Structures for Discrete-Time Systems

EE4015 Digital Signal Processing

Dr. Lai-Man Po

Department of Electrical Engineering

City University of Hong Kong

EE4015 Face-to-Face Mid-Term Exam

- The Face-to-Face Mid-Term Exam will be held on November 8, 2022 (Tuesday of Week 11).
- The exam time is 2 hours. Students should arrive at the venue at least 5 minutes before the start of the exam.
- The Mid-Term Exam is an open-note exam. Students can use "Scientific Calculator" and "All Handouts", including exercises and assignments.
  - In addition to hard copies of handouts, students can also use smartphones, tablets or iPads to read notes, but the electronic device must be set to airplane mode. During the exam, you are not allowed to communicate with others and search on the Internet. During the exam, investigators will check from time to time whether your electronic device is in airplane mode.
- Students need to use their own answer sheets (such as A4 paper) to answer the questions.
- The mid-term exam will cover up to week 8.

Content

Structures for Discrete-Time Systems

- In practical, we need to use different structures to realize discrete-time systems in hardware or software.
- Discrete-time LTI system with rational transfer function $H(z)$ can be represented as :

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

- Or the corresponding linear constant coefficient difference equation

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

where x[n] and y[n] are the system input and output

Recursive and Non-Recursive Structures

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

Feed Forward Feedback

There are two types of structures for realization of Discrete-Time Systems

- **Non-recursive** : No feedback paths $(a_k = 0)$
  - It is always used for implementation of Finite Impulse Response (FIR) systems.
- **Recursive** : At least one feedback path $(a_k \neq 0)$
  - It is commonly used for implementation of Infinite Impulse Response (IIR) systems, but FIR systems are also possible to be implemented by recursive structure.

Implementation of Difference Equations

- The implementation of difference equations requires delayed values of the
  - input
  - output
  - intermediate results
- Computing y[n] involves y[n-1], y[n-2], …, y[n-N], and x[n], x[n-1],…,x[n-M]. That is, we need
  - Delay elements or storage
  - Multipliers
  - Adders (subtraction is considered as addition)

Complexity of the Discrete-Time Systems

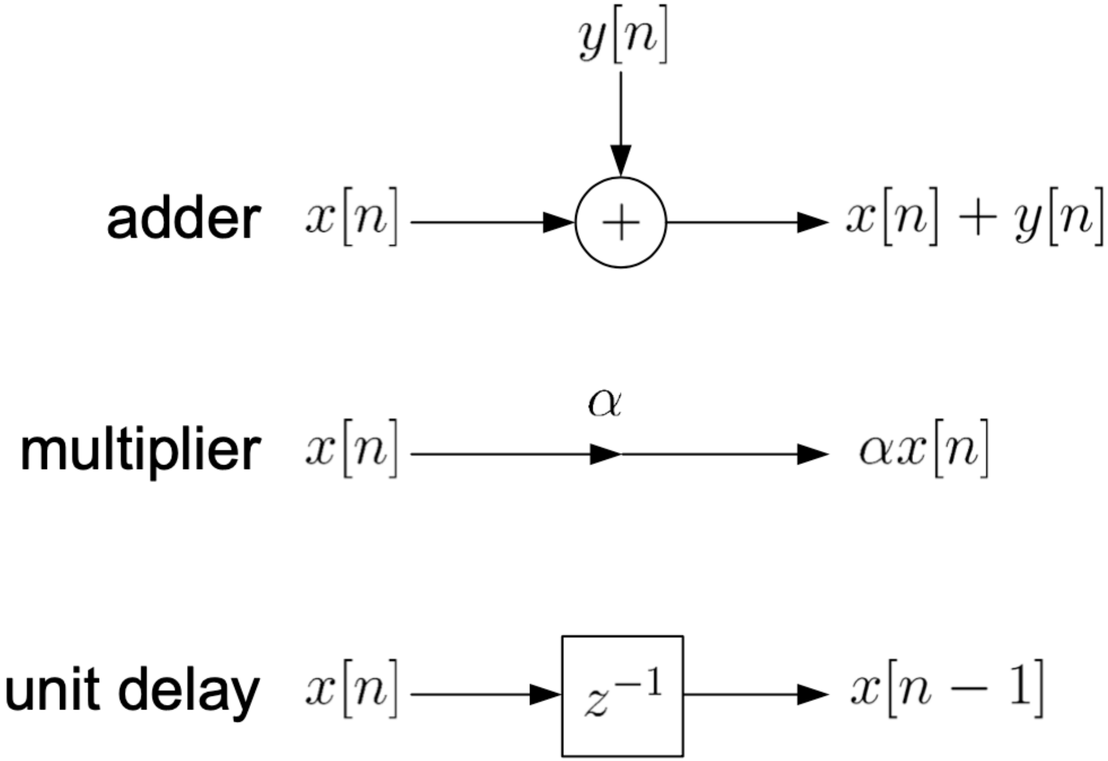
How many storage elements are needed?

How many multipliers are needed?

How many adders are needed?

- Computations of y[n] can be arranged in different ways to give the same difference equation, which leads to different structures for realization of discrete-time LTI systems
- **4 basic forms** of implementations : **Direct Form**, **Canonic Form**, Cascade Form and Parallel Form
- An implementation can be represented using either a **Block Diagram** or a **Signal Flow graph**

