EE2S11 Signals and Systems

Chapter 12.6 Discrete-time filter structures (realizations)

Alle-Jan van der Veen

–

19 January 2024

Contents

- FIR filters (direct form – tapped delay line)
- IIR filters (direct form 1, 2)
- Cascade and parallel structure
- Transposition

Minimal and canonical realizations

A structure which implements an N-th order transfer function is called minimal if it uses exactly N delay elements.

A canonical realization is a "textbook structure", the typical structure for a certain class of transfer functions (e.g. FIR, IIR, allpass, $\cdots$). It is usually minimal, with also a minimal number of operations (multiplications with coefficients).

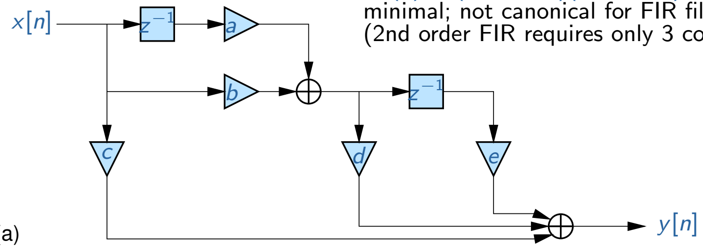
Generally, each filter coefficient should appear only *once* in the realization. This is important for zero-phase FIR filters,

$$H(z) = b_0 + b_1 z^{-1} + b_1 z^{-2} + b_0 z^{-3}$$

and allpass filters,

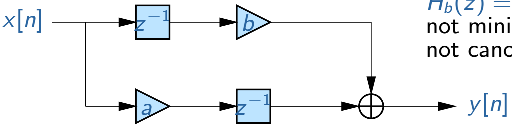
$$H(z) = \frac{a_2 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Examples



$H_a(z) = (z^{-1}a + b)(d + z^{-1}e) + c$
minimal; not canonical for FIR filters
(2nd order FIR requires only 3 coeff.)

(a)

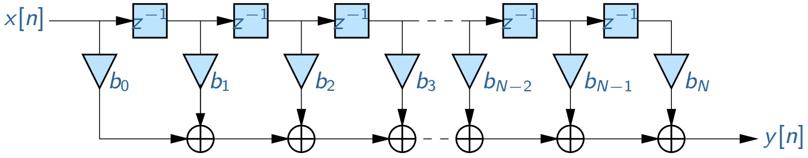


$H_b(z) = z^{-1}(a + b)$
not minimal;
not canonical for FIR filters

(b)

Transversal filter

An FIR filter can be realized using a *transversal filter*:



$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$

- Minimal and canonical for the class of N-th order FIR filters: N delays; $N+1$ multipliers for $N+1$ coefficients
- The coefficients $h[n] = b_n$ are directly used in the realization
- The transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}$$

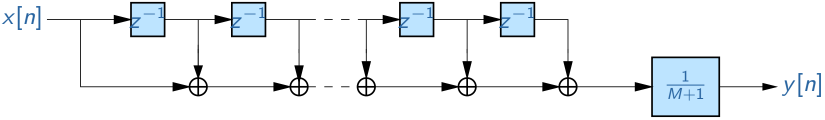
Recursive implementation of an FIR filter

An FIR filter can sometimes also be implemented recursively: e.g.,

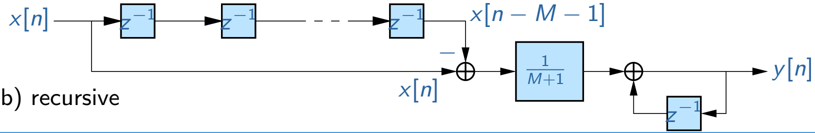
$$y[n] = \frac{1}{M+1}\sum_{k=0}^{M} x[n-k] = \frac{1}{M+1}\left(x[n] + x[n-1] + \cdots + x[n-M]\right)$$

can be written as

$$y[n] = y[n-1] \quad + \quad \frac{1}{M+1}\left(x[n] - x[n-M-1]\right)$$



(a) non-recursive

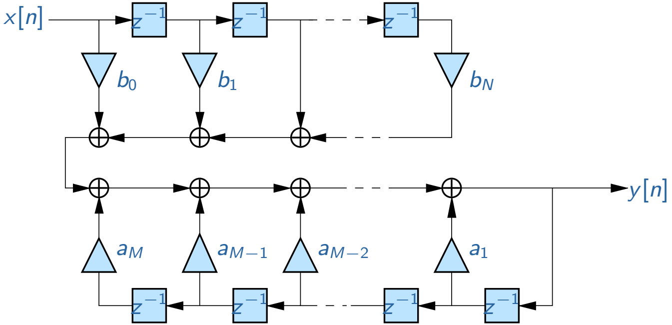


(b) recursive

Recursive filter: direct form no. 1

Realization for a general rational filter (IIR, $a_0 = 1$):

