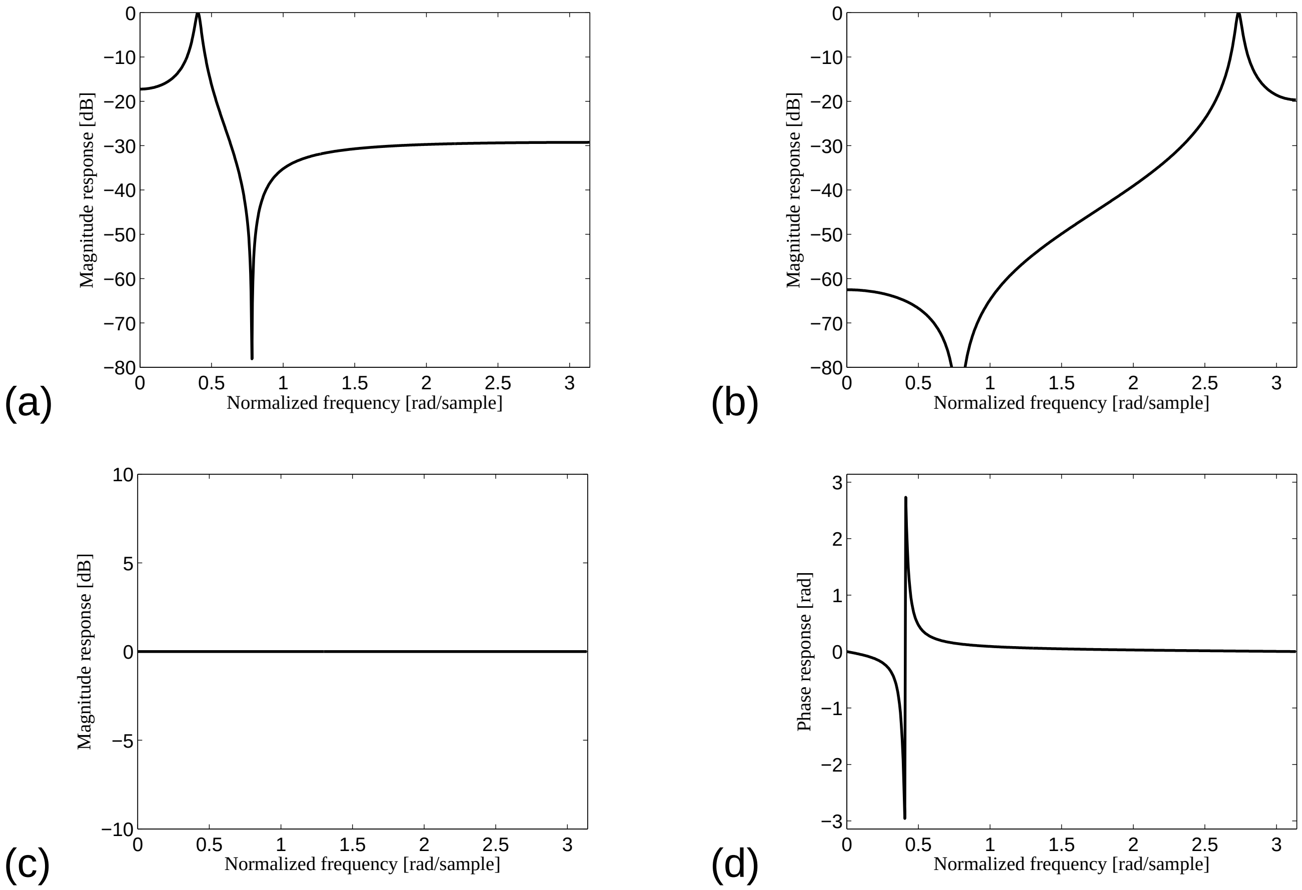


Figure 28: Magnitude (and phase) responses for normalized, standard, second-order blocks: (a) lowpass notch with $m_1 = -1.8$, $m_2 = 0.96$, and $m_3 = -1.42$; (b) highpass notch with $m_1 = 1.8$, $m_2 = 0.96$, and $m_3 = -1.42$; (c) allpass (magnitude) with $m_1 = -1.8$ and $m_2 = 0.96$; (d) allpass (phase) with $m_1 = -1.8$ and $m_2 = 0.96$.