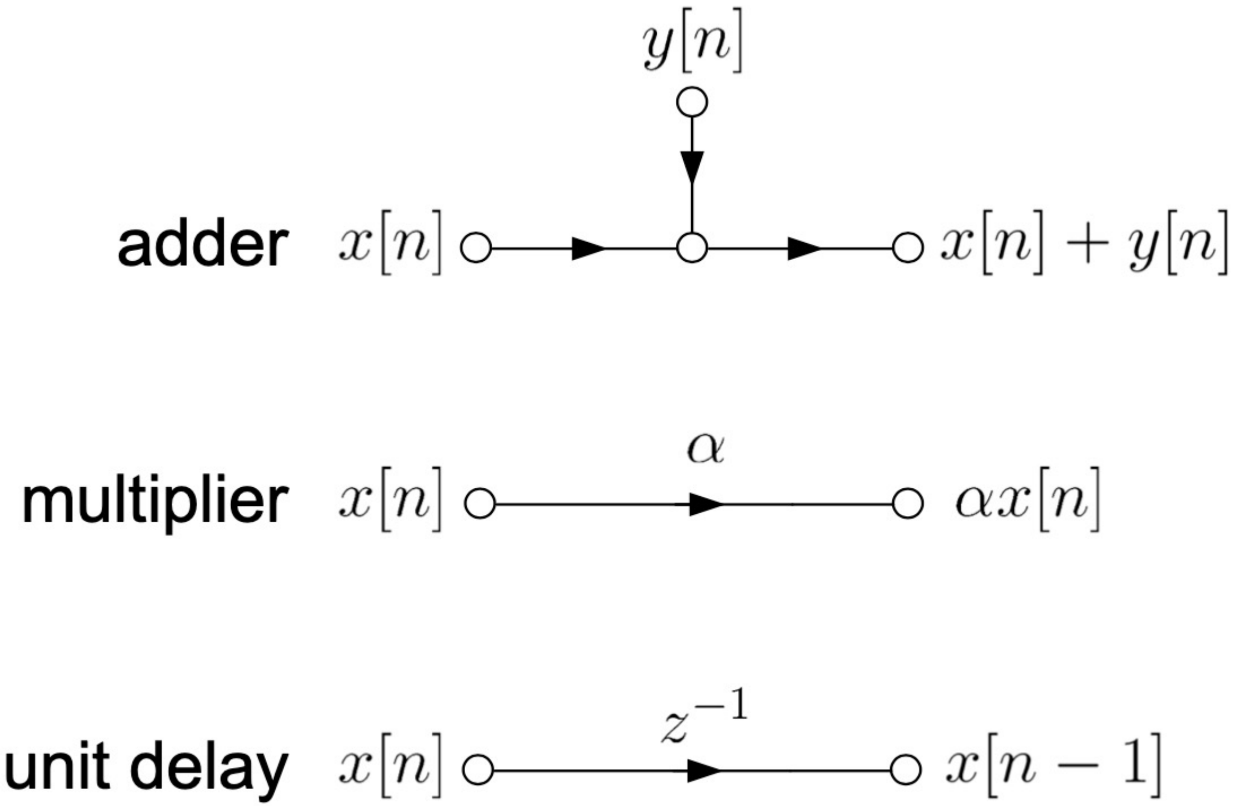
Block Diagram Representation



Features of these Basic Operations

- Although an **adder** can generally deal with more than two sequences, here we consider two signals in order to align with practical implementation in microprocessors.
- When $|\alpha| > 1$, it corresponds to signal amplification while the signal is attenuated for $|\alpha| < 1$. Note that a **multiplier** usually has the highest implementation or computational cost and thus it is desired to reduce the number of multipliers in different systems.
- The transfer function z^{-1} corresponds to a unit delay. It can be implemented by providing a storage register for each unit delay in digital implementation. If the required number of samples of delay is $D > 1$, then the corresponding system function is z^{-D}.

Signal Flow Graph Representation



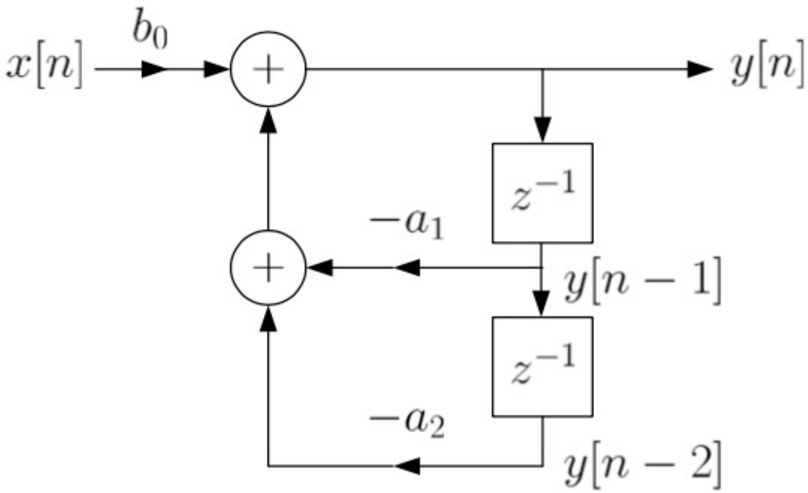
Features of Signal Flow Graph

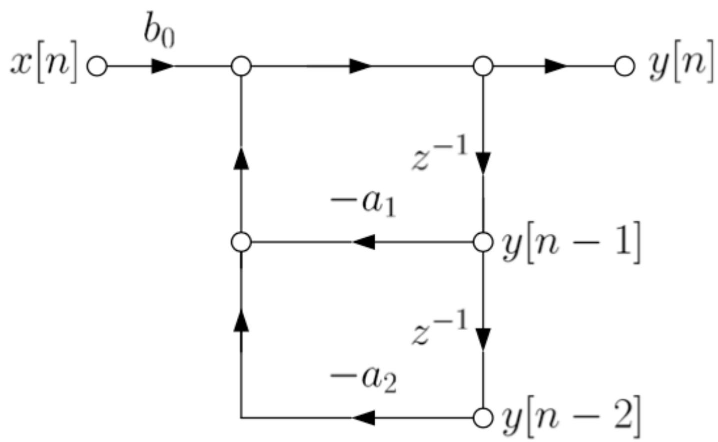
- Its basic elements are branches with directions, and nodes. That is, a signal flow graph is a set of directed branches that connect at nodes.
- Signal at a node of a flow graph is equal to the sum of the signals from all branches connecting to the node.
- Signal out of a branch is equal to the branch gain times the signal into the branch.
- Branch gain can refer to a scalar or a transfer function of z^{-1} corresponding to multiplication or unit delay operation, respectively.
- When the branch gain is unity, it is left unlabeled.
- A signal flow graph provides an alternative but equivalent graphical representation to a block diagram structure

Block Diagram and Signal Flow Graph Examples

- Draw the block diagram and signal flow graph representations of a LTI system whose input and output y[n] satisfy the following difference equation:

$$\boxed{y[n] = -a_1 y[n-1] - a_2 y[n-2] + b_0 x[n]}$$





2 adders, 3 multipliers and 2 delay elements are required to implement the system.

Structure for Non-Recursive DT Systems

- Non-recursive structures do not have feedback paths and commonly used for realization of Finite Impulse Response (FIR) Digital Filters.
- For FIR filter, its transfer function does not contain pole. That is, setting $a_0 = 1$ and $a_1 = a_2 = \cdots = a_N = 0$ in the general difference equation yields a FIR system:

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

- While the corresponding difference equation is:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

Direct Form

- Basically, the difference equation coefficients of the FIR DT systems corresponding to the impulse response h[n]:

$$h[n] = \begin{cases} b_n & 0 \le n \le M \\ 0 & otherwise \end{cases}$$

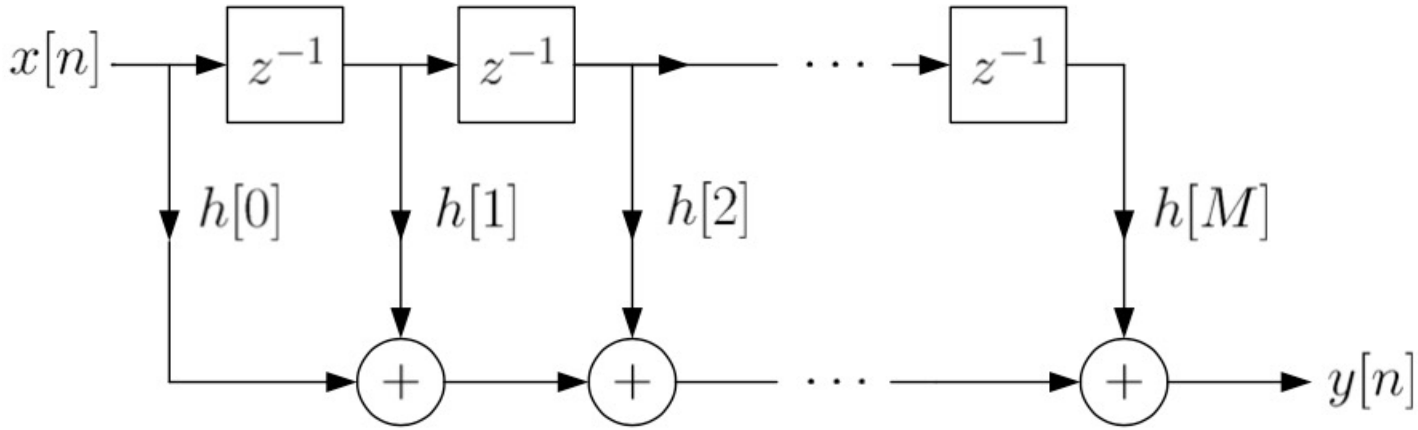
- Then, the difference equation can be written as

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k] = h[0]x[n] + h[1]x[n-1] + \cdots + h[M]x[n-M]$$

- The direct form follows straightforwardly form the difference equation.

Direct Form of FIR Systems

- The implementation needs memory locations for storing M previous inputs of $x[n]$, $M+1$ multiplications and M additions for computing each output value of $y[n]$.



