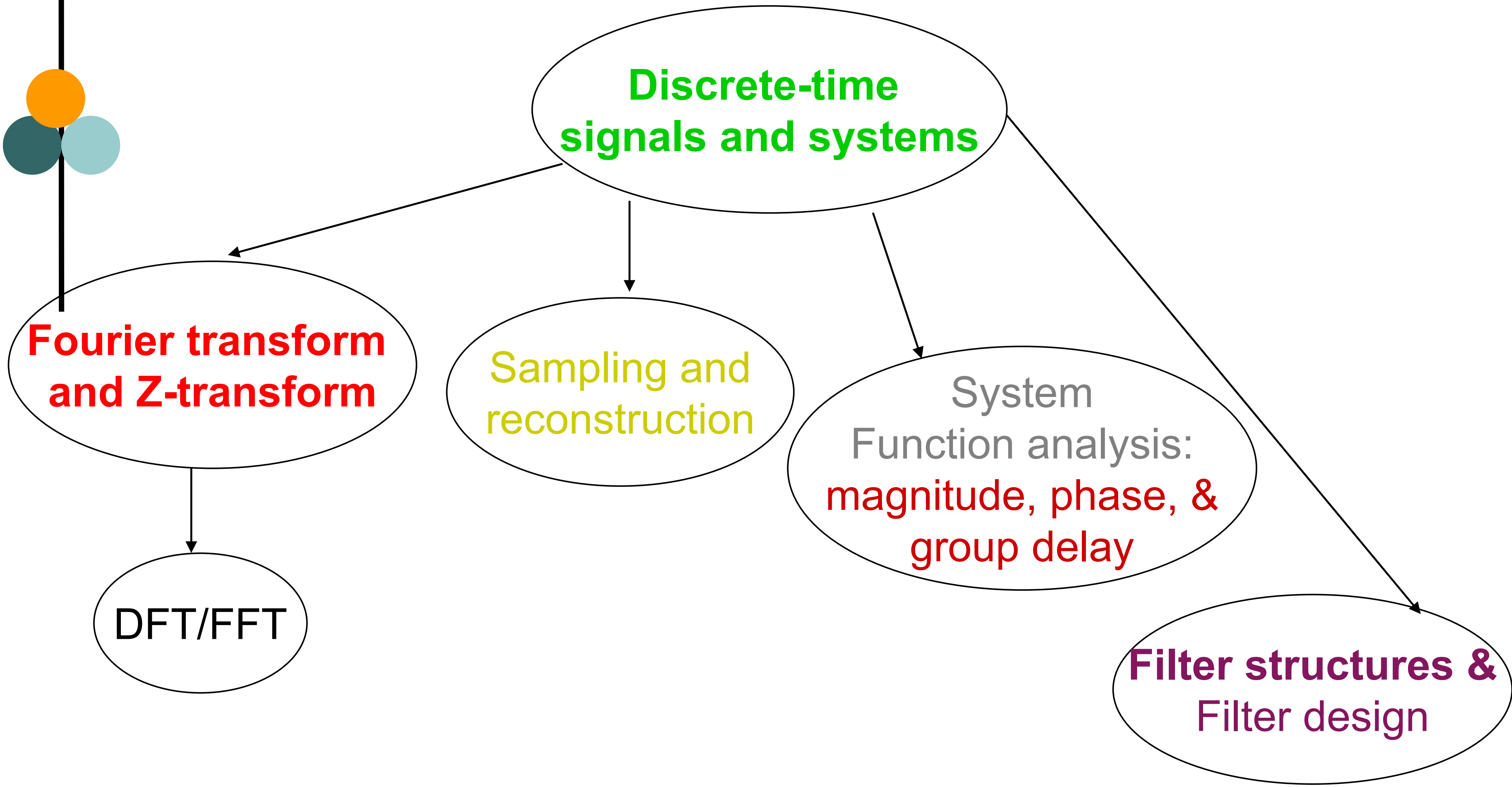
DSP Course at a glance



Types of Filters

A filter remove unwanted signal components and/or enhance wanted ones

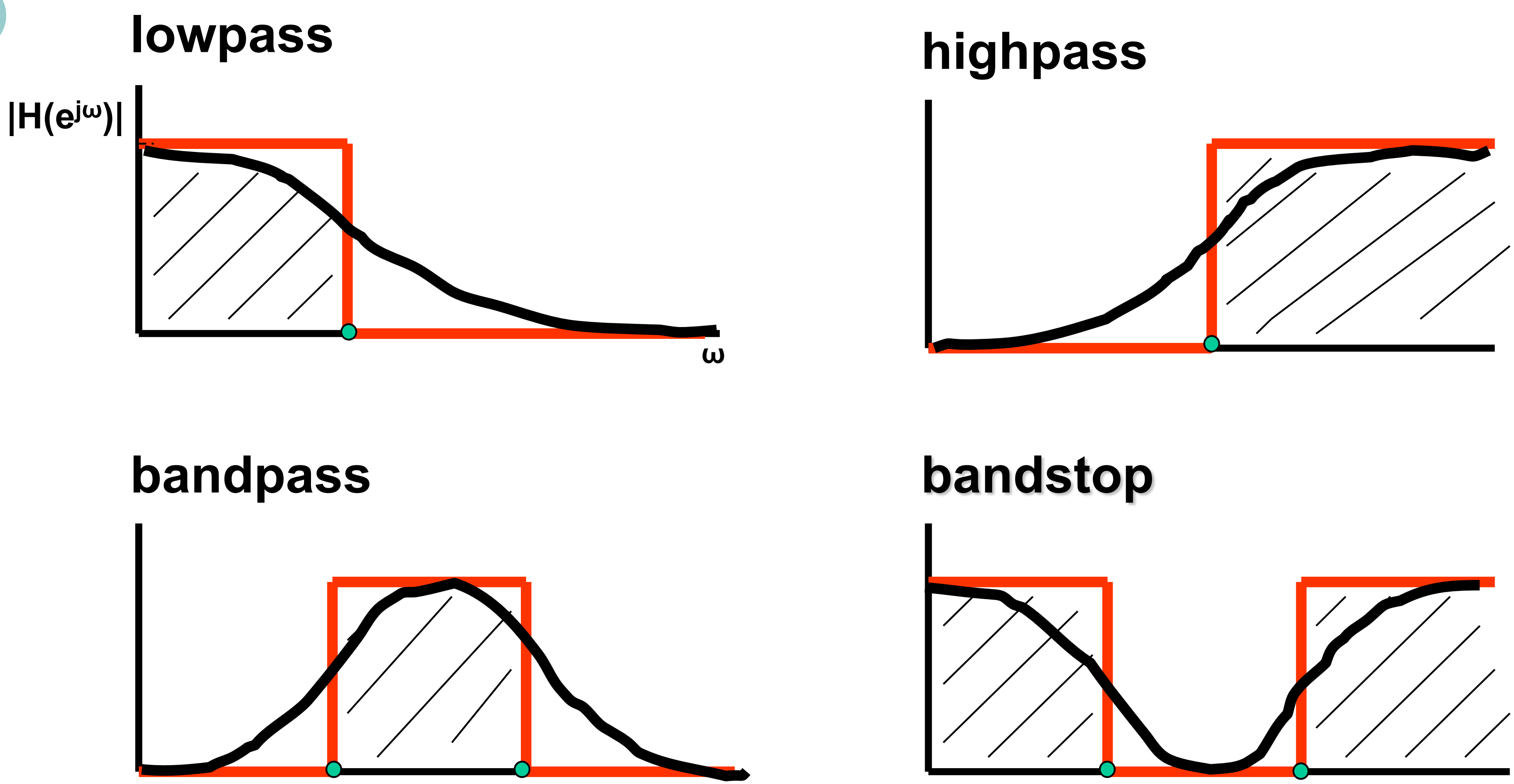
Four Main Filter Types:

- Low-pass: most common
  - Passes low frequencies, attenuates highs
- High-pass:
  - Passes high frequencies, attenuates lows
  - Used to brighten a signal
  - Careful: can also increase noise
- Band-pass:
  - Passes band of frequencies, attenuates those above and below band
  - Most common in implementations of DFF to separate out harmonics
- Band-stop (band-reject):
  - Stops band of frequencies, passes those above and below band

Any filter type can be realized either as a FIR or IIR filter

Types of Filters

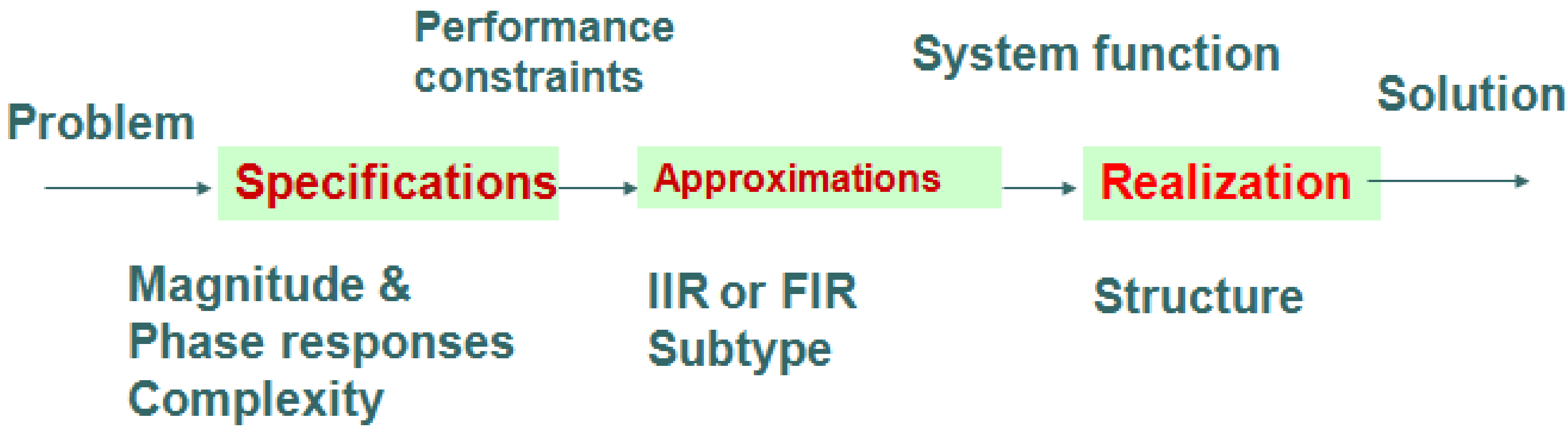
Ideal filters — **Real filters** —



Ideal frequency-selective filter has **unity** frequency response over a certain range of frequencies, and is **zero** at the remaining frequencies

Filter design process

- Filter, in broader sense, covers any system
- Three design steps



Structures for DT Systems & Finite Precision Numerical Effects

1. Structures For Discrete Time Systems:6.0->6.5
   - Description of LTI & Block Diagram Representation
   - Direct Form I
   - Direct Form II
   - Signal Flow Graph Representation
   - Basic Structures For IIR Systems
   - Transposed Forms
   - Basic Structures For FIR Systems
2. Finite Precision Numerical Effects: (not in the exam)
   - Quantization in Implementing Systems-Effects in IIR & FIR Systems
   - Round Off Noise and Analysis of Quantization Error
   - Limit Cycles

M. Amer
Concordia University
Electrical and Computer Engineering

Content and figures are based on/from

- Oppenheim & Schafer, DSP

Description of LTI systems

1. Difference Equations
2. Rational System Function
3. Realization Structures
   1. Block diagram
   2. Signal flow graph

After a filter is designed, it needs to be *realized or* implemented

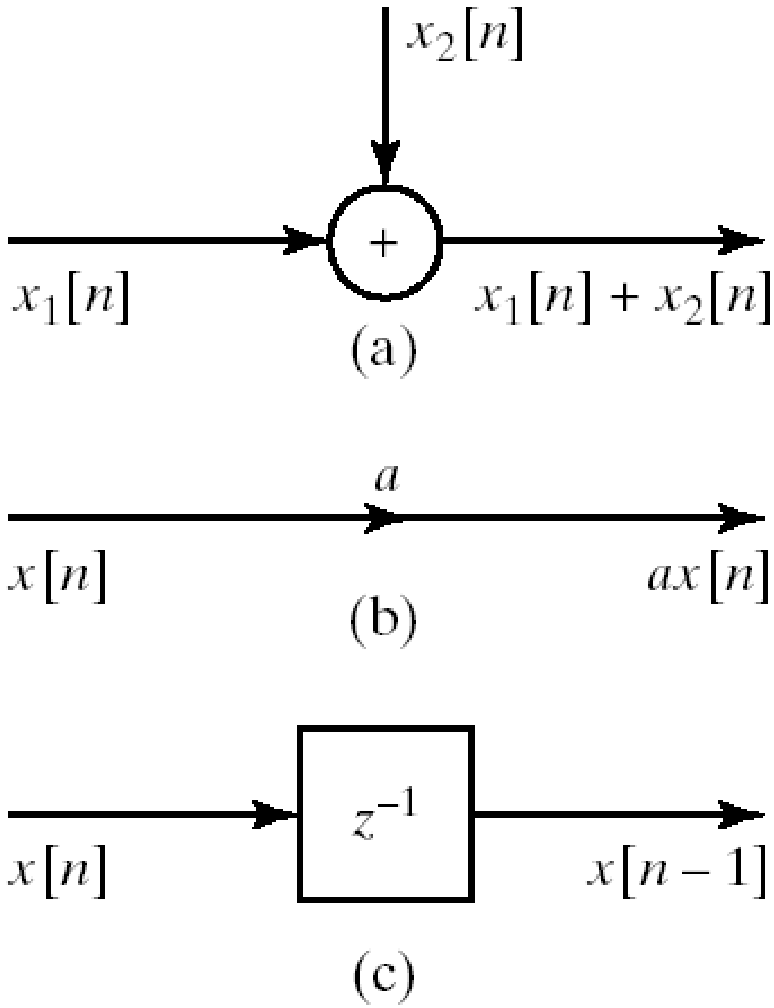
– Developing a structure (a signal flow diagram or a block diagram) that describes the filter in terms of operations on signals