Digital oscillators

- A realization of a digital oscillator has the transfer function

$$H(z) = \frac{z \sin(\omega_0)}{z^2 - 2\cos(\omega_0)z + 1} \tag{108}$$

  where the poles are placed exactly on the unit circle.

- According to Table 2.1, the impulse response for this system is an oscillation of the form $\sin(\omega_0 n)u(n)$, as illustrated in Figure 29 for $\omega_0 = \frac{7\pi}{10}$.

- Note that in this case the self-sustained oscillation does not look like a simple sinusoid, because the sampling frequency is not a multiple of the oscillation frequency.

Digital oscillators

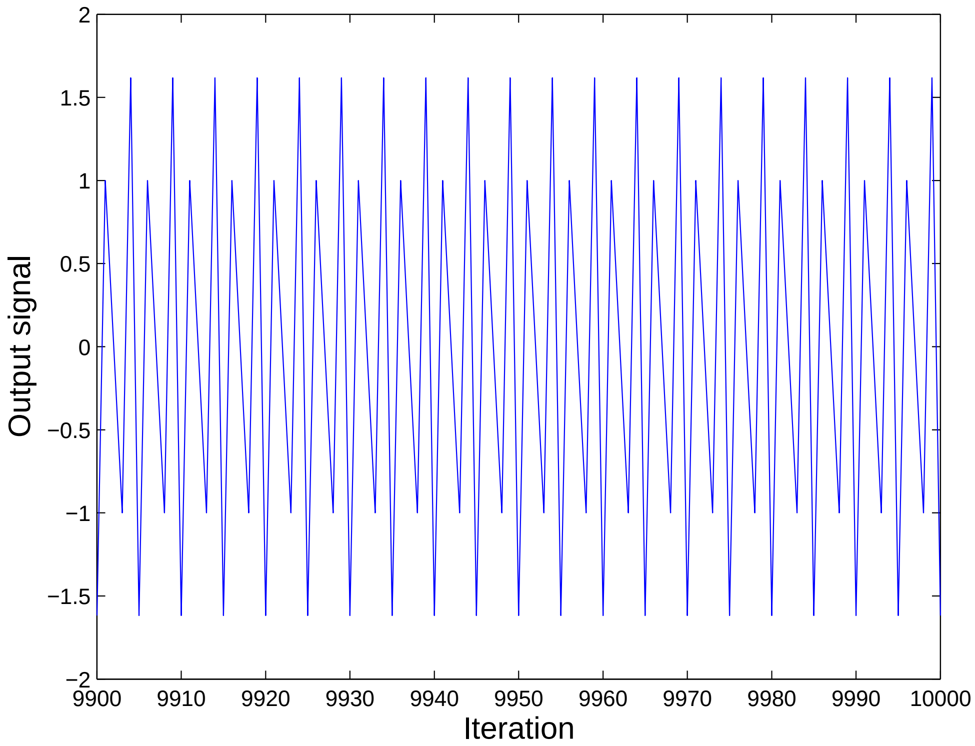


Figure 29: Example of a digital oscillator output.

Comb filter

- The comb filter is characterized by a magnitude response with multiple identical passbands. This device is a very useful building block for digital signal processing, finding applications in instrument synthesis in audio and harmonics removal (including DC), among others.
- The main task of a comb filter is to place equally spaced zeros on the unit circle, as illustrated in the following example.

Example 4.7

- For the first-order network seen in Figure 30:
  (a) Determine the corresponding transfer function.
  (b) Replace z^{-1} by z^{-L} and show its transposed version.
  (c) For the network obtained in item (b), determine the pole-zero constellation when $L = 8$ and $a = 0.5$, plotting the resulting frequency response.