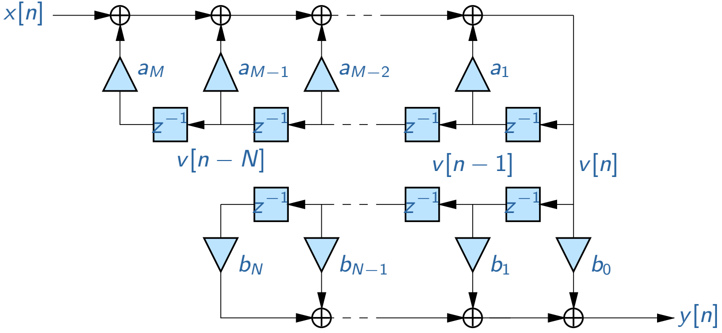
$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 - a_1 z^{-1} - \cdots - a_M z^{-M}} \quad \Leftrightarrow \quad \begin{aligned} y[n] &= b_0 x[n] + \cdots + b_N x[n-N] \\ &+ a_1 y[n-1] + \cdots + a_M y[n-M] \end{aligned}$$



This is not a minimal structure: $M + N$ delays instead of $\max(M, N)$.

Recursive filter: direct form no. 2

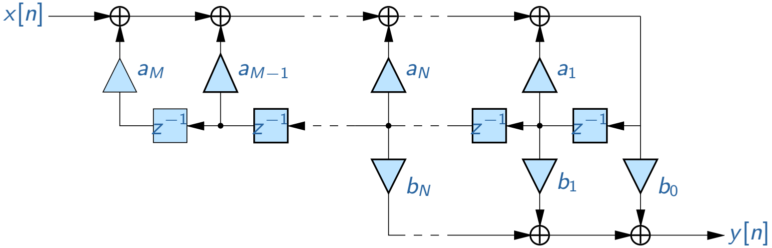
Use the commutative property of the convolution: $h_1 * h_2 = h_2 * h_1$.
We may reverse the order of both partial systems.



It is seen that the delay lines can be merged (they transport the same signal $v[n]$)

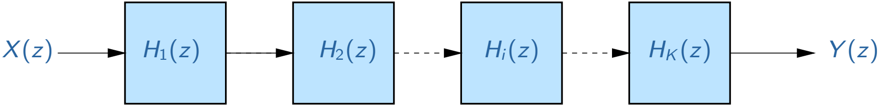
Recursive filter: direct form no. 2 (cont'd)

The resulting filter (minimal and canonical):



- Also in this realization the filter coefficients are directly related to the parameters in the difference equation.
- This realization is very sensitive to small disturbances (quantization) of the coefficients: the poles/zeros can move a lot. [See EE2S31]

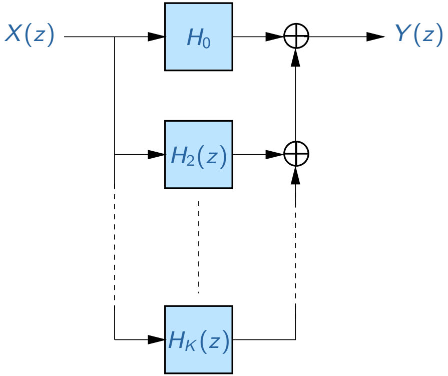
Cascade structure



$$H(z) = H_1(z) \cdot H_2(z) \cdots H_K(z), \quad \text{e.g. } H_k(z) = G_k \frac{(1 - z_k z^{-1})(1 - z_k^* z^{-1})}{(1 - p_k z^{-1})(1 - p_k^* z^{-1})}$$

- Usually second order sections: less sensitive.
  Second order sections are needed for a canonical realization of transfer functions with *real-valued* coefficients.
- Used if the $H_k(z)$ all have the same passband (otherwise, large gains are needed).