Realization Structures

- Different realizations/structures will have different numerical properties (e.g., precision)
  - A system function may be realized in many ways
    - Example: $ax + bx + c$ or $x(a + b) + c$
- A realization maybe more efficient in terms of
  - the number of operations or
  - storage elements required for their implementation
- A realization may show better numerical stability and reduced round-off error
- A realization maybe more optimal for
  - fixed-point arithmetic or
  - floating-point arithmetic

Block Diagram Representation

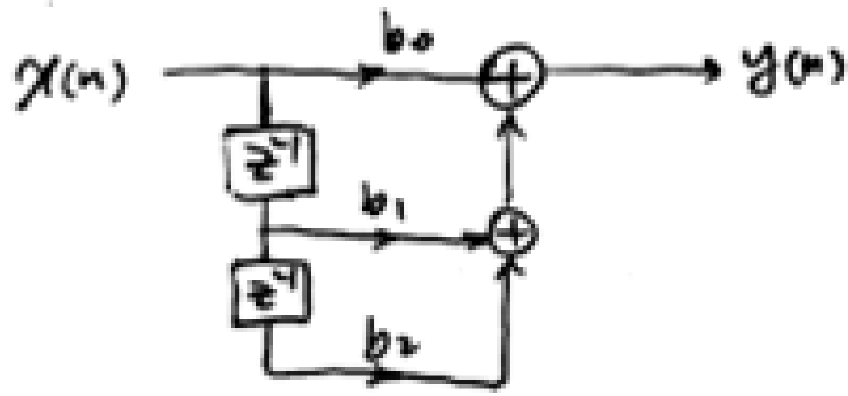
- LTI systems with rational system function can be represented as constant-coefficient <u>difference equation</u>
- The implementation of difference equations <u>requires delayed values</u> of the
  - input
  - output
  - intermediate results
- The requirement of delayed elements implies need for storage
- We also need means of
  - addition
  - multiplication



Examples

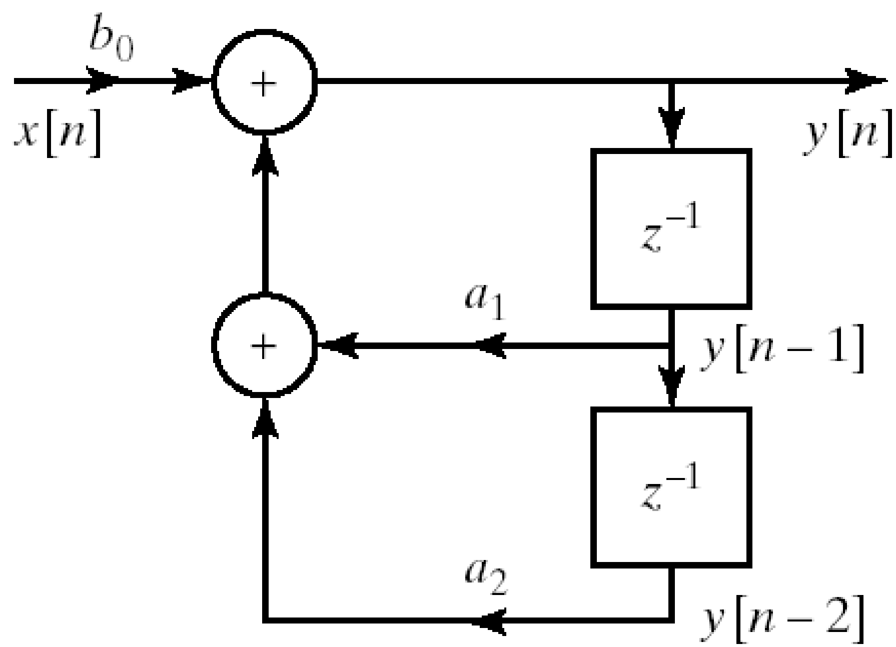
Ex 1: $y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2}$$



Ex2: $y(n) = a_1 y(n-1) + a_2 y(n-2) + b_0 x(n)$

$$H(z) = \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2}}$$



$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n]$$

OUTLINE

Structures For Discrete Time Systems:

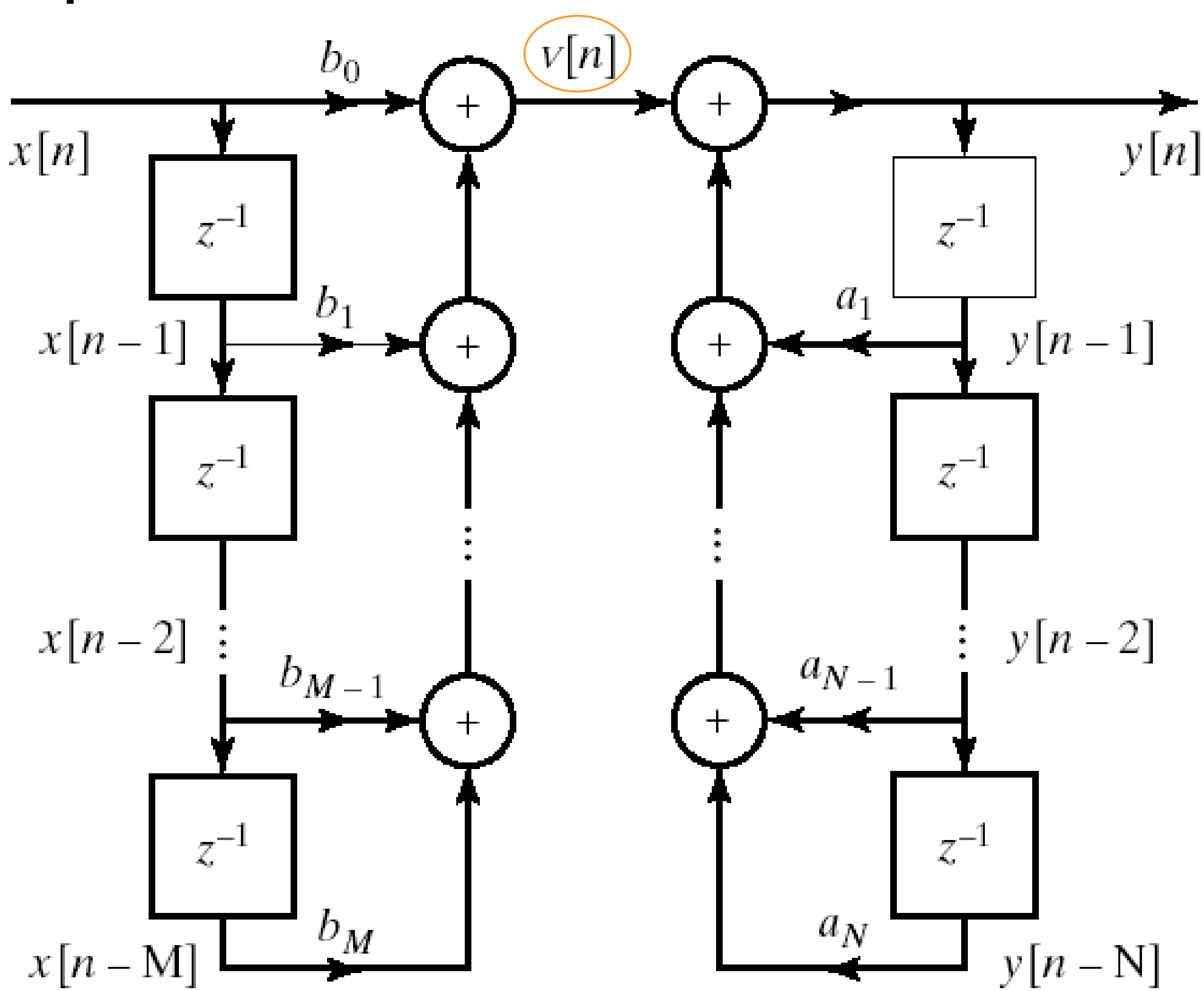
- o Description of LTI & Block Diagram Representation
- o **Direct Form I & II**
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o Transposed Forms
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

Direct Form I

- General form of difference equation $\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$
- Alternative equivalent form $y[n] + \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$



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

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

1) Realize zeros of H(z)
2) Realize the poles of H(z)

- M+N+1 multiplications
- M+N additions
- M+N delays

Direct Form I

Should be "1"

- Transfer function

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

- Direct Form I represents

$$H(z)=H_2(z)H_1(z)=\left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right)\left(\sum_{k=0}^{M} b_k z^{-k}\right)$$

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

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

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

$$y[n]=\sum_{k=1}^{N} a_k y[n-k]+v[n]$$

Alternative Representation

- Replace order of cascade LTI systems
  - Note: $H_1H_2 = H_2H_1$ theoretically but practically have different properties

$$H(z) = H_1(z)H_2(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right)\left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right)$$

$$W(z) = H_2(z)X(z) = \left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right)X(z)$$

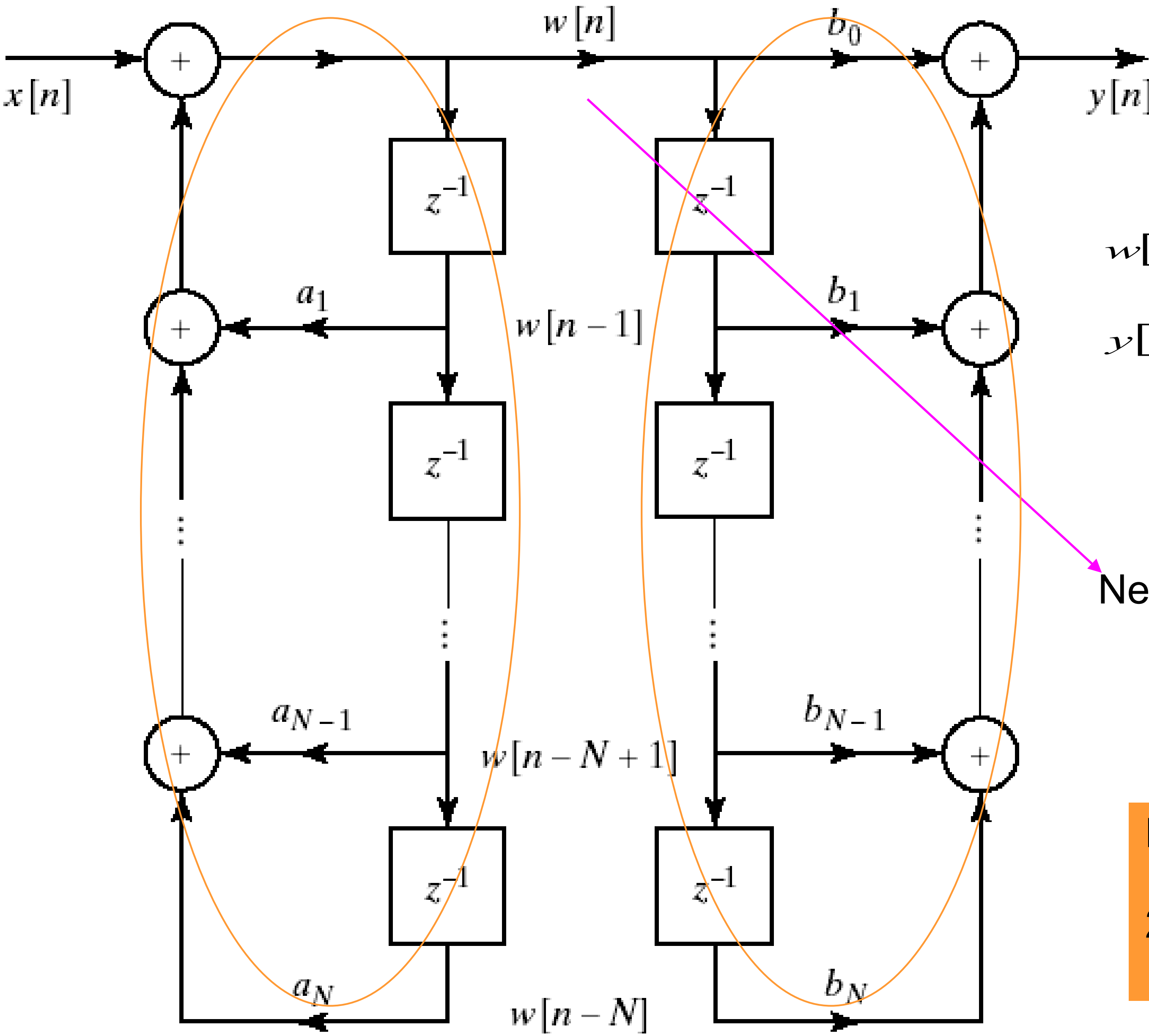
$$Y(z) = H_1(z)W(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right)W(z)$$