Example 4.7

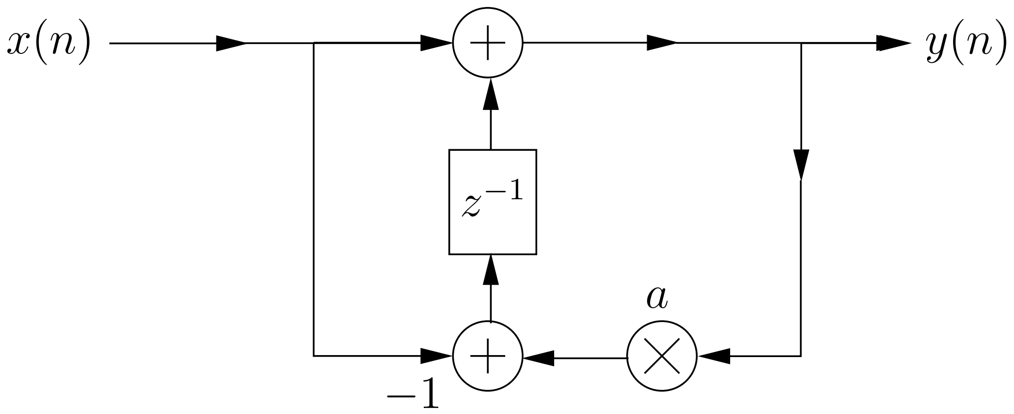


Figure 30: Comb filter structure in Example 4.7.

Example 4.7 - Solution

(a) The transfer function of the first-order comb filter is

$$H(z) = \frac{1 - z^{-1}}{1 - az^{-1}} \tag{109}$$

where is has a zero at $z = 1$ and a real pole at $z = a$.

(b) The transposed realization is depicted in Figure 31, with z^{-1} replaced by z^{-L}. The corresponding transfer function is given by

$$H(z) = \frac{1 - z^{-L}}{1 - az^{-L}} \tag{110}$$

Example 4.7 - Solution

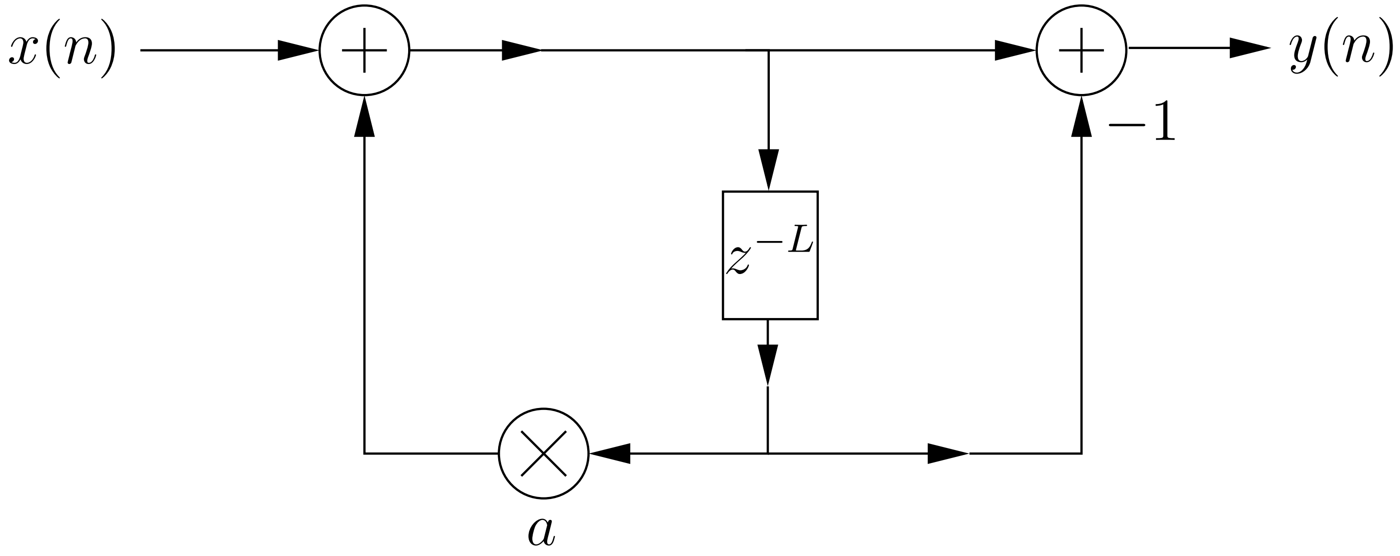


Figure 31: Transposed comb filter in Example 4.7.