Parallel structure



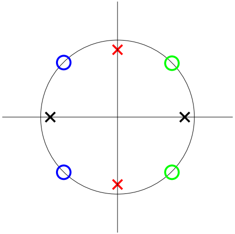
$H(z) =$
$H_1(z)+H_2(z)+\cdots+H_K(z)$
e.g. $H_k(z) =$
$\dfrac{A_k}{1-p_k z^{-1}} + \dfrac{A_k^*}{1-p_k^* z^{-1}}$
(2nd order section for complex conj. poles)

- Less control over the location of zeros.

Example

$$H(z) = G\frac{(1-e^{j\pi/4}z^{-1})(1-e^{-j\pi/4}z^{-1})(1-e^{j3\pi/4}z^{-1})(1-e^{-j3\pi/4}z^{-1})}{(1-0.9z^{-1})(1+0.9z^{-1})(1-0.9j\,z^{-1})(1+0.9j\,z^{-1})}$$

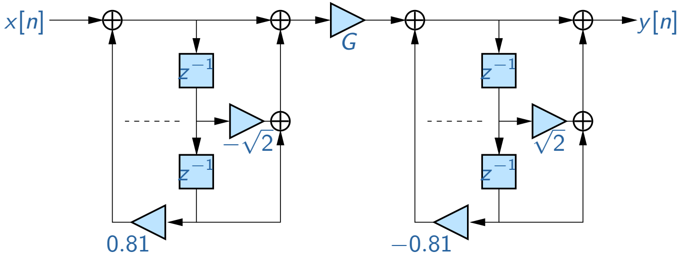
There are several possibilities to split this into 2nd order sections with real-valued coefficients. For a cascade, we can also choose which pair of zeros we combine with which pair of poles. With infinite accuracy (no quantization) this does not make a difference.



Example (cont'd)

$$H_1(z) = \frac{(1-e^{j\pi/4}z^{-1})(1-e^{-j\pi/4}z^{-1})}{(1-0.9z^{-1})(1+0.9z^{-1})} = \frac{1-\sqrt{2}z^{-1}+z^{-2}}{1-0.81z^{-2}}$$

$$H_2(z) = \frac{(1-e^{j3\pi/4}z^{-1})(1-e^{-3j\pi/4}z^{-1})}{(1-0.9j\,z^{-1})(1+0.9j\,z^{-1})} = \frac{1+\sqrt{2}z^{-1}+z^{-2}}{1+0.81z^{-2}}$$



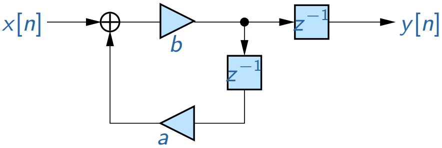
Transposition

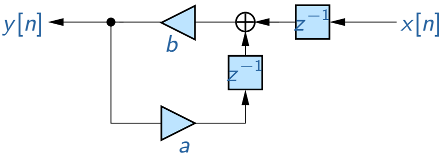
Proposition: Given a realization (graph/network with nodes and edges). Make the following changes:

1. Reverse the direction of every edge (adders $\leftrightarrow$ nodes)
2. Reverse input and output

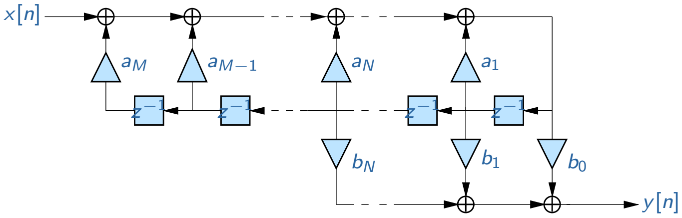
The transfer function is *not changed* (cf. Tellegen's theorem).

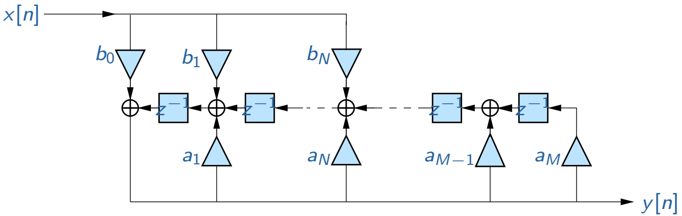
Example: $H(z) = \dfrac{bz^{-1}}{1 - abz^{-1}}$





Application to direct form no. 2





Advantage: a much shorter critical path (all adders can operate in parallel).