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$

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

Alternative Block Diagram

- We can change the order of the cascade systems
- Assume N=M; if not, some of the coef. Will be zero



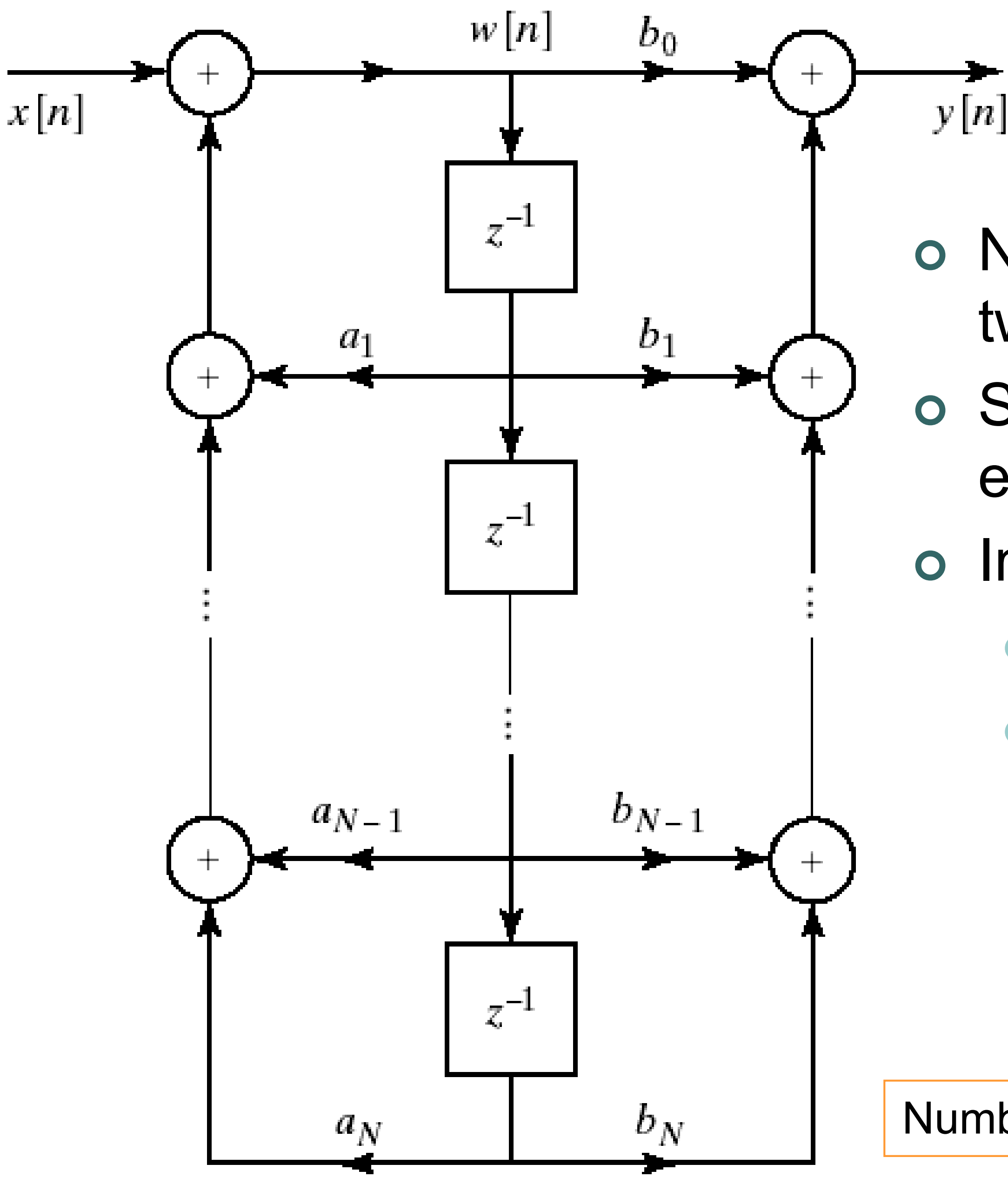
$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$
$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

Need to store the same data twice

1) Realize poles of H(z)
2) Realize zeros of H(z)

Difference
1) in numerical precision
2) This form can be redrawn to reduce delays

Direct Form II/Canonical Form



- No need to store the same data twice in previous system
- So we can collapse the delay elements into one chain
- Implementation wise
  - Less memory
  - Difference when using finite-precision arithmetic

Number of delays: ***max(N,M)***

OUTLINE

Structures For Discrete Time Systems:

- o Description of LTI & Block Diagram Representation
- o Direct Form I & II:
  - o In the exam: given system function, draw a structure
- o **Signal Flow Graph Representation**
- o Basic Structures For IIR Systems
- o Transposed Forms
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

- Quantization in Implementing Systems-Effects in IIR & FIR Sys.
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

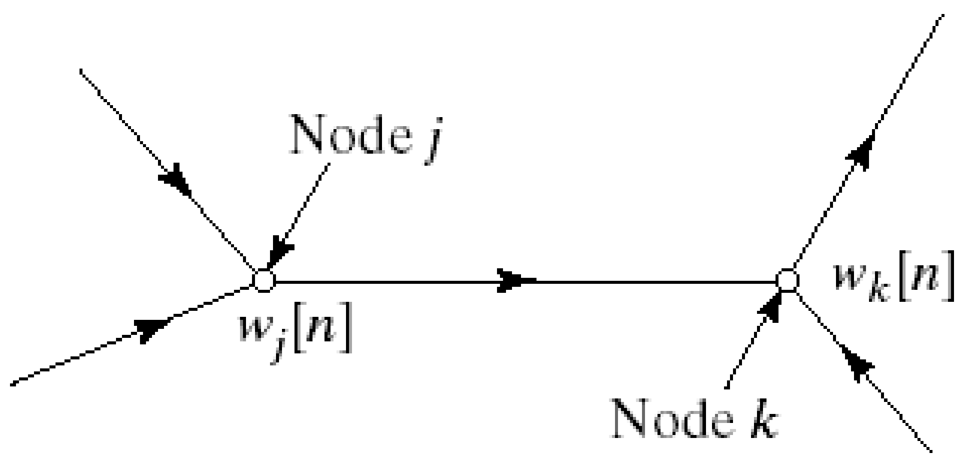
Signal Flow Graph (SFG) Representation

- Similar to block diagram representation: Notational differences

- Describe a system from network perspective
- Not only for LTI systems
- Very useful in analysis and representation of complicated systems

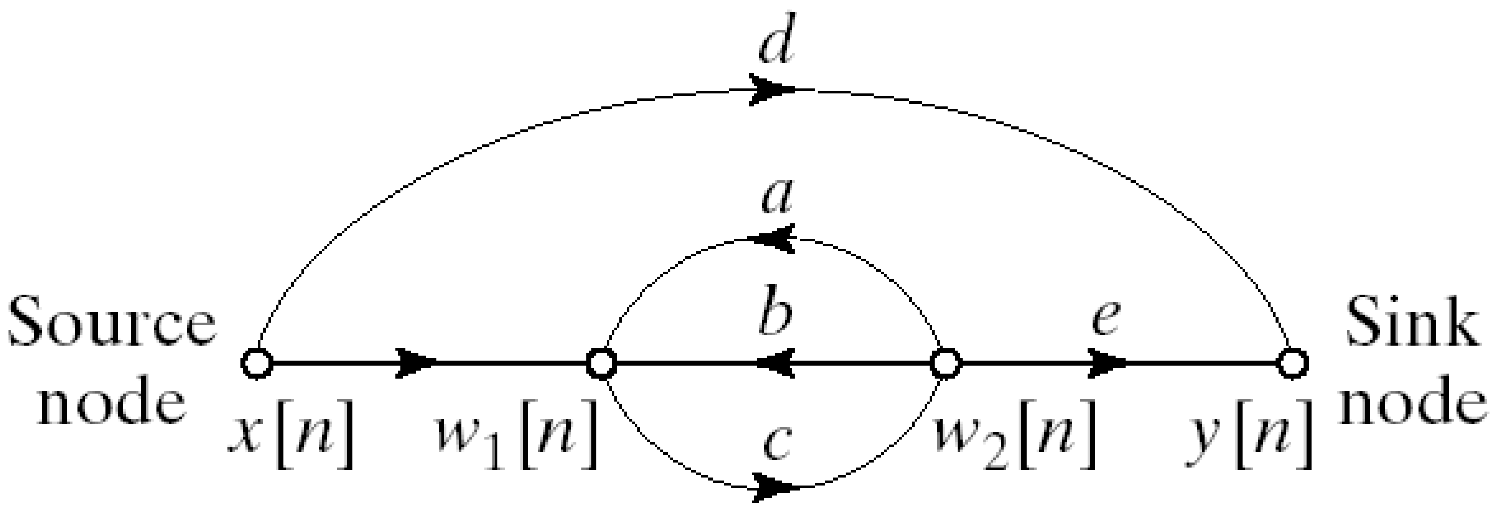
- Two elements in flow graph: nodes and branches

- A network of directed branches connected at nodes



Signal Flow Graph Representation

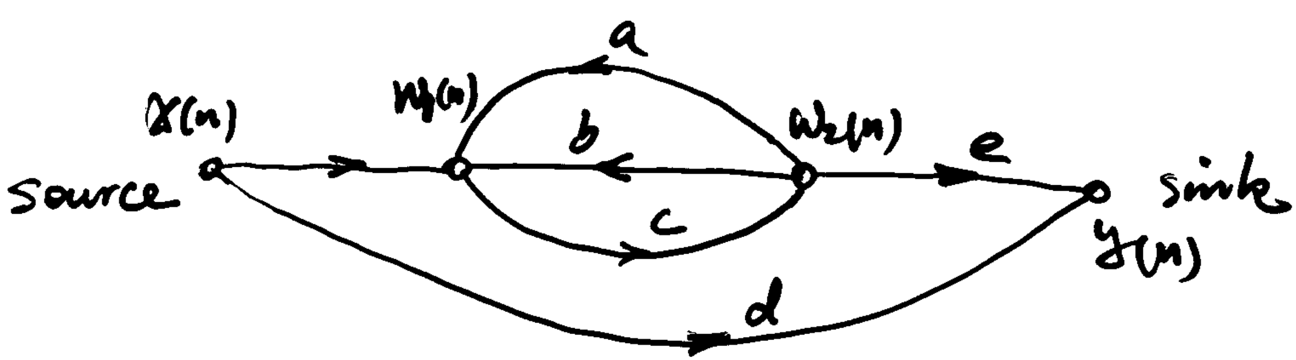
- 2 special nodes: source node x[n] and sink node y[n]
- A branch point: receives one entry/input
- Difference to block diagrams:
  node represents both branching points and adder
- Adder: receives two or more
- Example representation of a difference equation



- Advantages of SFG:
  - Simpler to draw
  - Many SFG theories exists

Signal Flow Graph Representation

How to establish a relationship between $x(n)$ and $y(n)$?



$$w_1(n) = x(n) + a\,w_2(n) + b\,w_2(n) \quad (A)$$

$$w_2(n) = c\,w_1(n) \quad (B)$$

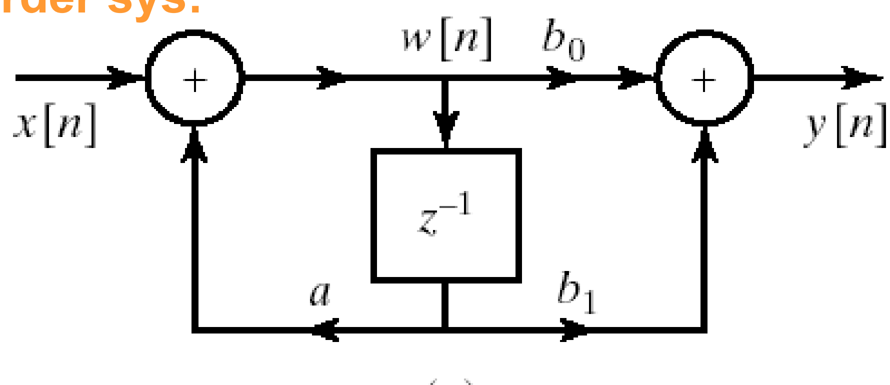
$$y(n) = d\,x(n) + e\,w_2(n) \quad (C)$$

From (A), (B), $w_1(n) = \dfrac{x(n)}{1-(a+b)c}$ (D)