Cascade Form

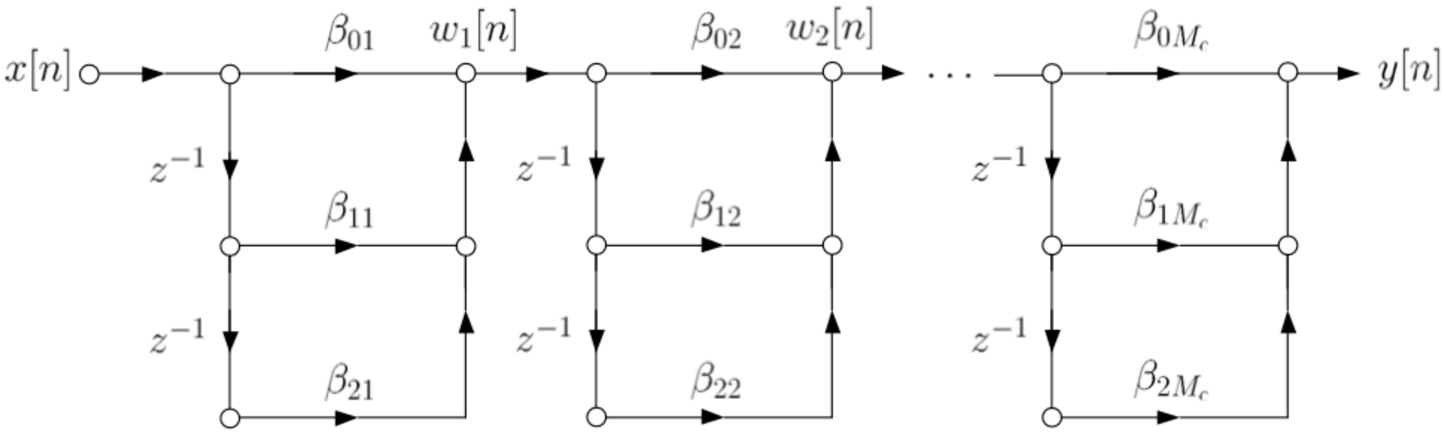
- The transfer function of FIR system can also be expressed as products of second-order polynomial system functions via factorization:

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \prod_{k=1}^{M_c} \left(\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}\right)$$

- Where $M_c = \lfloor (M+1)/2 \rfloor$ is the largest integer contained in $(M+1)/2$.
- Note that when M is odd, one of the $\{\beta_{2k}\}$ will be zero.
- Assuming that M is even, this implementation needs M storage elements, $3M/2$ multiplications and M additions, for computing each output value of $y[n]$.

Why second-order polynomial instead of first-order polynomial?

Cascade Form of FIR System



$$w_1[n] = \beta_{01}x[n] + \beta_{11}x[n-1] + \beta_{21}x[n-2]$$

and

$$w_2[n] = \beta_{02}w_1[n] + \beta_{12}w_1[n-1] + \beta_{22}w_1[n-2]$$

Four Types of Causal Linear Phase FIR Systems

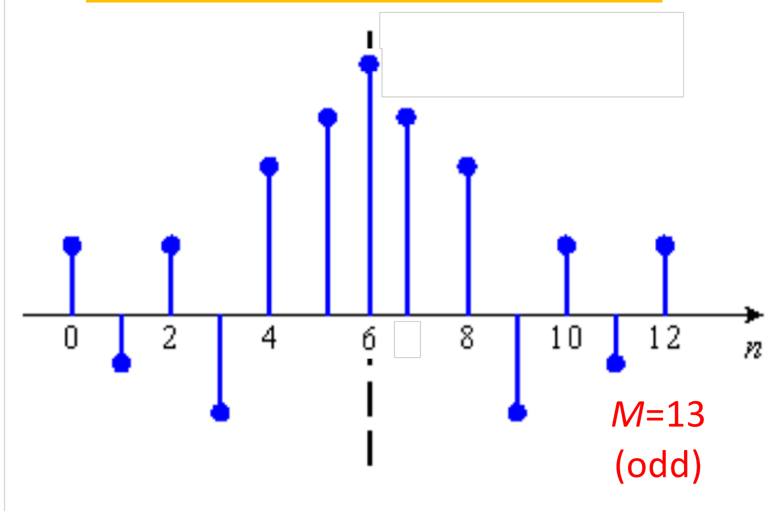
- For casual FIR systems, if their impulse response $h[n]$ satisfied the symmetrical property, then the systems will have linear phase responses.
- The symmetrical impulse response property is defined as

$$h[n] = \pm h[M-1-n], \qquad n = 0,1,\ldots,M-1$$

- **There 4 types of linear phase FIR systems**:
  - **Type I : Odd Positive Symmetric** – M is odd and $h[n] = h[M-1-n]$
  - **Type II : Even Positive Symmetric** – M is even and $h[n] = h[M-1-n]$
  - **Type III : Odd Negative Symmetric** – M is odd and $h[n] = -h[M-1-n]$
  - **Type IV : Even Negative Symmetric** – M is even and $h[n] = -h[M-1-n]$

Positive Symmetry Impulse Responses

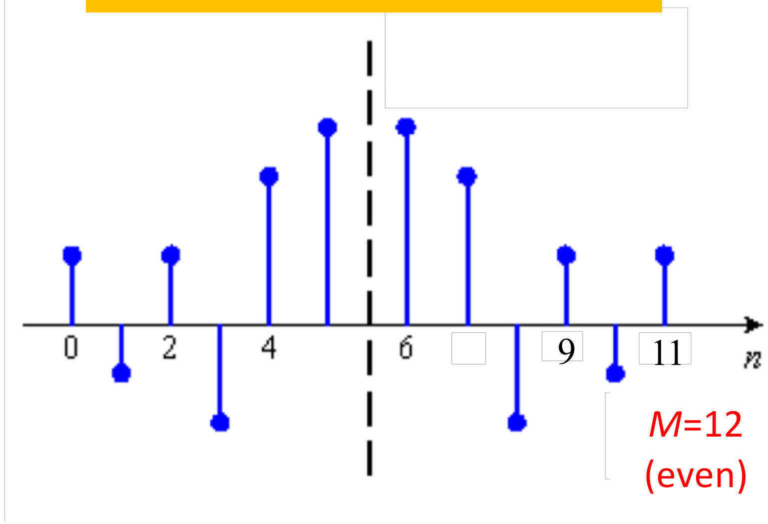
Odd Positive Symmetry (Type I)



$$h[n] = h[M-1-n]$$

$$H(e^{j\omega}) = e^{-j\left(\frac{\omega(M-1)}{2}\right)}\left(h\left[\frac{M-1}{2}\right] + 2\sum_{k=1}^{(M-3)/2} h\left[\frac{M-1}{2}-k\right]\cos k\omega\right)$$

Even Positive Symmetry (Type II)

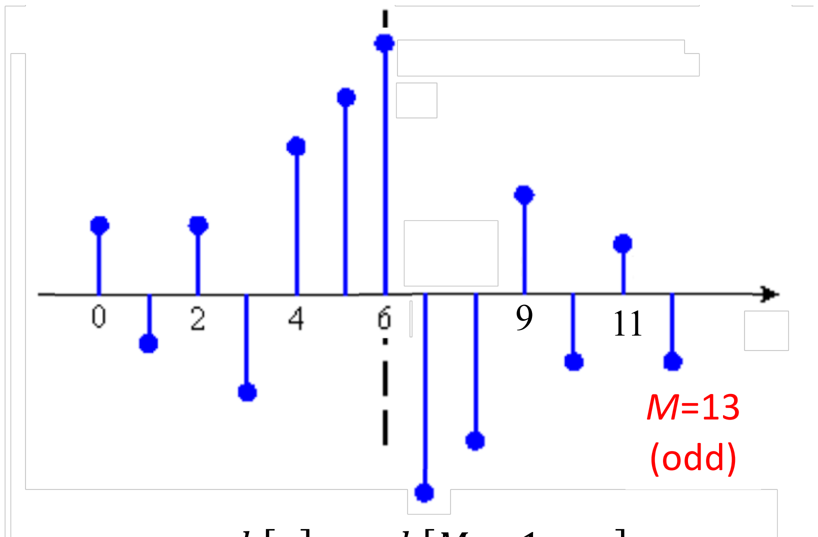


$$h[n] = h[M-1-n]$$

$$H(e^{j\omega}) = e^{-j\left(\frac{\omega(M-1)}{2}\right)}\left(2\sum_{k=1}^{(M-3)/2} h\left[\frac{M-1}{2}-k\right]\cos\left(\left(k-\frac{1}{2}\right)\omega\right)\right)$$