Example 4.7 - Solution

(c) The pole-zero constellation associated to equation (110) consists of L equally spaced zeros on the unit circle placed at $z = e^{\frac{2\pi}{L}}$ with L poles placed at the same angles but on a circle with radius $a^{\frac{1}{L}}$. For $L = 8$ the pole-zero constellation of the comb filter is depicted in Figure 32.

- Figures 33a and 33b show the magnitude and phase responses, respectively, of the comb filter, where it can be observed the effects of the equally spaced zeros.
- The transitions between the peaks and valleys of the magnitude response are related to the value of a. The closer the value of a is to one, a sharper magnitude response and a more nonlinear phase result.
- In particular, for $a = 0$, the comb filter becomes a linear-phase FIR filter. This effect is further explored in an end-of-chapter exercise.

Example 4.7 - Solution

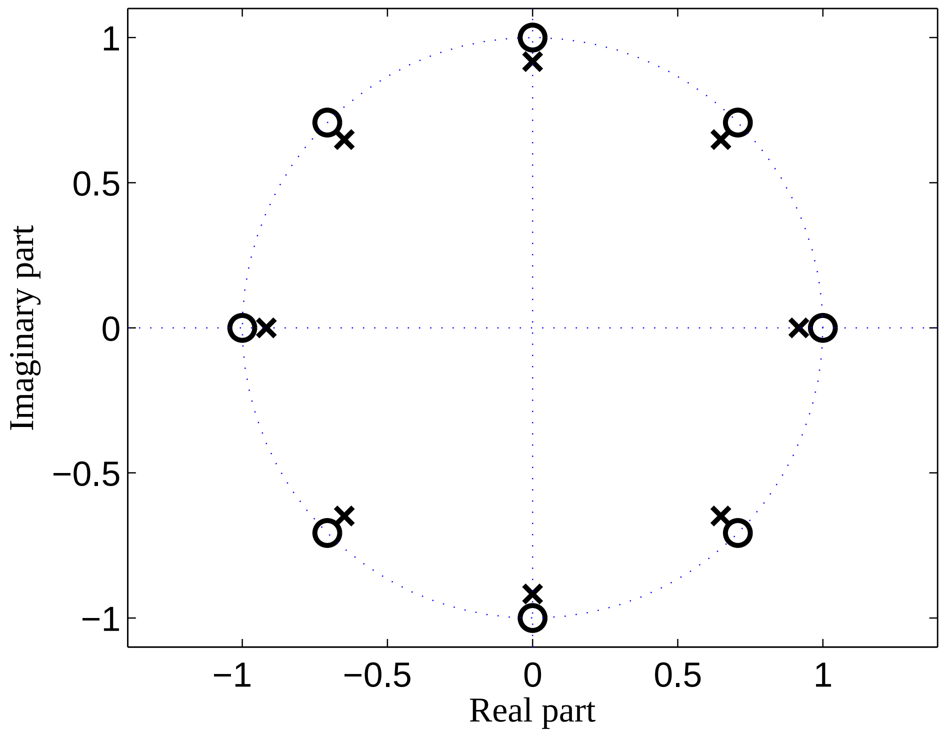
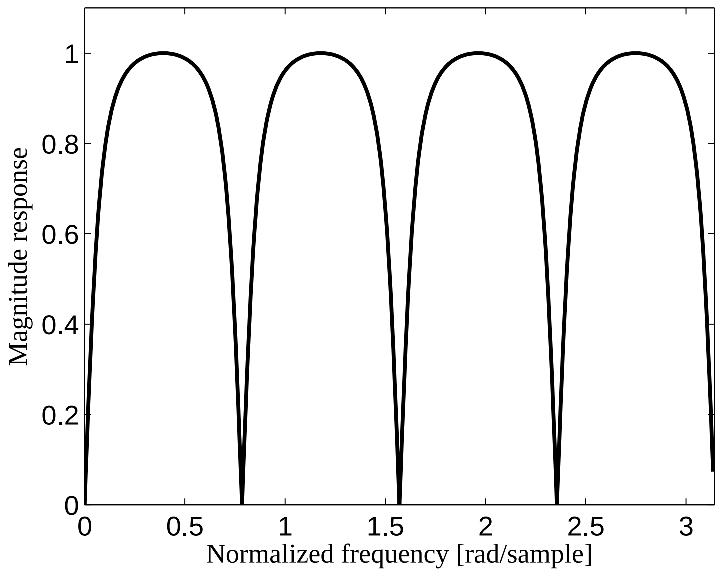
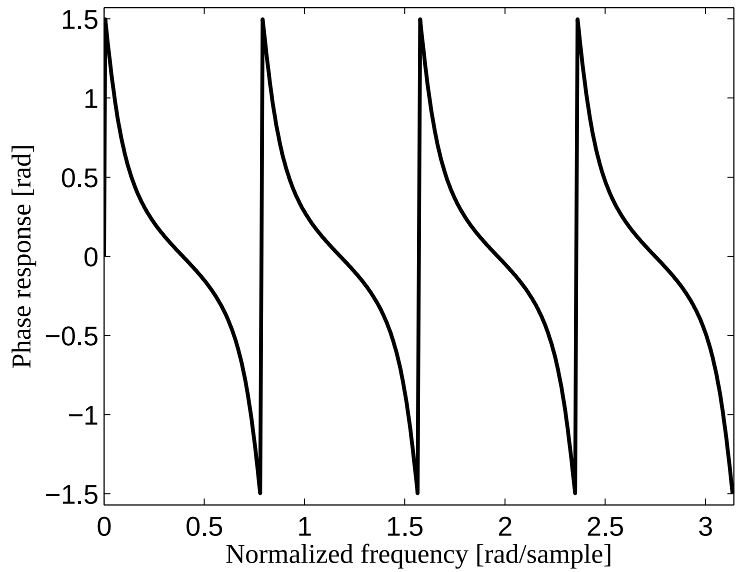