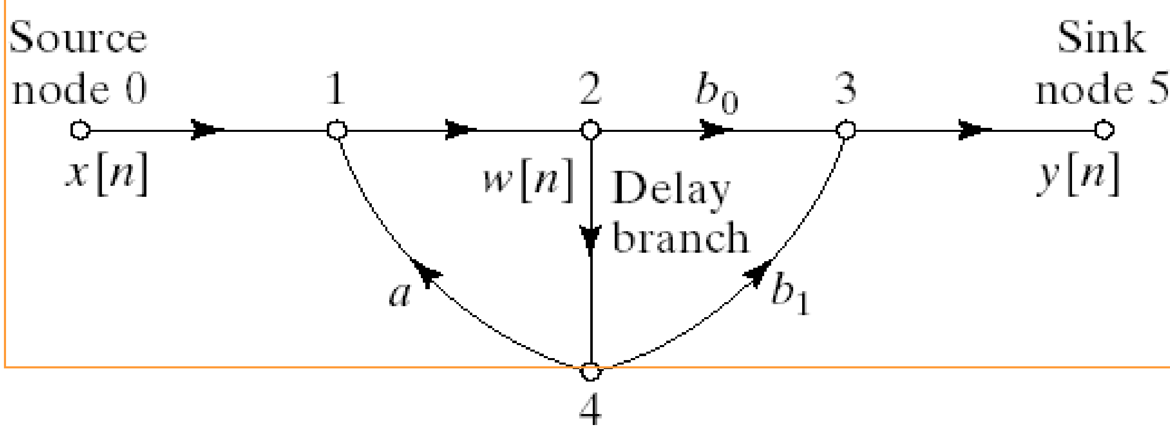
From (C), (D), $y(n) = \left[d + \dfrac{ec}{1-(a+b)c}\right] x(n)$

Example: Representation of Direct Form II with SFG

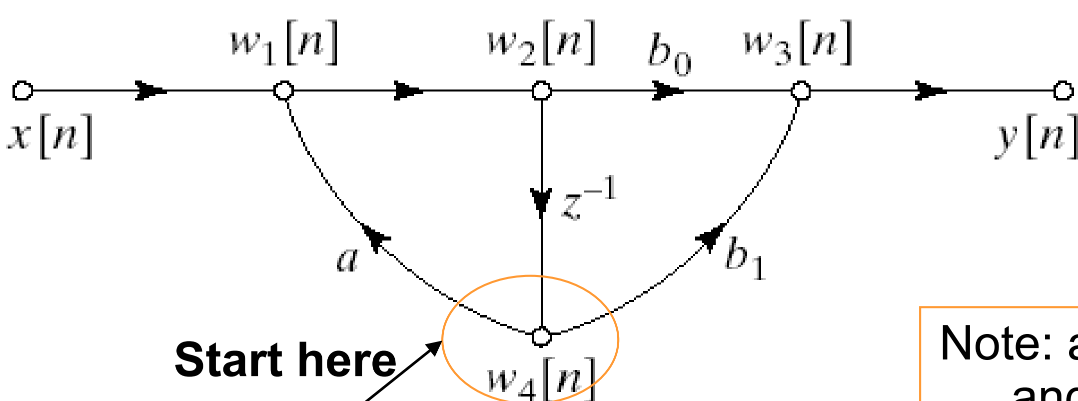
1st-order sys:



(a)



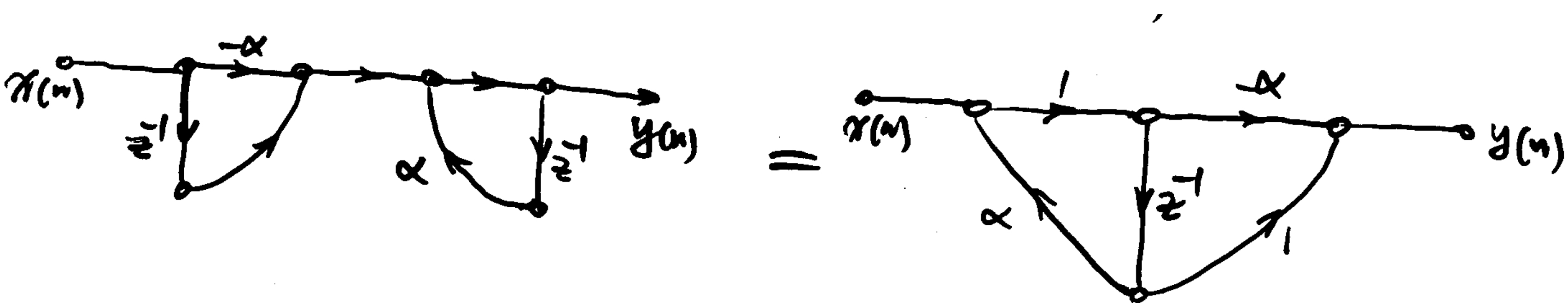
$$w_1[n] = a w_4[n] + x[n]$$
$$w_2[n] = w_1[n]$$
$$w_3[n] = b_0 w_2[n] + b_1 w_4[n]$$
$$w_4[n] = w_2[n-1]$$
$$y[n] = w_3[n]$$



$$w_1[n] = a w_1[n-1] + x[n]$$
$$y[n] = b_0 w_1[n] + b_1 w_1[n-1]$$

Note: a node represents both branching points and adder

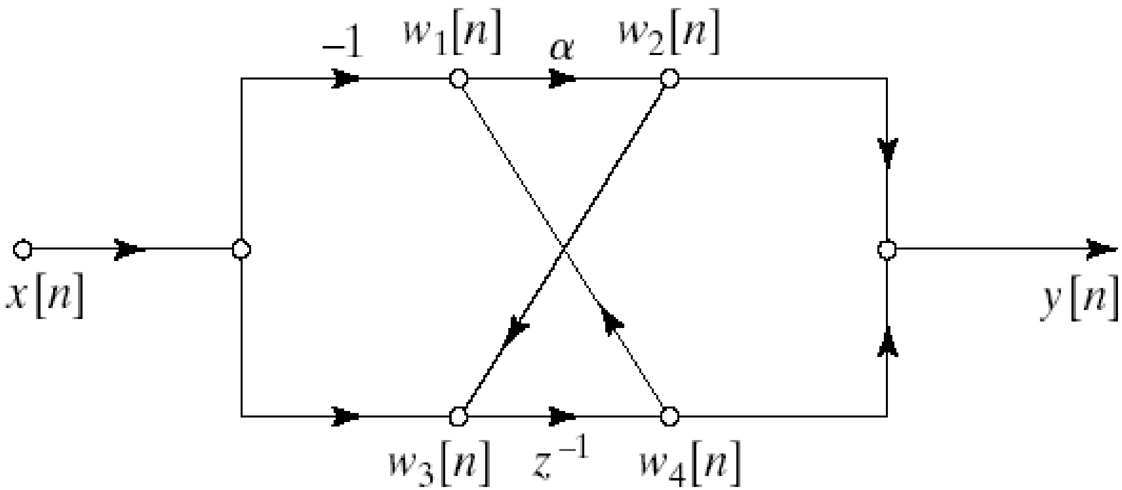
SFG: Direct Form I versus Direct Form II



Direct form I:
2 delays &
2 multiplications

Direct form II:
1 delay &
2 multiplications

Example: find System Function from SFG



1 delay &
1 multiplication

$$
\begin{aligned}
w_1[n] &= w_4[n] - x[n] \\
w_2[n] &= \alpha w_1[n] \\
w_3[n] &= w_2[n] + x[n] \\
w_4[n] &= w_3[n-1] \\
y[n] &= w_2[n] + w_4[n]
\end{aligned}
\quad \rightarrow \quad
\begin{aligned}
W_1(z) &= W_4(z) - X(z) \\
W_2(z) &= \alpha W_1(z) \\
W_3(z) &= W_2(z) + X(z) \\
W_4(z) &= W_3(z) z^{-1} \\
Y(z) &= W_2(z) + W_4(z)
\end{aligned}
\quad \rightarrow \quad
\begin{aligned}
W_2(z) &= \frac{\alpha X(z)\left(z^{-1} - 1\right)}{1 - \alpha z^{-1}} \\
W_4(z) &= \frac{X(z) z^{-1}(1 - \alpha)}{1 - \alpha z^{-1}} \\
Y(z) &= W_2(z) + W_4(z)
\end{aligned}
$$

Start simplifying here

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$$

$$h[n] = \alpha^{n-1} u[n-1] - \alpha^{n+1} u[n]$$

➔ All pass system

OUTLINE

Structures For Discrete Time Systems:

- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o **Basic Structures For IIR Systems**
  - o **In the exam: Able to draw and recognize a form**
- o Transposed Forms
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