Negative Symmetry Impulse Responses

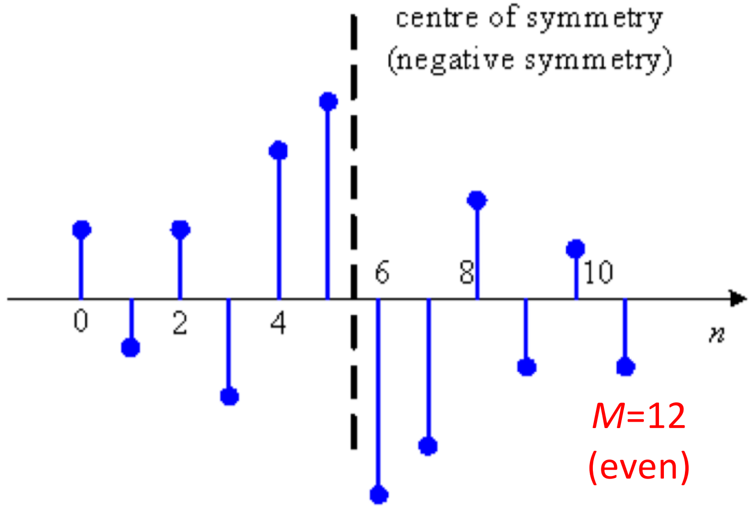
Odd Negative Symmetry (Type III)



$$h[n] = -h[M-1-n]$$

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

Even Negative Symmetry (Type IV)



$$h[n] = -h[M-n-1]$$

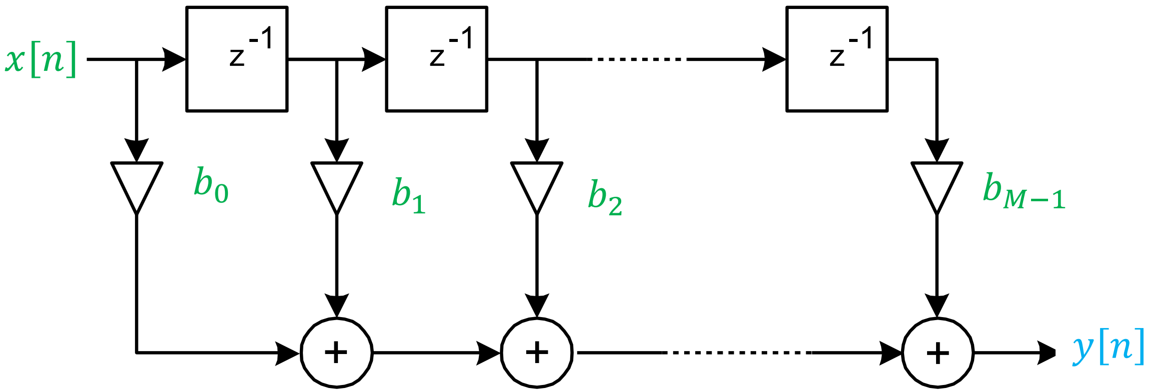
$$H(e^{j\omega}) = e^{-j\left(\frac{\omega(M-1)}{2}-\frac{\pi}{2}\right)}\left(2\sum_{k=1}^{(M-1)/2} h\left[\frac{M}{2}-k\right]\sin\left(\left(k-\frac{1}{2}\right)\omega\right)\right)$$

FIR Filter Structures

- Direct Form structure for an FIR filter:

$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_{M-1} z^{M-1} = \sum_{k=0}^{M-1} b_k z^{-k}$$

$$Y(z) = Y(z)X(z) \qquad y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_{M-1} x[n-M+1]$$



Direct Form Structure with Linear-Phase FIR Structures

- Direct form structure for an FIR filter:

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

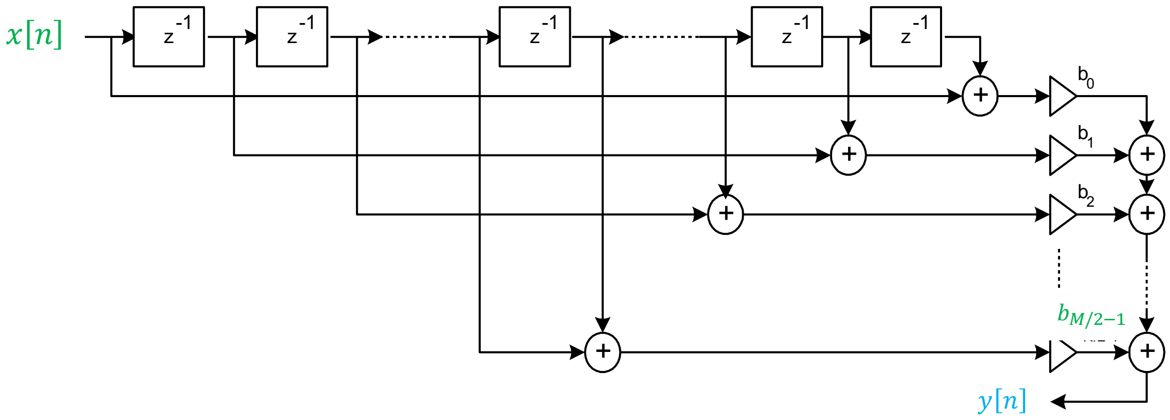
- **Linear-Phase** structures:
  - N even:

$$H(z) = \sum_{k=0}^{\frac{M}{2}-1} b_k\left(z^{-k} + z^{M-k-1}\right)$$

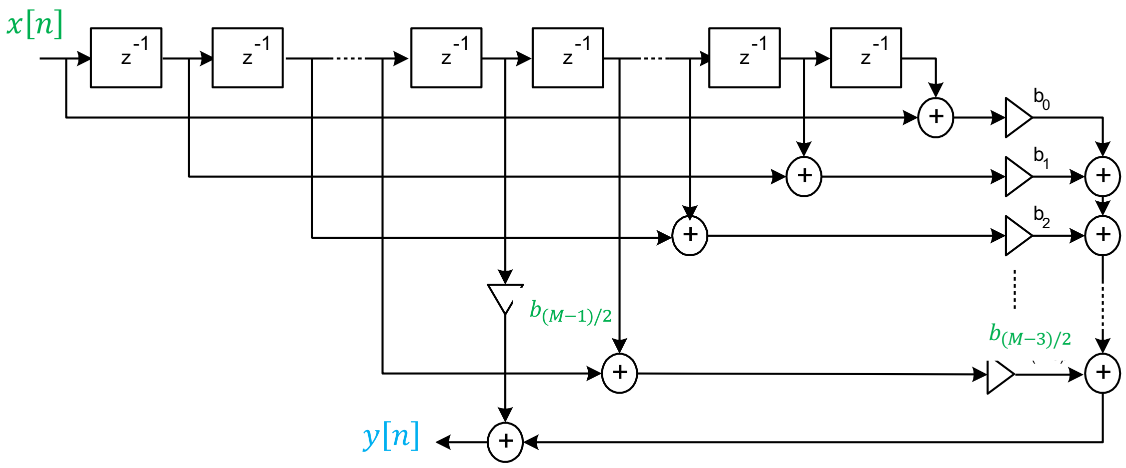
  - N Odd:

$$H(z) = \sum_{k=0}^{\frac{M-1}{2}} b_k\left(z^{-k} + z^{M-k-1}\right) + b_{\frac{M-1}{2}} z^{-\frac{M-1}{2}}$$

Linear-Phase FIR Filter Structures



(a) N even.