Figure 32: Pole-zero constellation of comb filter with $L = 8$ in Example 4.7.

Example 4.7 - Solution



(a)



(b)

Figure 33: Frequency response of the normalized comb filter in Example 4.7: (a) magnitude response; (b) phase response.

Do-it-yourself: Digital filters

- **Experiment 4.1:** Consider the direct-form transfer function

$$H(z) = \frac{z^6 + z^5 + z^4 + z^3 + z^2 + z + 1}{z^6 + 3z^5 + \frac{121}{30}z^4 + \frac{92}{30}z^3 + \frac{41}{30}z^2 + \frac{1}{3}z + \frac{1}{30}} \tag{111}$$

- One can easily find the poles and zeros of such a function in MATLAB using the command lines:

```
num = [1 1 1 1 1 1 1];
den = [1 3 121/30 92/30 41/30 1/3 1/30];
[zc,pc,kc] = tf2zp(num,den);
```

leading to the results shown in Table 2.

Do-it-yourself: Digital filters

Table 2: Zeros and poles of transfer function $\mathrm{H}(z)$ in Experiment 4.1.

Zeros	Poles
$0.6235 + 0.7818\mathrm{j}$	$-0.5000 + 0.5000\mathrm{j}$
$0.6235 - 0.7818\mathrm{j}$	$-0.5000 - 0.5000\mathrm{j}$
$-0.9010 + 0.4339\mathrm{j}$	-0.7236
$-0.9010 - 0.4339\mathrm{j}$	$-0.5000 + 0.2887\mathrm{j}$
$-0.2225 + 0.9749\mathrm{j}$	$-0.5000 - 0.2887\mathrm{j}$
$-0.2225 - 0.9749\mathrm{j}$	-0.2764

Do-it-yourself: Digital filters

- In general, the `tf2zp` command groups the complex poles and zeros in conjugate pairs. However, as one can see in the pole column of Table 2, these complex-conjugate pairs must be sorted out from the real roots to compose the second-order blocks of a cascade realization with real coefficients.