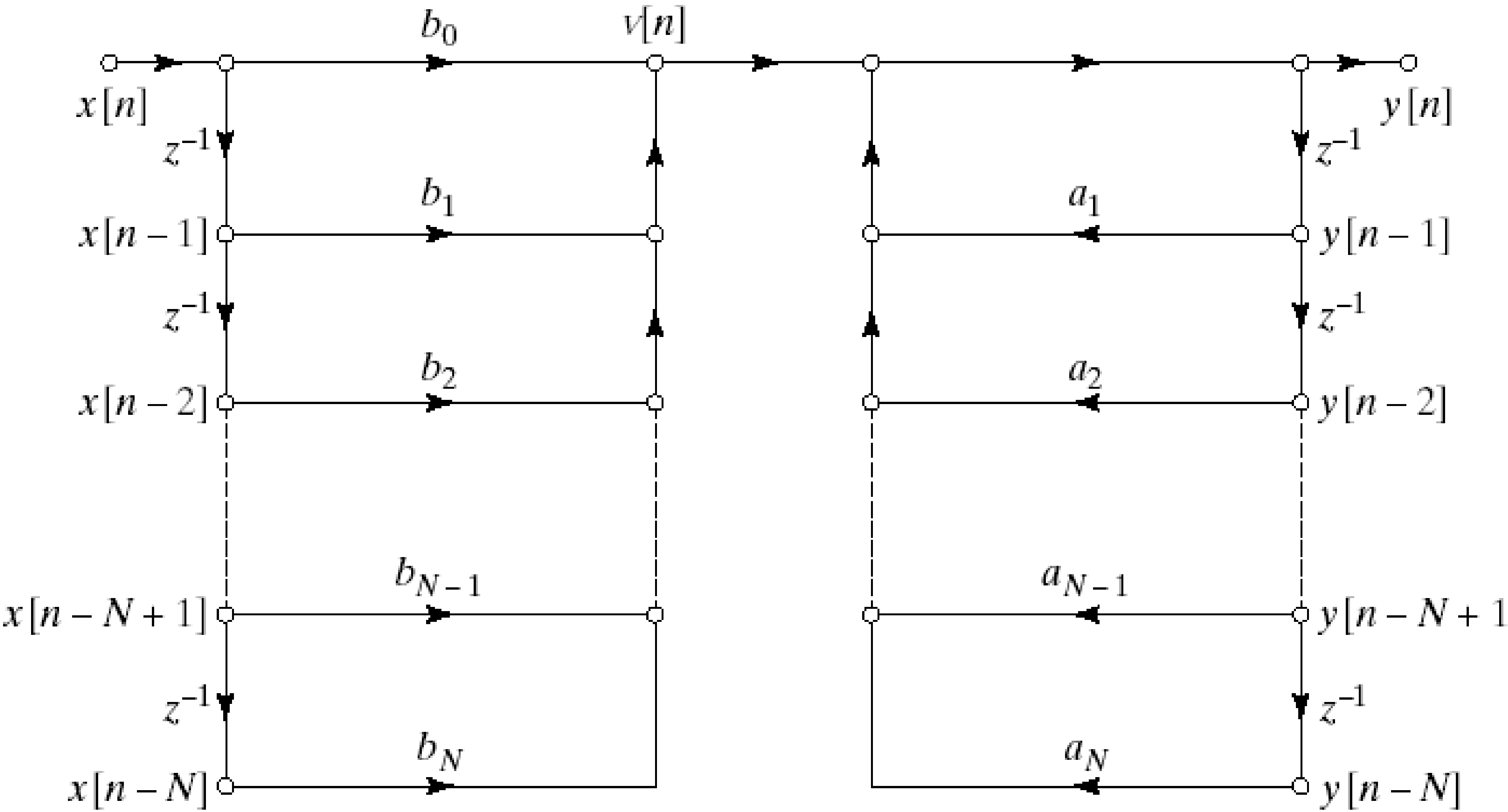
Basic Structures for IIR Systems

- Direct forms
- Cascade forms
- Parallel form

➔They differ in

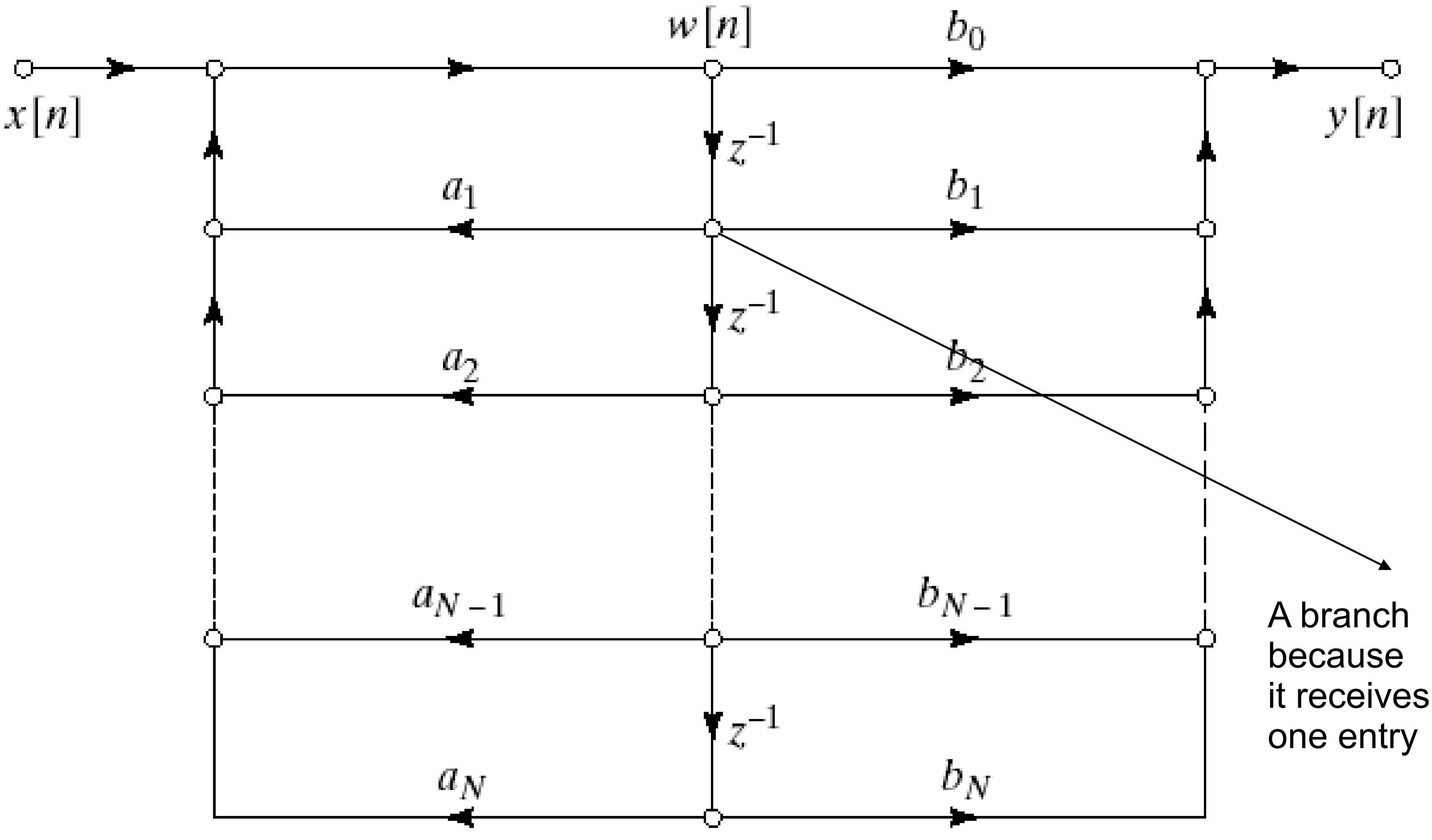
1. computational and storage complexity,
2. effect in finite precision,
3. modularity and simplicity of data transfer (important for hardware implementation)

Basic Structures for IIR Systems: Direct Form I



Signal flow graph of <u>direct form I</u> structure for an Nth order system

Basic Structures for IIR Systems: Direct Form II



Signal flow graph of <u>direct form II</u> structure for an Nth order system

Basic Structures for IIR Systems: Cascade Form

- *H(z)=N(z)/D(z)*
- If we factor *N(z)* and *D(z)* we get the general form for cascade implementation

$$H(z) = A\frac{\prod_{k=1}^{M_1}\left(1 - f_k z^{-1}\right)\prod_{k=1}^{M_2}\left(1 - g_k z^{-1}\right)\left(1 - g_k^* z^{-1}\right)}{\prod_{k=1}^{N_1}\left(1 - c_k z^{-1}\right)\prod_{k=1}^{N_2}\left(1 - d_k z^{-1}\right)\left(1 - d_k^* z^{-1}\right)}$$

$$M = M_1 + 2M_2; \;\; N = N_1 + 2N_2$$

fk and ck are real zeros and poles of $H(z)$

gk and dk are complex zeros and poles of $H(z)$

g_k^* : complex conjugate pair of g_k

Most general distribution of poles & zeros

Basic Structures for IIR Systems: Cascade Form

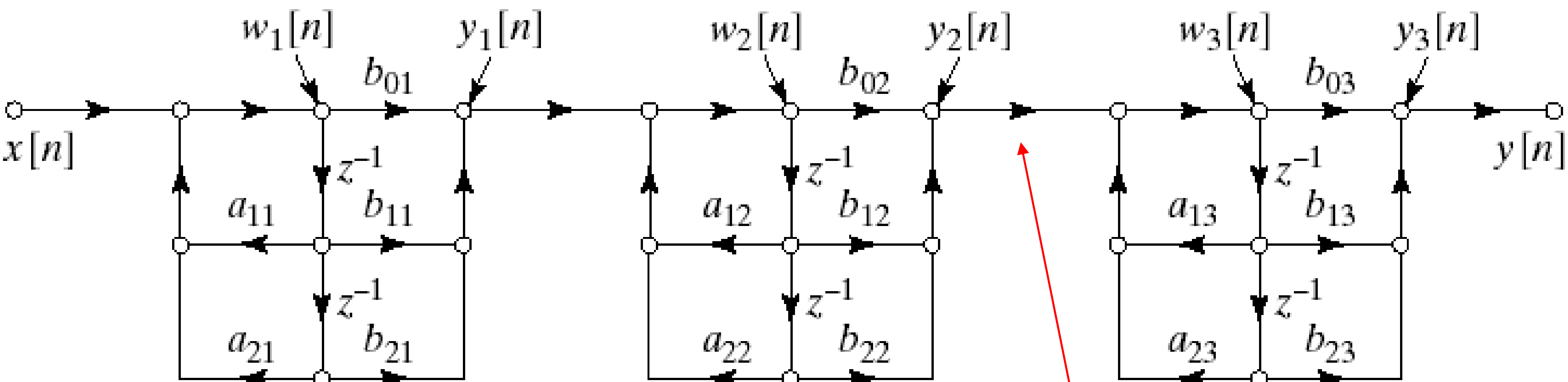
- More practical form in 2nd order systems: group poles and zeros
- We get

$$H(z)=\prod_{k=1}^{Ns}\frac{b_{0k}+b_{1k}z^{-1}-b_{2k}z^{-2}}{1-a_{1k}z^{-1}-a_{2k}z^{-2}}=H_1(z)H_2(z)..H_K(z)$$

$$M<=N;\quad Ns=\lfloor N+1/2\rfloor$$

- Example:

Cascade structure for a 6th order system = three 2nd order subsystems of direct form II

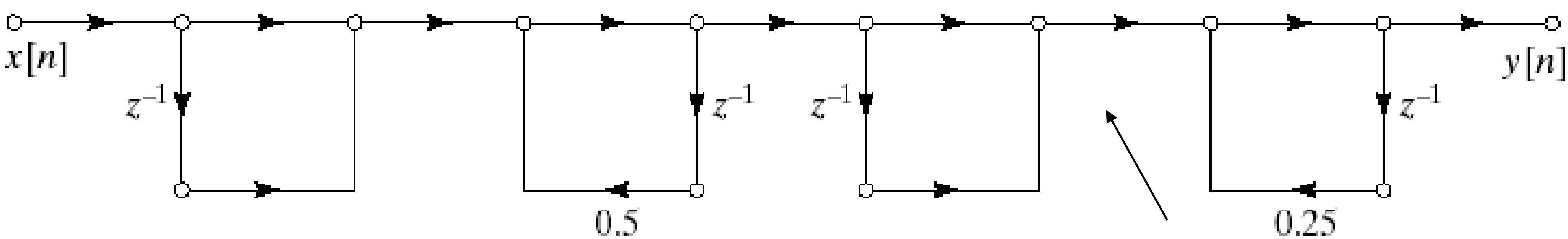


To get higher precision arithmetic: change the order of the sub-systems, i.e., pair different zeros/poles

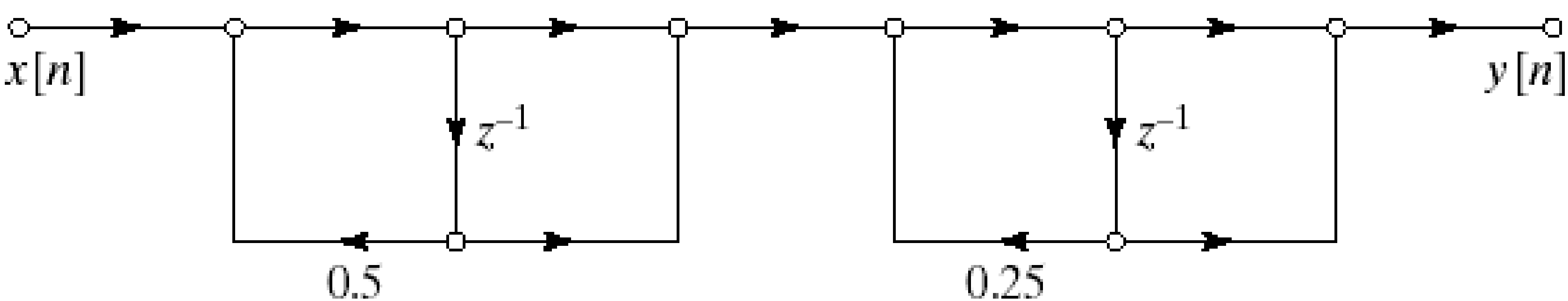
Example

$$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-0.75z^{-1}+0.125z^{-2}}=\frac{\left(1+z^{-1}\right)\left(1+z^{-1}\right)}{\left(1-0.5z^{-1}\right)\left(1-0.25z^{-1}\right)}$$

- Cascade of Direct Form I



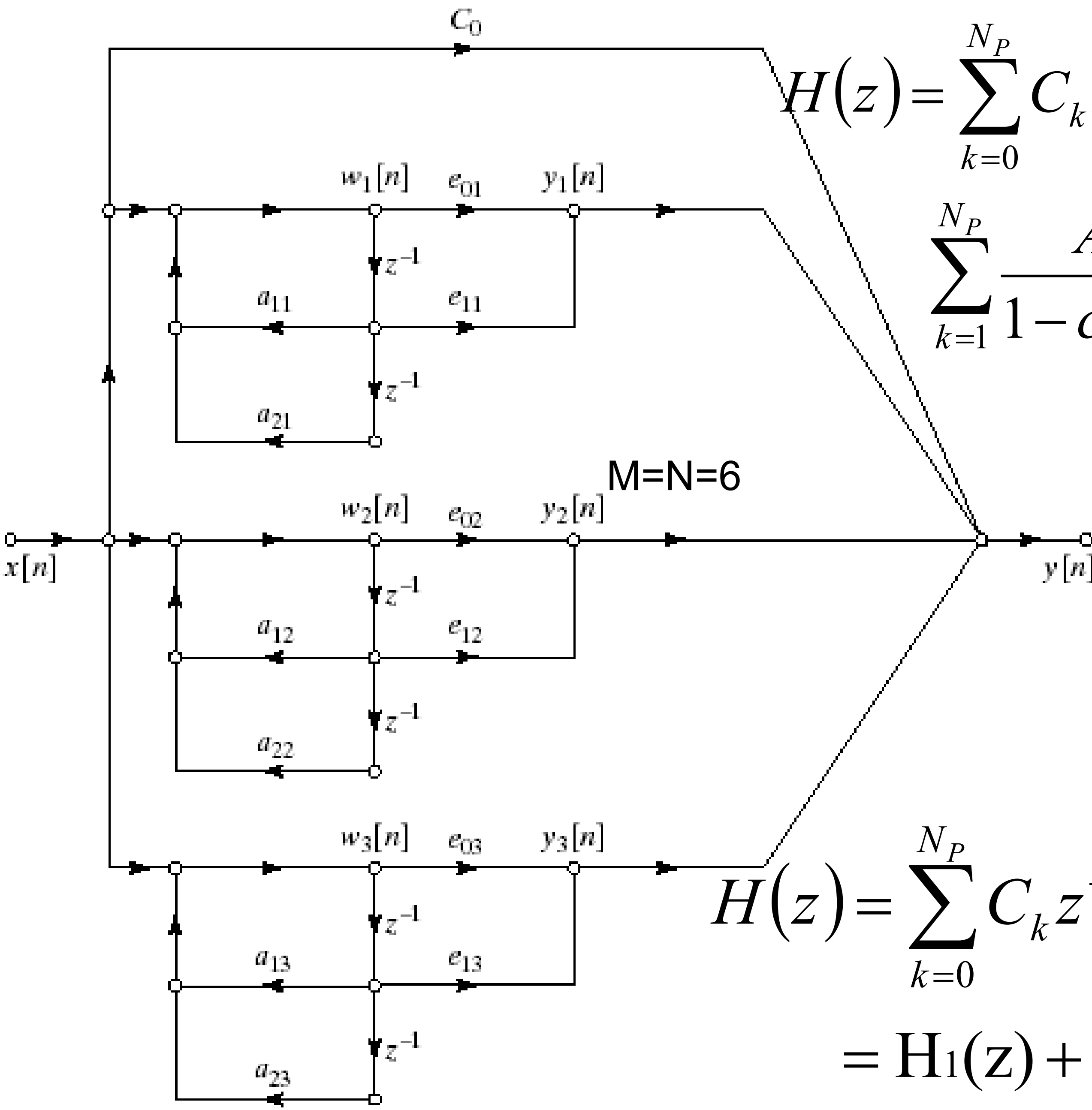
To get higher precision arithmetic: change the order of the sub-systems