(b) N odd.

IIR Filter Structures

Structures for Recursive DT System

- **Infinite Impulse Response (IIR)** DT systems are always realized by recursive structures with at least one feedback paths.
- The corresponding transfer function of IIR system is

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

- Basically, it is the general form of any discrete-time LTI system.

Direct Form (or Direct Form I)

- Based on the general form of IIR transfer function, its difference equation can be decomposed into a pair of difference equations

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

- The direct form can also be obtained by decomposing H(z) into two transfer functions as

$$\boxed{H(z) = H_1(z) \cdot H_2(z)}$$

- where

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \quad \text{and} \quad H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

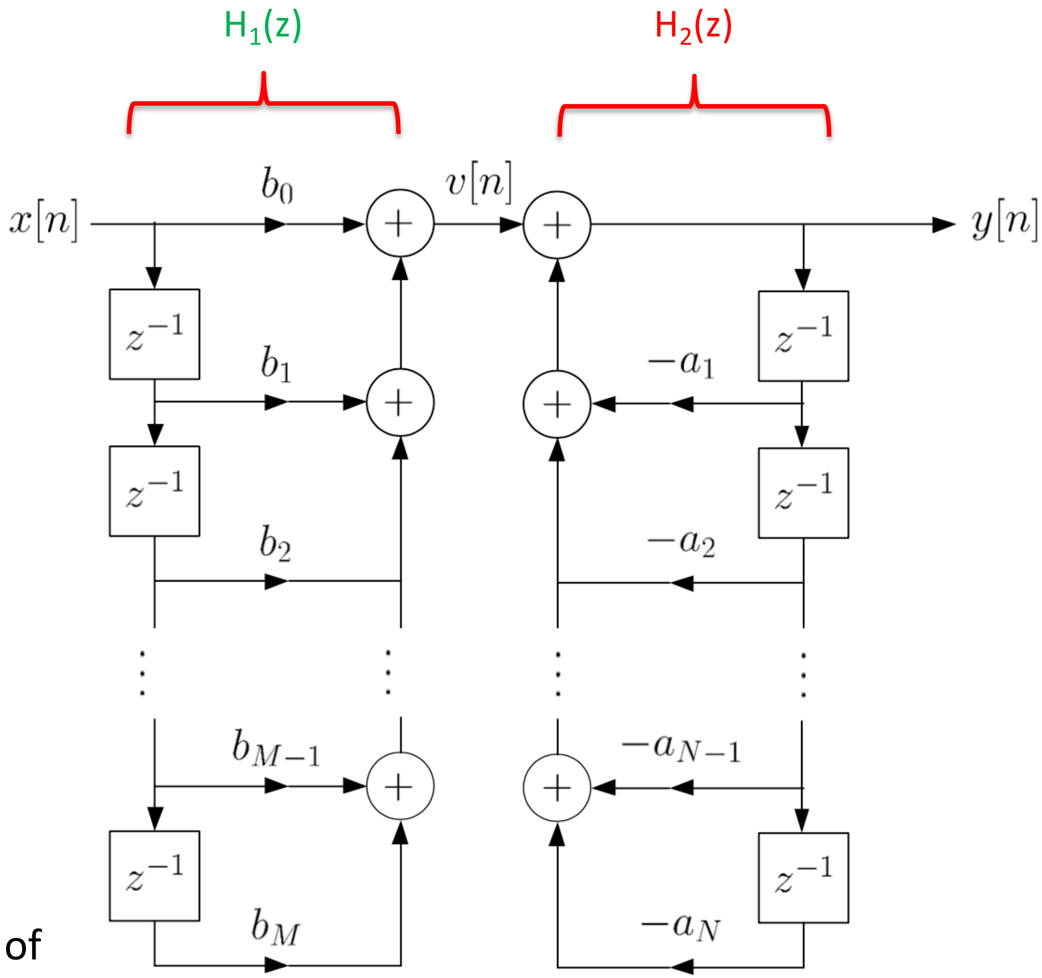
Direct Form

- In the z-transform domain, we have

$$V(z) = X(z)H_1(z) = X(z)\sum_{k=0}^{M} b_k z^{-k}$$

$$Y(z) = V(z)H_2(z) = \frac{V(z)}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

This direct form implementation needs (M+N) memory locations, (M+N+1) multiplications and (M+N) additions, for computing each output values of y[n].



Canonic Form (or Direct Form II)

- For the IIR LTI system, we can first pass x[n] through the filter $H_2(z)$ to produce an intermediate signal w[n]. The w[n] is then passed through the system $H_1(z)$ to give y[n]:

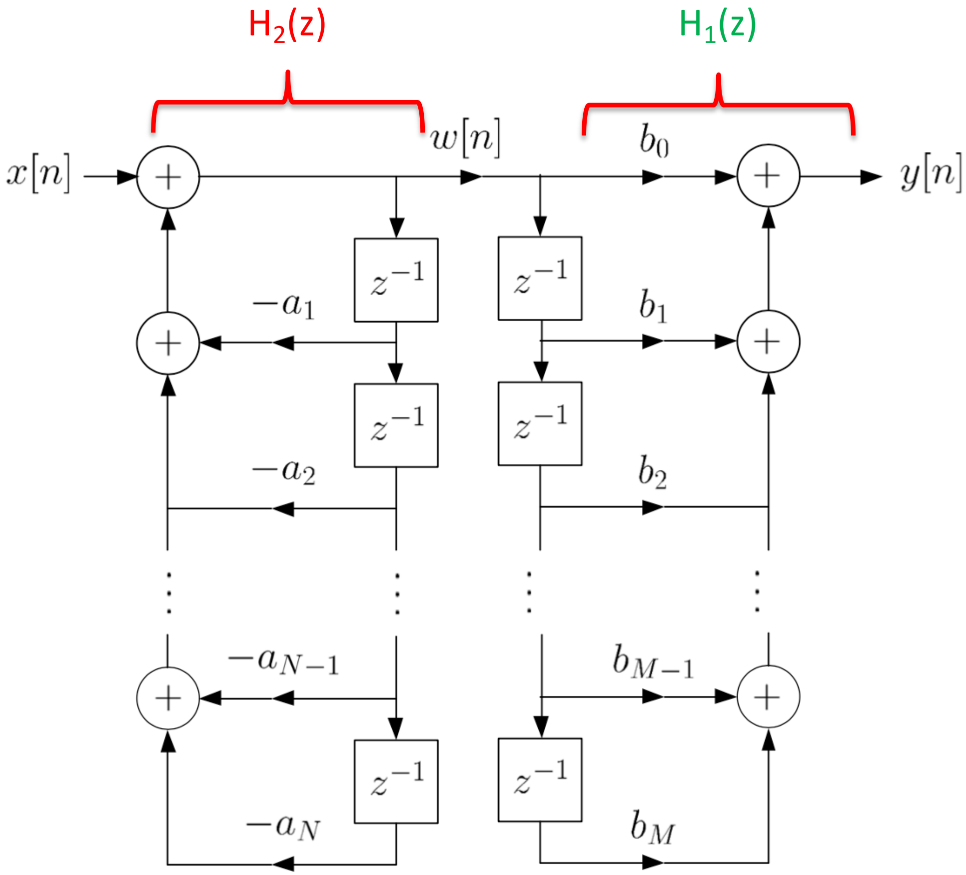
$$W(z) = X(z)H_2(z) = \frac{X(z)}{1 + \sum_{k=1}^{N} a_k z^{-k}} \quad \text{and} \quad Y(z) = W(z)H_1(z) = W(z)\sum_{k=0}^{M} b_k z^{-k}$$

- Applying inverse z-transform, we get:

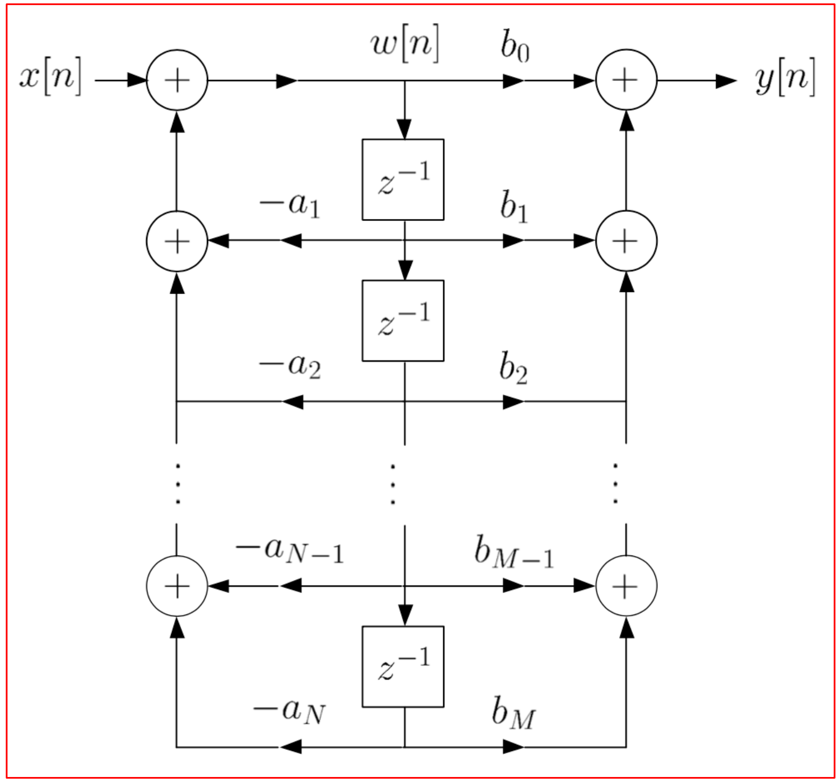
$$w[n] = -\sum_{k=1}^{N} a_k w[n-k] + x[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

- Which can be considered as an alternative direct form.

Canonic Form



Canonic form of IIR filter



This structure can save N memory locations.

Why Canonic Form?

- Assume M=N. Since the same signals w[n], w[n-1],..., w[n-N], are stored in the two chains of storage elements, they can be combined to reduce the memory requirement.
- In general, the minimum number of delay elements required is max(M,N).
- It is called canonic form because this implementation involves the minimum number of storages.

Example of Recursive Filter Structures

- Draw the block diagrams using the direct and canonic forms for the LTI system whose transfer function is

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

- Then, the difference equations
  - For direct form

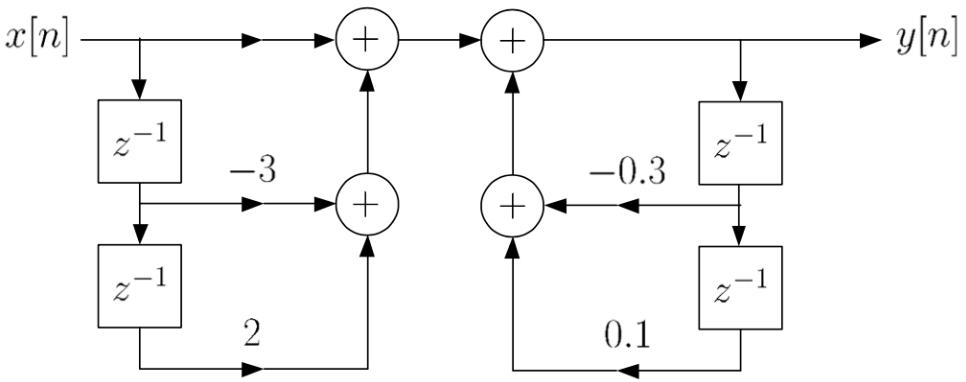
$$v[n] = x[n] - 3x[n-1] + 2x[n-2]$$
$$y[n] = -0.3y[n-1] + 0.1y[n-2] + v[n]$$

  - For canonic form

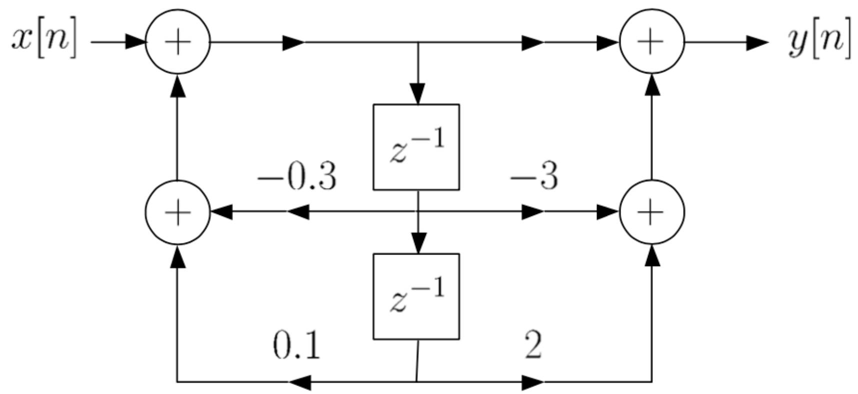
$$w[n] = -0.3w[n-1] + 0.1w[n-2] + x[n]$$
$$y[n] = w[n] - 3w[n-1] + 2w[n-2]$$

Examples of Direct Form and Canonic Form

Direct Form of Second-Order IIF Filter



Canonic Form of Second-Order IIF Filter



Cascade Form (1)

- We factorize the numerator and denominator polynomials in terms of second-order polynomial system functions as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \prod_{k=1}^{N_c} \frac{\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}}$$

- Without loss of generality, it is assumed that $N \geq M$ so that $N_c = \lfloor (N+1)/2 \rfloor$.
- Note that when M or N is odd, one of the $\{\beta_{2k}\}$ or $\{\alpha_{2k}\}$ will be zero.

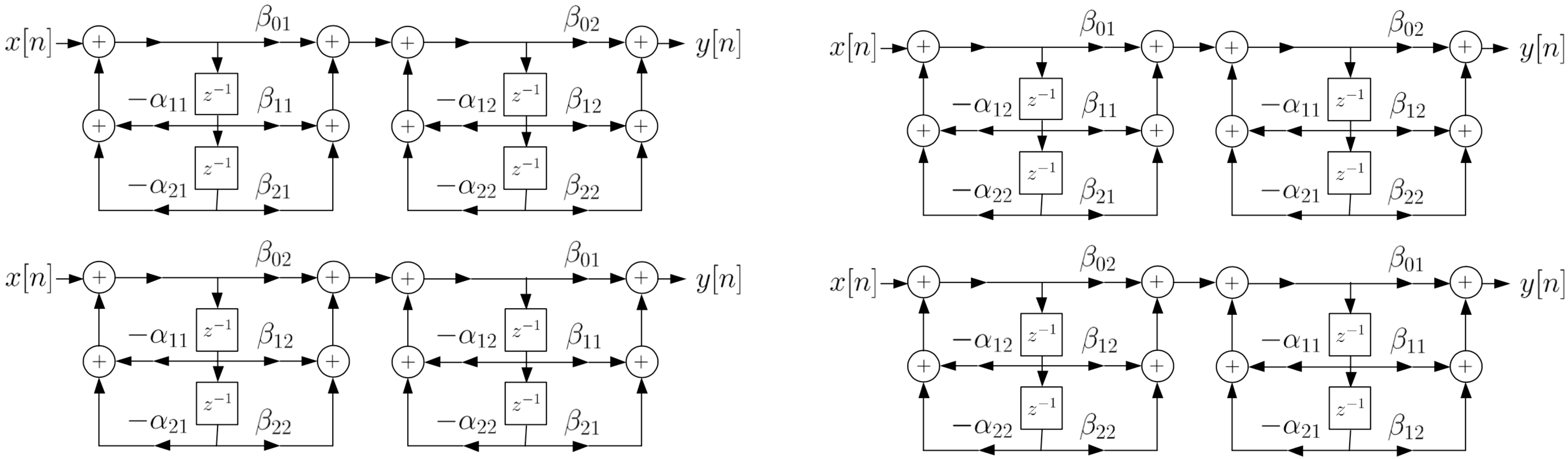
Cascade Form (2)

- Each second-order subsystem

$$\frac{\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}}$$

- can be realized in either the direct or canonic form. Nevertheless, the canonic form is preferred because it requires the minimum number of delay elements.
- In IIR filter implementation, we can group the numerator and denominator in different ways, leading to different pole and zero combinations in each of the second-order sections.