- This is automatically done by the `zp2sos` command, whose usage is exemplified below:

```
Hcascade = zp2sos(zc,pc,kc)
```

yielding

```
Hcascade =
 1.0000  -1.2470  1.0000  1.0000  1.0000  0.2000
 1.0000   0.4450  1.0000  1.0000  1.0000  0.3333
 1.0000   1.8019  1.0000  1.0000  1.0000  0.5000
```

Do-it-yourself: Digital filters

- This result corresponds to the realization

$$H(z) = \frac{z^2 - 1.2470z + 1}{z^2 + z + \frac{1}{5}} \frac{z^2 + 0.4450z + 1}{z^2 + z + \frac{1}{3}} \frac{z^2 + 1.8019z + 1}{z^2 + z + \frac{1}{2}} \qquad (112)$$

- The parallel realization of a given transfer function can be determined with the aid of the `residue` command that expresses $H(z)$ as the sum

$$H(z) = \frac{r_1}{z - p_1} + \frac{r_2}{z - p_2} + \cdots + \frac{r_N}{z - p_N} + k \qquad (113)$$

where N is the number of poles, and the parameters r_i, p_i, and k are the outputs of

```
[rp,pp,kp] = residue(num,den);
```

- Once again, one needs to sort out the pairs of complex-conjugate poles from the real ones in `pp` to determine the second-order parallel blocks with strictly real coefficients. This time, however, we can not rely on the `zp2sos` command, which is suitable only for the cascade decomposition.

Do-it-yourself: Digital filters

- The solution is to employ the `cplxpair` command, that places the real roots after all complex pairs, and rearrange the residue vector `rp` accordingly, to allow the proper combination of residues and poles to form the second-order terms, as in the script below:

```
N = length(pp);
pp2 = cplxpair(pp);
rp2 = zeros(N,1);
for i = 1:N,
  rp2(find(pp2 == pp(i)),1) = rp(i);
end;
num_blocks = ceil(N/2);
Hparallel = zeros(num_blocks,6);
```

Do-it-yourself: Digital filters

```
for count_p = 1:num_blocks,
  if length(pp2) ~= 1,
    Hparallel(count_p,2) = rp2(1)+rp2(2);
    Hparallel(count_p,3) =
-rp2(1)*pp2(2)-rp2(2)*pp2(1);
    Hparallel(count_p,5) = -pp2(1)-pp2(2);
    Hparallel(count_p,6) = pp2(1)*pp2(2);
    rp2(1:2) = []; pp2(1:2) = [];
  else,
    Hparallel(count_p,2) = rp2(1);
    Hparallel(count_p,5) = -pp2(1);
  end;
Hparallel(count_p,4) = 1;
end;
Hparallel = real(Hparallel);
```

Do-it-yourself: Digital filters

- This script, for this experiment, yields

```
Hparallel =

 0    10    17.5000  1  1  0.5000
 0   -20   -38.3333  1  1  0.3333
 0     8    21.8000  1  1  0.2000
```

which, taking `kp = 1` also into consideration, corresponds to the parallel decomposition

$$H(z) = \frac{10z + 17.5}{z^2 + z + \frac{1}{2}} - \frac{20z + 38.3333}{z^2 + z + \frac{1}{3}} + \frac{8z + 21.8}{z^2 + z + \frac{1}{5}} + 1 \tag{114}$$

Do-it-yourself: Digital filters

- A state-space description corresponding to a given transfer function is easily determined in MATLAB with the command

```
[A,B,C,D] = tf2ss(num,den);
```

which, for $\mathrm{H}(z)$ as given in equation (111), results in

$$\mathsf{A} = \begin{bmatrix} -3 & -\frac{121}{30} & -\frac{92}{30} & -\frac{41}{30} & -\frac{1}{3} & -\frac{1}{30} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}; \ \mathsf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};$$

$$\mathsf{C} = \begin{bmatrix} -2 & -\frac{91}{30} & -\frac{62}{30} & -\frac{11}{30} & \frac{2}{3} & \frac{29}{30} \end{bmatrix}; \ \mathsf{D} = 1 \qquad (115)$$