Basic Structures for IIR Systems: Parallel Form

Parallel processing: reduces time to output

- Represent system function using partial fraction expansion



M=N=6

$$H(z) = \sum_{k=0}^{N_P} C_k z^{-k} + \sum_{k=1}^{N_P} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_P} \frac{B_k\left(1 - e_k z^{-1}\right)}{\left(1 - d_k z^{-1}\right)\left(1 - d_k^* z^{-1}\right)}$$

$$N_P = M - N$$

Or by pairing the real poles

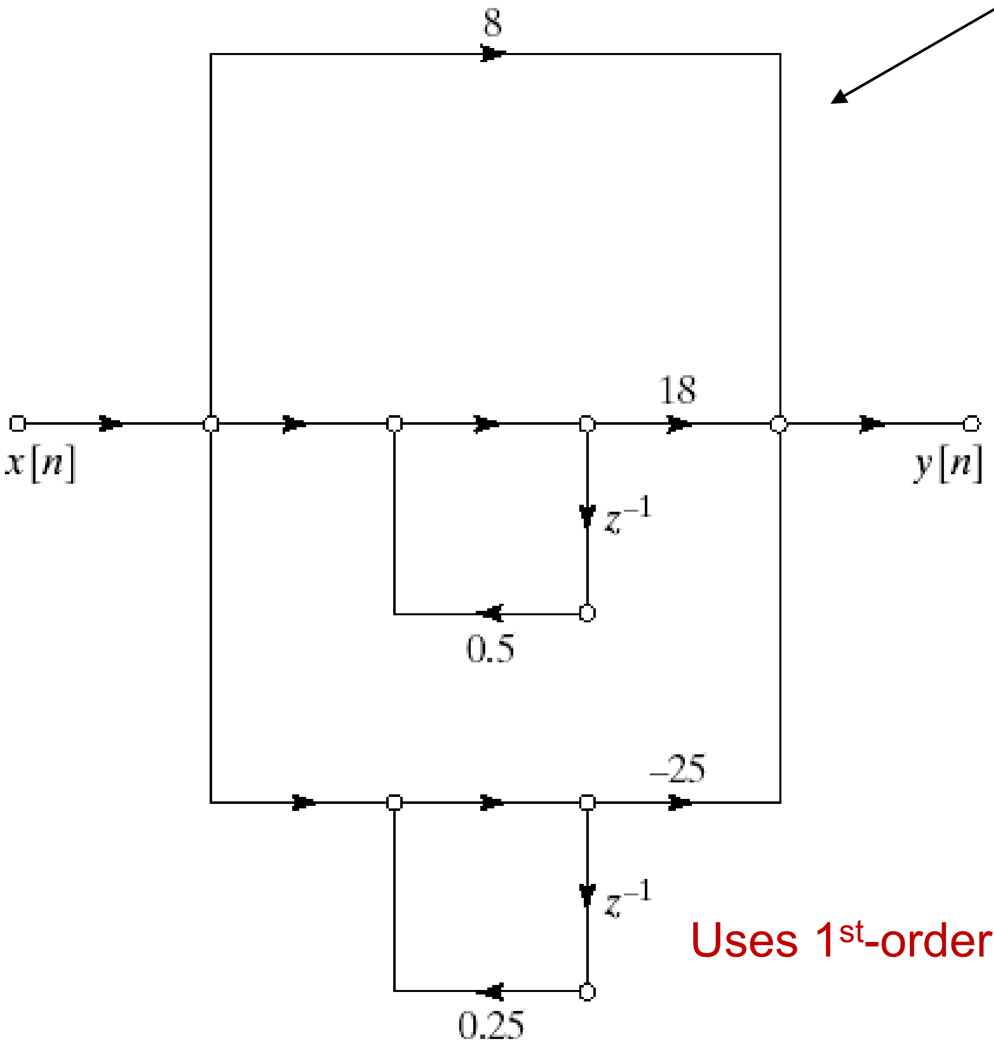
$$H(z) = \sum_{k=0}^{N_P} C_k z^{-k} + \sum_{k=1}^{N_S} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

$$= \mathrm{H_1(z)} + \mathrm{H_2(z)} + \ldots + \mathrm{H_K(z)}$$

Example

- Partial Fraction Expansion

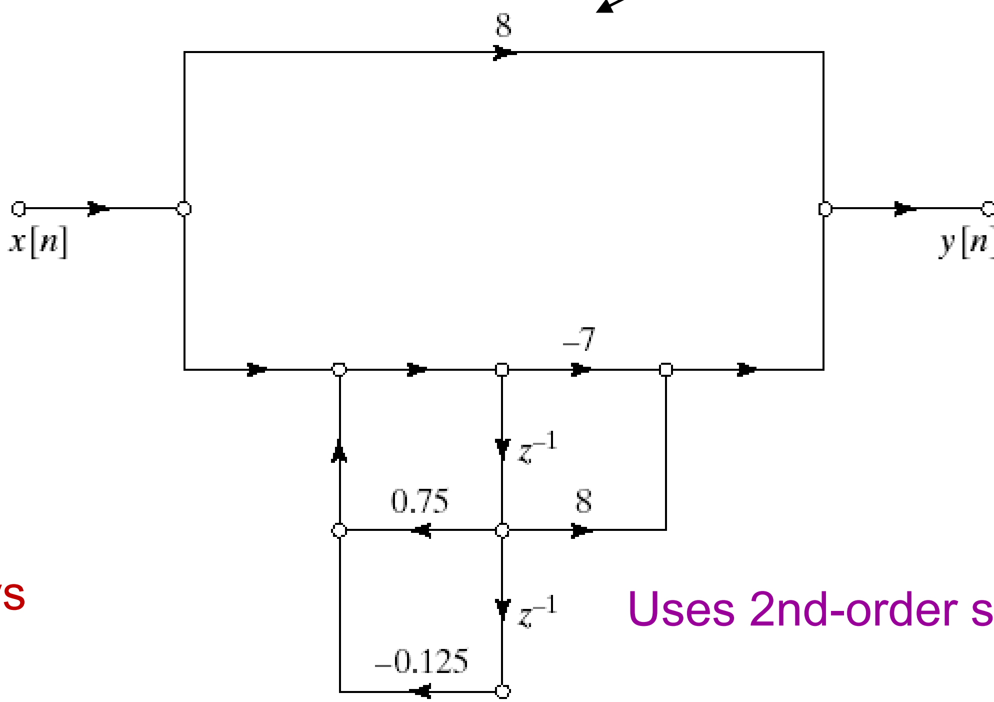
$$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-0.75z^{-1}+0.125z^{-2}}\qquad =8+\frac{18}{\left(1-0.5z^{-1}\right)}-\frac{25}{\left(1-0.25z^{-1}\right)}$$



Uses 1st-order sys

Combine poles to get

$$H(z)=8+\frac{-7+8z^{-1}}{1-0.75z^{-1}+0.125z^{-2}}$$



Uses 2nd-order sys

OUTLINE

Structures For Discrete Time Systems:

- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o **Transposed Forms**
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

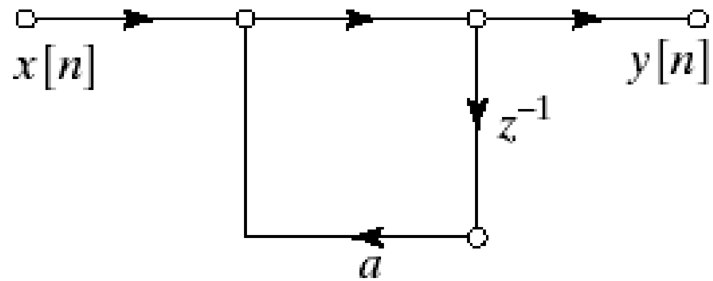
Transposed Forms

- Transposing:
  - Reverse directions of all branches
  - Keep the branch operations unchanged
  - Interchange input and output nodes

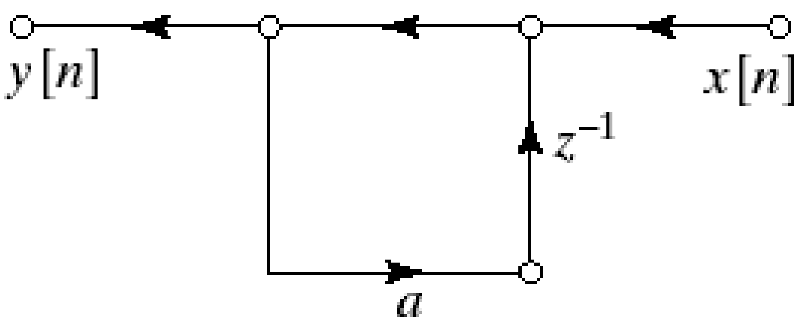
- Linear signal flow graph property:
  - Transposing doesn't change the input-output relation

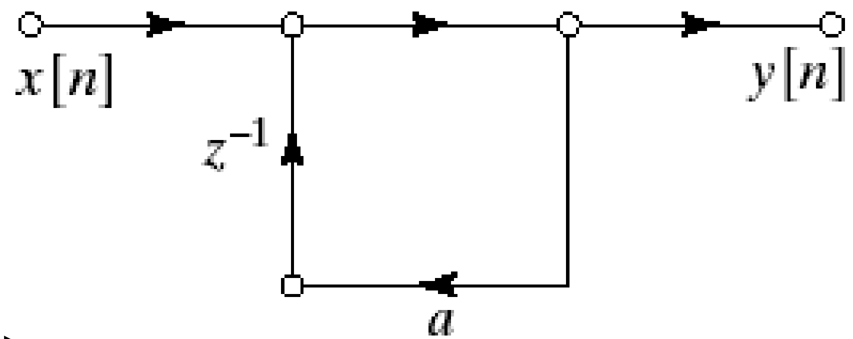
Example 1: 1st order system with no zero

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



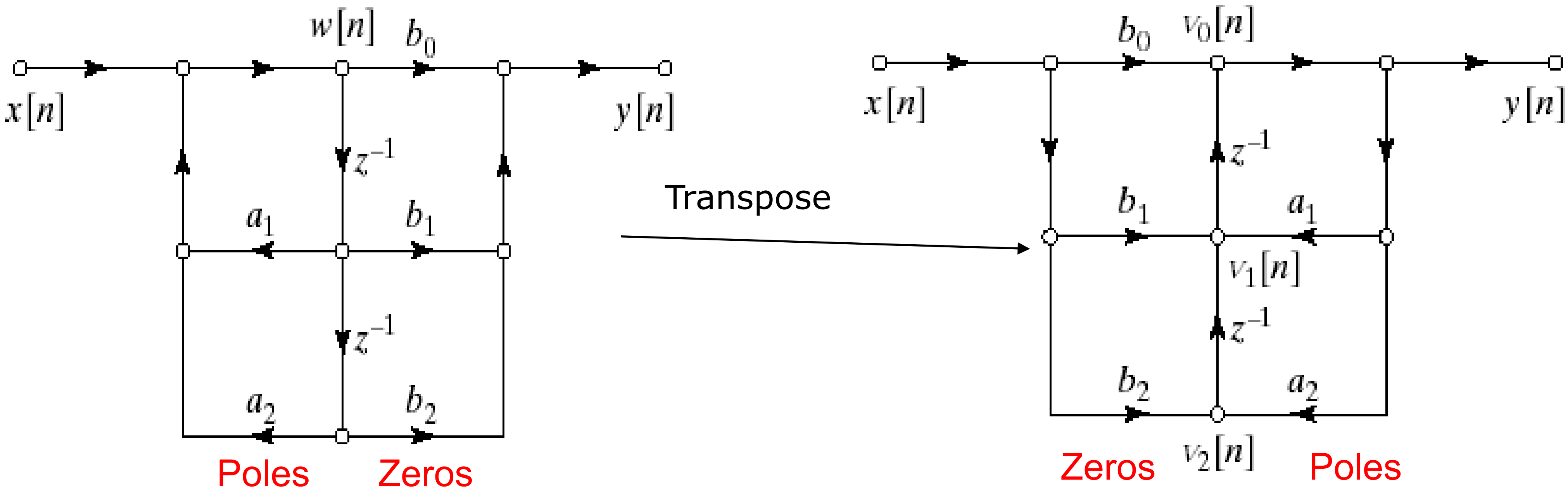
Reverse directions of branches and interchange input and output





Redraw (flip)

Example 2: 2nd order system



Direct form II

Transposed direct form II

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

➔Both have the same system function or difference equation

OUTLINE

Structures For Discrete Time Systems:

- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o Transposed Forms
- o **Basic Structures For FIR Systems**

Finite Precision Numerical Effects:

- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

Basic Structures for FIR Systems

$$\text{FIR}: y[n] = \sum_{k=0}^{M} b_k x[n-k] \Rightarrow H(z) \text{ has only zeros}$$

$$\text{For a causal FIR system}: \ h[n] = \begin{cases} b_n \text{ for } n = 0,1,\ldots,M \\ 0 \quad \text{otherwise} \end{cases}$$

- Direct form $$H(z) = \sum_{n=0}^{M} h[n] z^{-n}$$

- Cascade form $$H(z) = \prod_{k=1}^{M_S} \left(b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}\right)$$