Four Possible Cascade Realizations for 4th Order IIR Filter



Complexity of Cascade Form

- To save the computational complexity, we express the transfer function in cascade form as

$$H(z) = G \prod_{k=1}^{N_c} \frac{1 + \beta'_{1k} z^{-1} + \beta'_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}}$$

- Where $G = \beta_{01}\beta_{02} \ldots \beta_{0N_c}$, $\beta'_{01} = \beta_{1k}/\beta_{0k}$ and $\beta'_{21} = \beta_{2k}/\beta_{0k}$, $k = 1, 2, \ldots, N$
- Assuming that N is even with $N = M$, the cascade implementation needs N or $2N$ delay elements, $(2N + 1)$ multiplications and $2N$ additions, for computing each y[n]. That is , its memory and computational requirements are equal to those of the direct or canonic form.

Example

- Draw the signal flow graph using the cascade form with first-order sections for the LTI system whose transfer function is

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

- For each first-order section, canonic form is assumed.

Solution

- Solving the quadratic equations of the numerator and denominator polynomials, we can factorize H(z) as

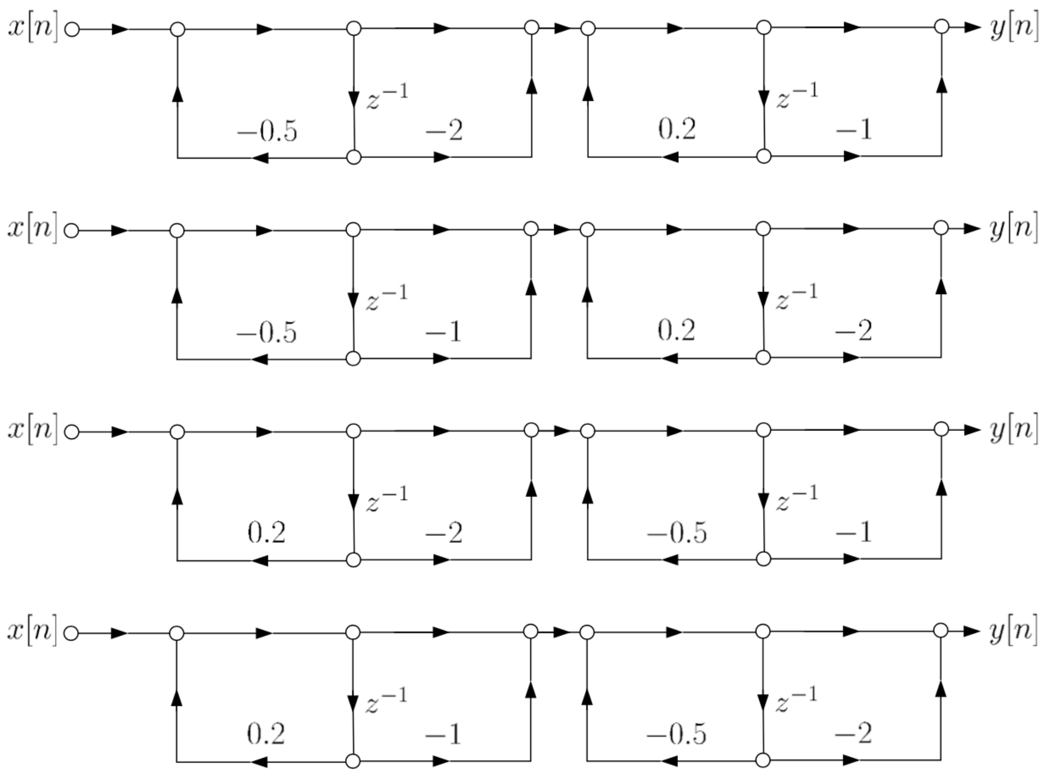
$$H(z) = \frac{\left(1 - 2z^{-1}\right)\left(1 - z^{-1}\right)}{\left(1 + 0.5z^{-1}\right)\left(1 - 0.2z^{-1}\right)}$$

- There are four possible cascade forms for H(z)

$$\begin{aligned} H(z) &= \frac{1 - 2z^{-1}}{1 + 0.5z^{-1}} \cdot \frac{1 - z^{-1}}{1 - 0.2z^{-1}} \\ &= \frac{1 - z^{-1}}{1 + 0.5z^{-1}} \cdot \frac{1 - 2z^{-1}}{1 - 0.2z^{-1}} \\ &= \frac{1 - 2z^{-1}}{1 - 0.2z^{-1}} \cdot \frac{1 - z^{-1}}{1 + 0.5z^{-1}} \\ &= \frac{1 - z^{-1}}{1 - 0.2z^{-1}} \cdot \frac{1 - 2z^{-1}}{1 + 0.5z^{-1}} \end{aligned}$$

- Note that although all four realizations are equivalent for infinite precision, they may differ in actual implementation when finite-precision numbers are employed.

4 Possible Cascade Realizations



Parallel Form

- The ideal of the parallel form is similar to the partial fraction expansion of z-transform

$$H(z) = \sum_{l=0}^{M-N} B_l z^{-l} + \sum_{k=1}^{N_c} \frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}}$$

- Where $N_c = \lfloor (N+1)/2 \rfloor$. But now we use second-order sections in order to ensure all $\{\gamma_{0k}\}$, $\{\gamma_{1k}\}$, $\{\alpha_{1k}\}$ and $\{\alpha_{2k}\}$ are real.
- Note that when $M < N$, the first summation term will not be included.

Example : Parallel Form

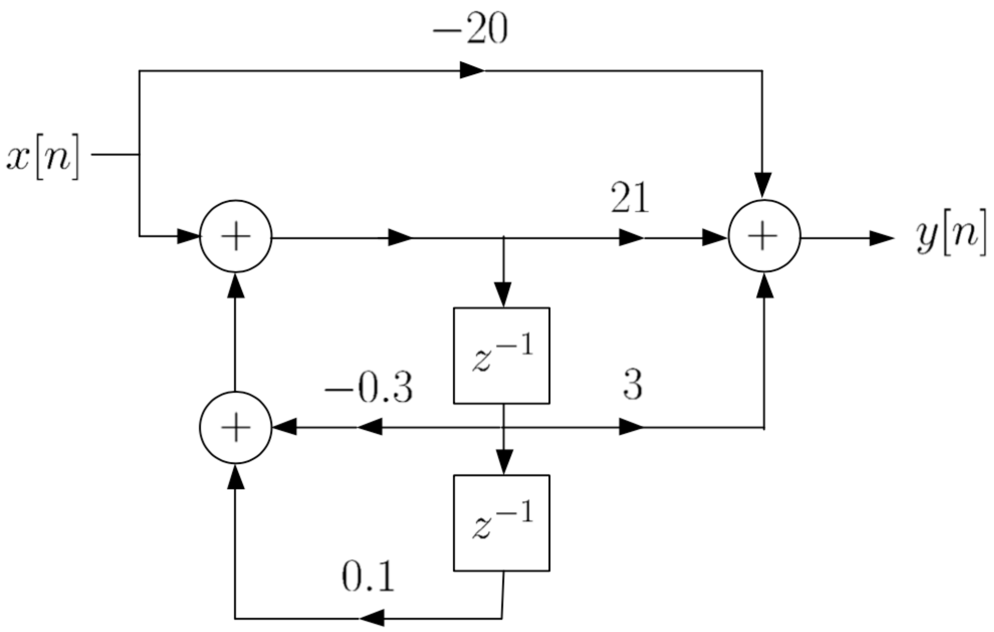
- Draw the block diagram using parallel form for a LTI system whose transfer function is