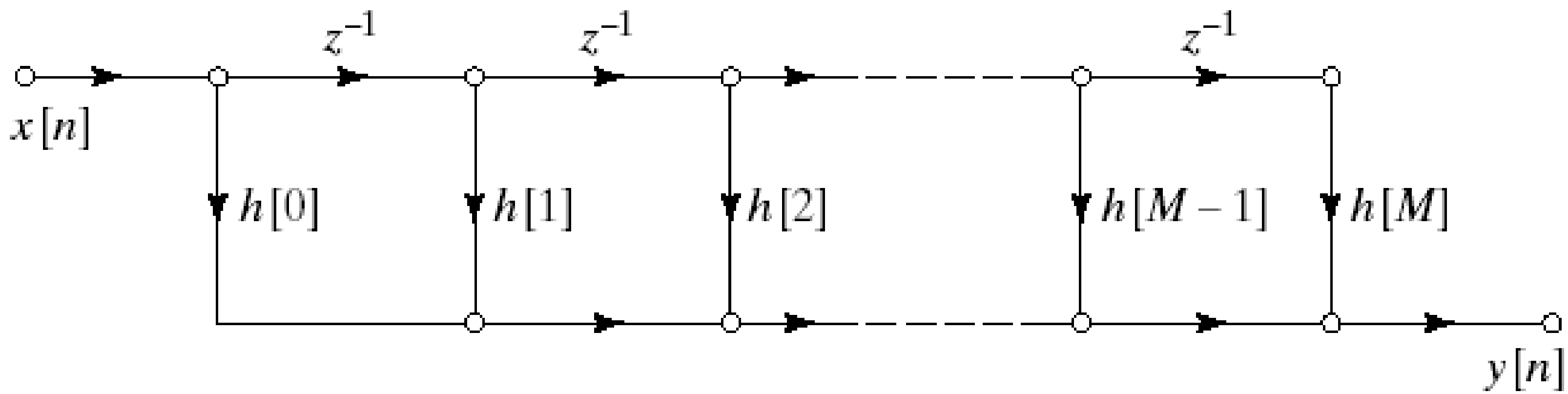
- Linear phase FIR structures

$$\text{Symmetry}: \ h[M-n] = h[n] \qquad \text{n} = 0,1,\ldots,\text{M} \quad (\text{type I or III})$$

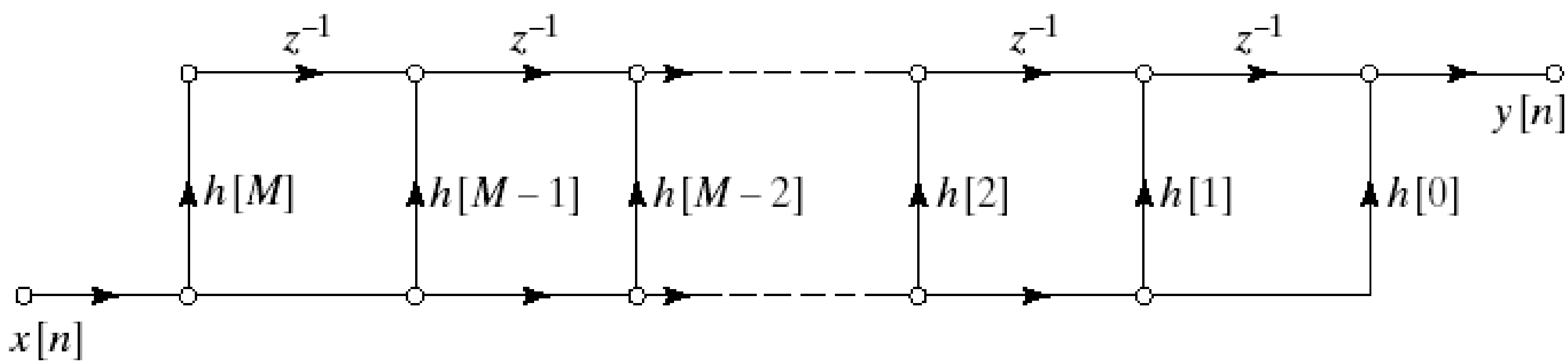
$$\text{AntiSymmetry}: \ h[M-n] = -h[n] \qquad \text{n} = 0,1,\ldots,\text{M} \quad (\text{type II or IV})$$

Basic Structures for FIR Systems: Direct Form

- FIR are special cases of IIR: H(z) has zeros only; no poles
- Direct form structures (delay & multiply)



- Transpose of direct form I gives direct form II (multiple & then delay)

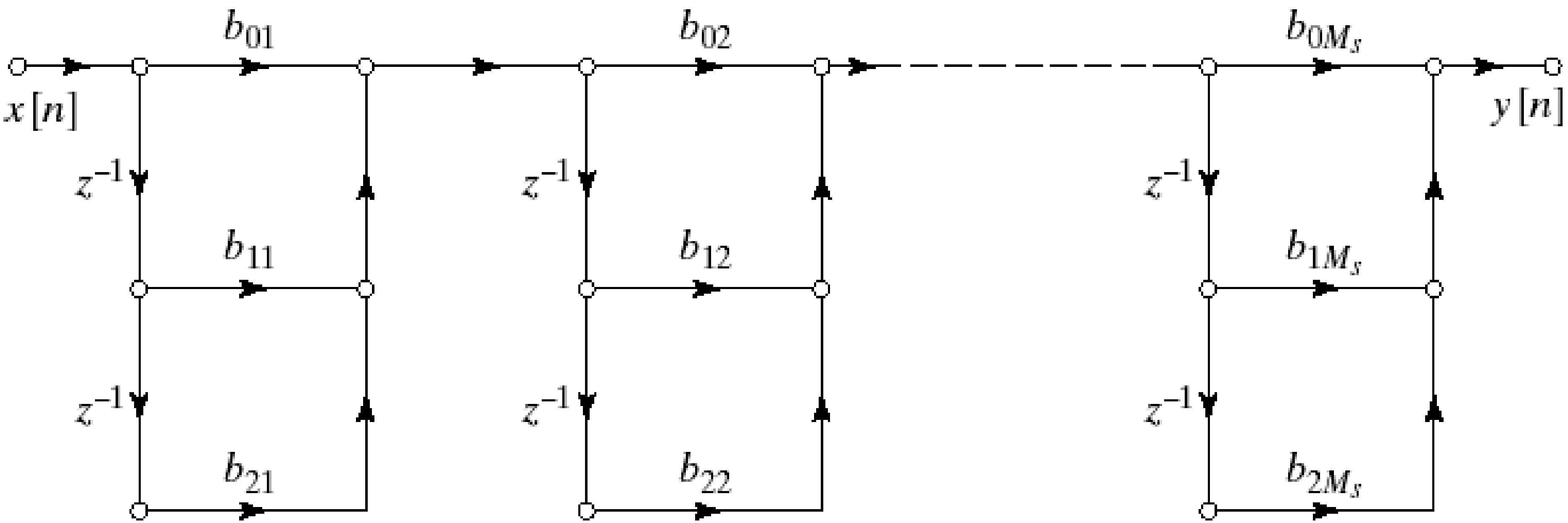


- Both forms are equal for FIR systems

Basic Structures for FIR Systems: Cascade Form

- Obtained by factoring the polynomial system function

$$H(z) = \sum_{n=0}^{M} h[n] z^{-n} = \prod_{k=1}^{M_S} \left(b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2} \right)$$



Basic Structures for FIR Systems: Linear-Phase FIR

- Linear phase filters

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{-j\omega\alpha}$$

- Generalized linear phase filters

$$H(e^{j\omega}) = A(e^{j\omega}) e^{-j\omega\alpha + j\beta}$$

$A(e^{j\omega})$ is a real function of ω,

α and β are real constants

Basic Structures for FIR Systems: Linear-Phase FIR

- If the system is causal and generalized linear-phase

$$h[M-n] = \mp h[n] \quad \text{(symmetry/antisymmetry)}$$

$$M \text{ is an even integer}$$

- Causal: h[n]=0 for n<0 , we get

$$h[n] = 0 \qquad \mathrm{n} < 0 \quad \text{and} \quad \mathrm{n} > \mathrm{M}$$

- **An FIR impulse response of length M+1 is generalized linear phase if they are symmetric**

Basic Structures for FIR Systems: Linear-Phase FIR

- Causal FIR system with generalized linear phase are symmetric:

$$\text{Symmetry:} \quad h[M-n] = h[n] \qquad \text{n} = 0,1,\ldots,\text{M} \quad (\text{type I or III})$$
$$\text{Antisymmetry:} \quad h[M-n] = -h[n] \qquad \text{n} = 0,1,\ldots,\text{M} \quad (\text{type II or IV})$$

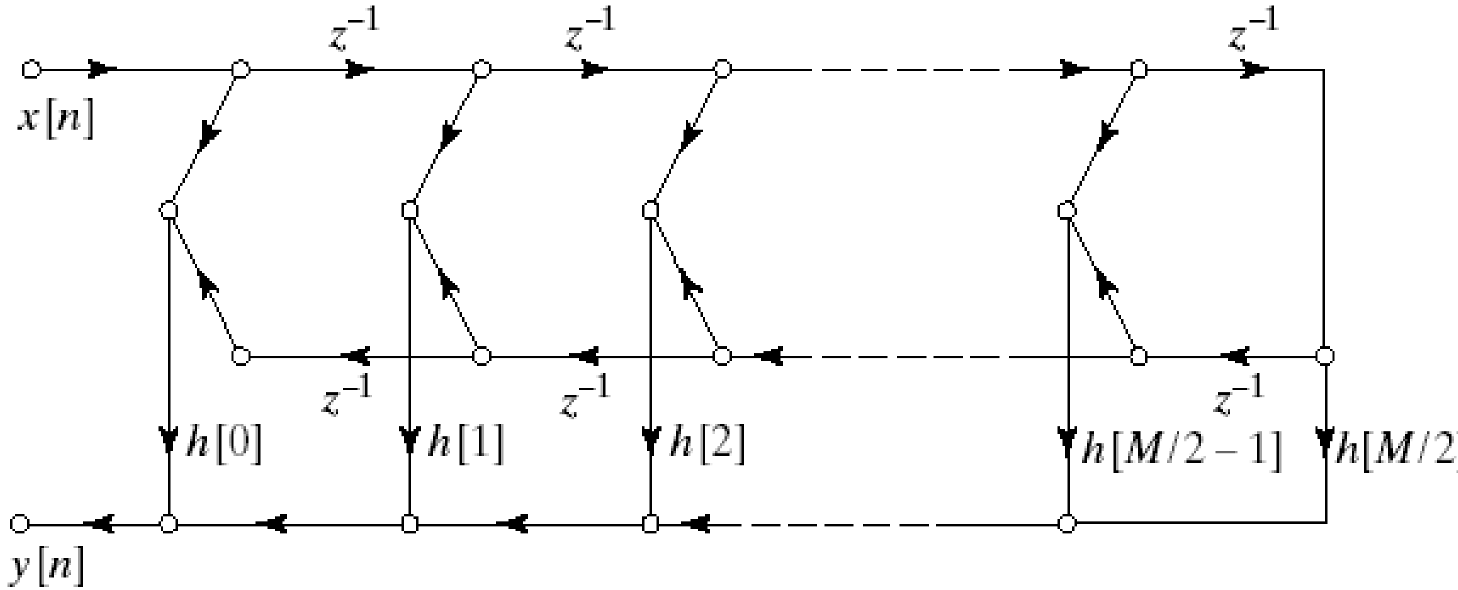
- Symmetry means we can half the number of multiplications

- **Example: For even M and type I or type III systems:**

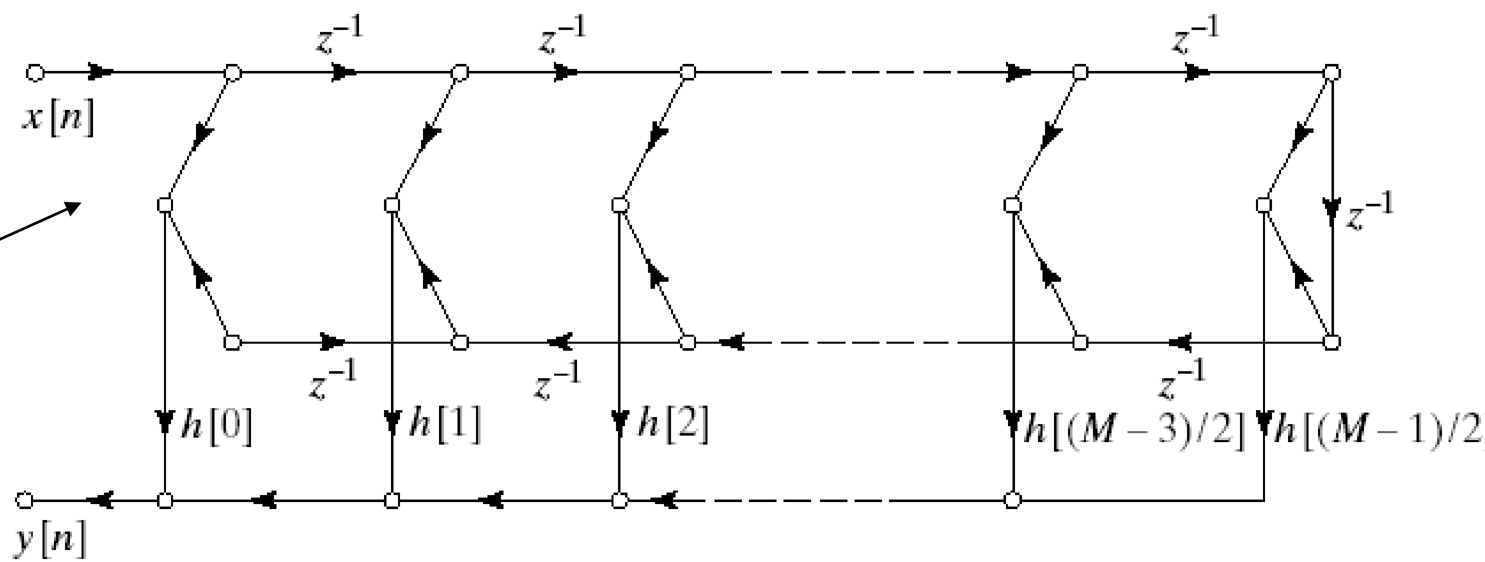
$$y[n] = \sum_{k=0}^{M} h[k]x[n-k] = \sum_{k=0}^{M/2-1} h[k]x[n-k] + h[M/2]x[n-M/2] + \sum_{k=M/2+1}^{M} h[k]x[n-k]$$
$$= \sum_{k=0}^{M/2-1} h[k]x[n-k] + h[M/2]x[n-M/2] + \sum_{k=0}^{M/2-1} h[M-k]x[n-M+k]$$
$$= \sum_{k=0}^{M/2-1} h[k](x[n-k] + x[n-M+k]) + h[M/2]x[n-M/2]$$

Basic Structures for FIR Systems: Linear-Phase FIR

- Structure for even M: we need half the number of coeff.