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

Solution (1)

- Following the long division, we can obtain

$$H(z) = -20 + \frac{21 + 3z^{-1}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

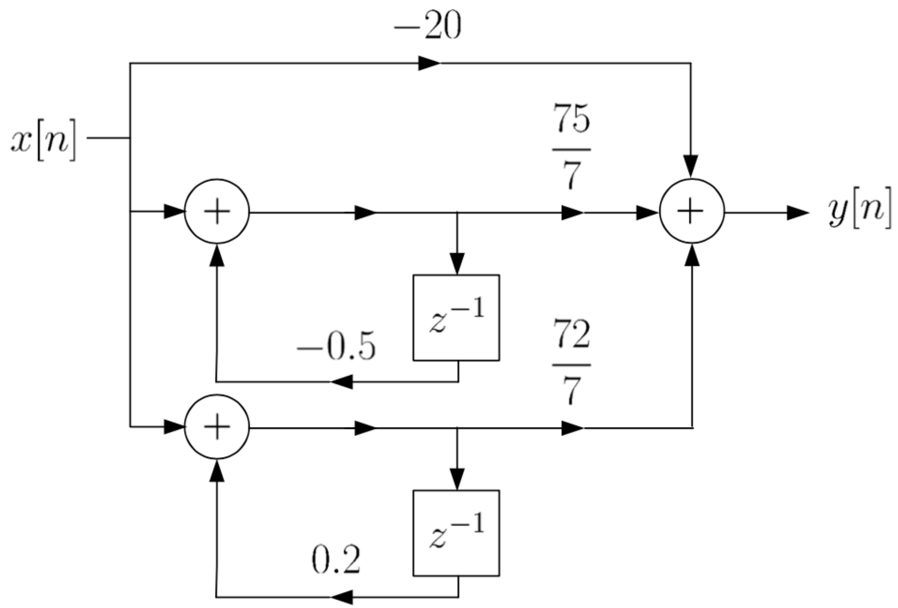


Parallel Form with Second-Order Section

Solution (2)

- As the poles of H(z) are real, we can also express H(z) in terms of first-order sections as

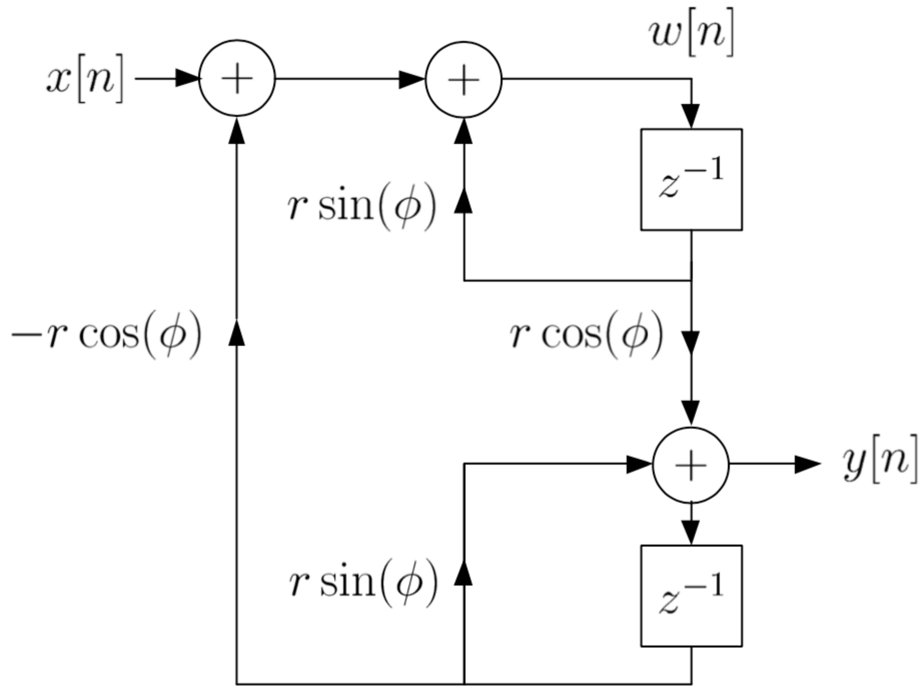
$$H(z) = -20 + \frac{\frac{75}{7}}{1+0.5z^{-1}} + \frac{\frac{72}{7}}{1-0.2z^{-1}}$$



Parallel Form with First-Order Sections

Example

- Determine the transfer function H(z) and the difference equation which relates x[n] and y[n] for



Solution (1)

We first introduce an intermediate sequence $w[n]$ to relate $x[n]$ and $y[n]$. Then we can establish:

$$w[n] = w[n-1] \cdot r\sin(\phi) - y[n-1] \cdot r\cos(\phi) + x[n]$$

and

$$w[n-1] \cdot r\cos(\phi) + y[n-1] \cdot r\sin(\phi) = y[n]$$

Applying z transform yields:

$$W(z) = W(z)z^{-1}r\sin(\phi) - Y(z)z^{-1}r\cos(\phi) + X(z)$$

and

$$W(z)z^{-1}r\cos(\phi) + Y(z)z^{-1}r\sin(\phi) = Y(z)$$

From the second equation, we have:

$$W(z) = \frac{1 - z^{-1}r\sin(\phi)}{z^{-1}r\cos(\phi)}Y(z)$$

Solution (2)

Substituting it into the first equation, we finally have:

$$\frac{1 - z^{-1} r \sin(\phi)}{z^{-1} r \cos(\phi)} Y(z) \left(1 - z^{-1} r \sin(\phi)\right) = -Y(z) z^{-1} r \cos(\phi) + X(z)$$

$$\Rightarrow Y(z) \left(1 - 2z^{-1} r \sin(\phi) + z^{-2} r^2\right) = X(z) z^{-1} r \cos(\phi)$$

$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1} r \cos(\phi)}{1 - 2z^{-1} r \sin(\phi) + z^{-2} r^2}$$

Taking the inverse z transform gives:

$$y[n] - 2r \sin(\phi) y[n-1] + r^2 y[n-2] = r \cos(\phi) x[n-1]$$

Comparison of Difference Structures

- The major factors that affect our choice of a specific realization are computational complexity, memory requirement, and finite word-length effects. Assuming that M is even with M=N:

FIR Filter using Non-Recursive Structure Comparison

Structure	Multiplication	Addition	Register
Direct form	$M+1$	M	M
Cascade form	$M+1$	M	M

IIR Filter using Recursive Structure Comparison

Structure	Multiplication	Addition	Register
Direct form	$2M+1$	$2M$	$2M$
Canonic form	$2M+1$	$2M$	M
Cascade form	$2M+1$	$2M$	M
Parallel form	$2M+1$	$3M/2+1$	M

- Computations of the direct form can be reduced if the FIR filter coefficients are symmetric or anti-symmetric.
- When the filter coefficients are expressed using infinite precision numbers, all realizations are same. However, in practice, they are processed in registers which have finite word-lengths. In the presence of quantization errors, the cascade and parallel realizations are more robust than the direct and canonic forms, that is, they have frequency responses closer to the desired responses.
- FIR filters are less sensitive than IIR filters to finite word-length effects.
- For a feasible system, it should be causal and stable.
- In cascade and parallel realizations of IIR filters, system stability can be easily monitored by checking pole locations.