- Structure for odd integer M



Symmetric

OUTLINE

Structures For Discrete Time Systems:

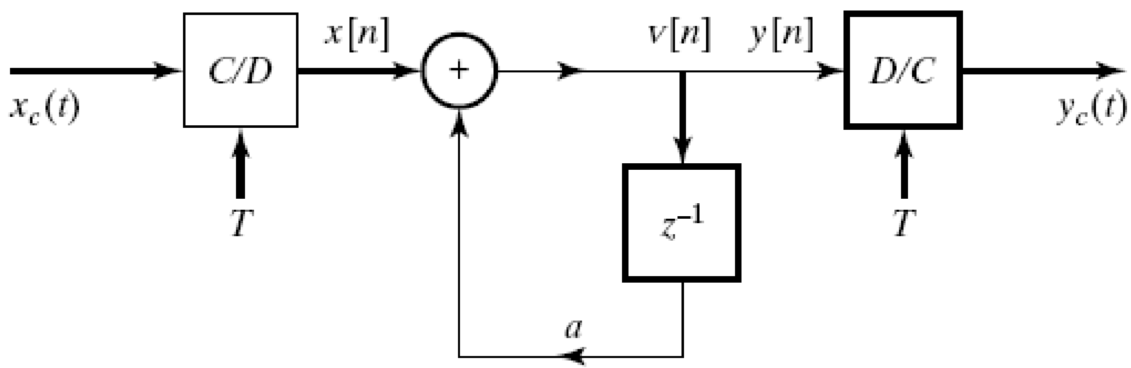
- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o Transposed Forms
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

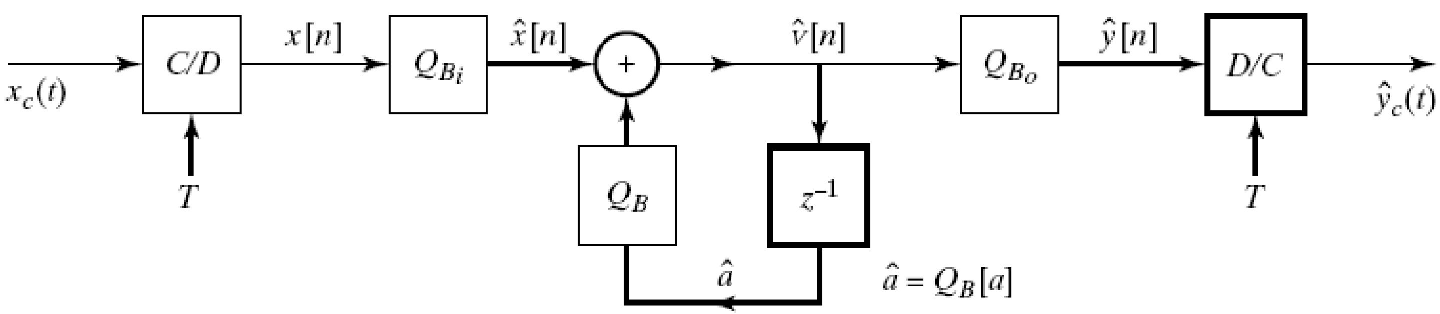
- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

Quantization in Implementing Systems

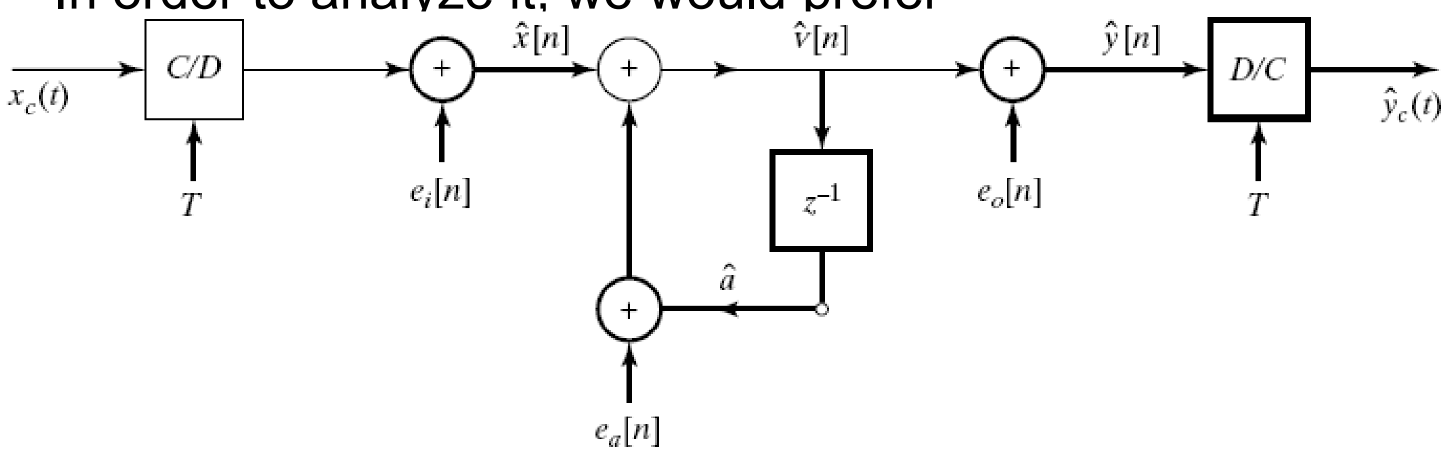
- Consider the following system



- A more realistic model would be



- In order to analyze it, we would prefer



Effects of Coefficient Quantization in IIR Systems

- When the parameters of a rational system are quantized
  - The poles and zeros of the system function move
- If the system structure of the system is sensitive to perturbation of coefficients
  - The resulting system may no longer be stable
  - The resulting system may no longer meet the original specs

➔We need to do a detailed sensitivity analysis

  - Quantize the coefficients and analyze frequency response
  - Compare frequency response to original response

➔We would like to have a general sense of the effect of quantization

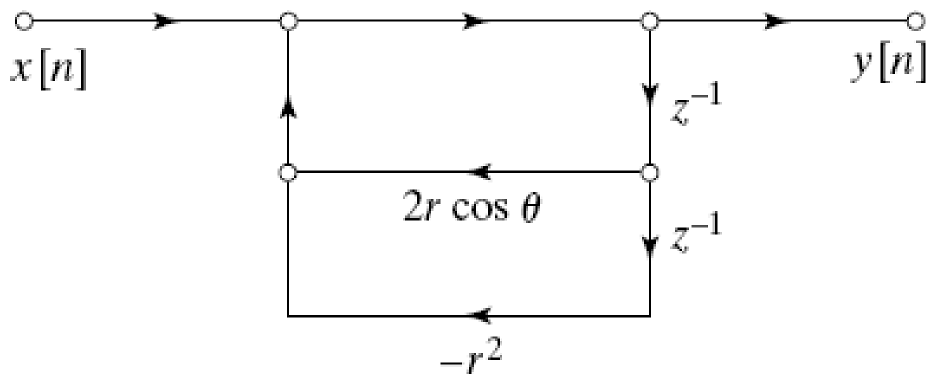
Effects on Roots

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} \xrightarrow{\text{Quantization}} \hat{H}(z) = \frac{\sum_{k=0}^{M} \hat{b}_k z^{-k}}{1 - \sum_{k=1}^{N} \hat{a}_k z^{-k}}$$

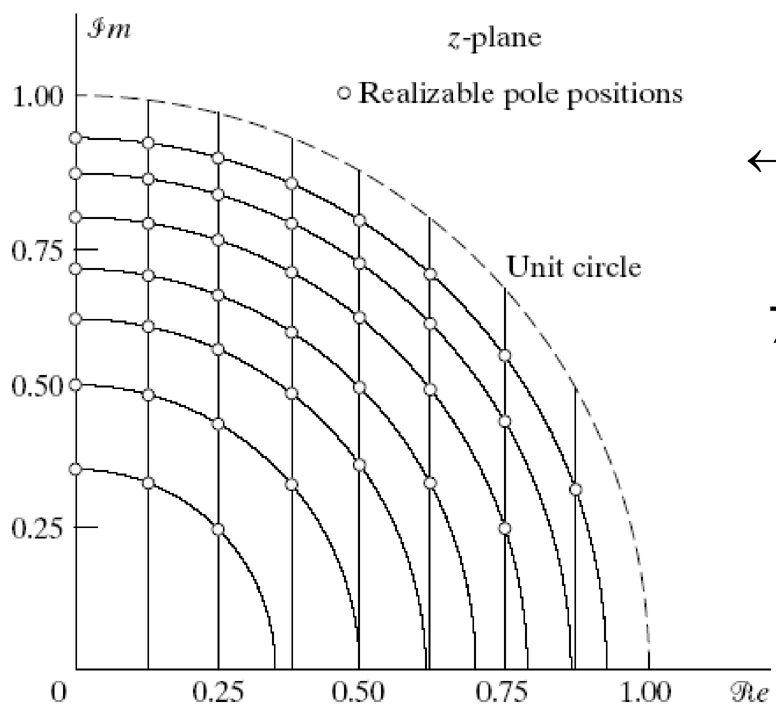
- Each root is affected by quantization errors in ALL coefficient
- Tightly clustered roots can be significantly effected
  - Narrow-bandwidth lowpass or bandpass filters can be very sensitive to quantization noise
- The larger the number of roots in a cluster the more sensitive it becomes
- This is the reason why second order cascade structures are less sensitive to quantization error than higher order system
  - Each second order system is independent from each other

Poles of Quantized Second-Order Sections

- Consider a 2nd order system with complex-conjugate pole pair

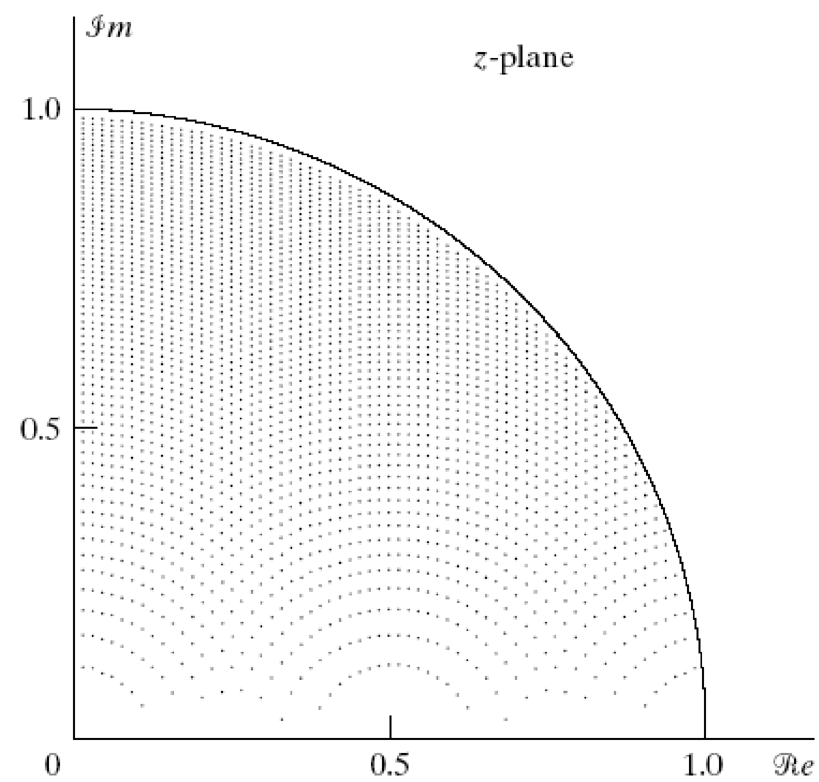


- The pole locations after quantization will be on the grid point



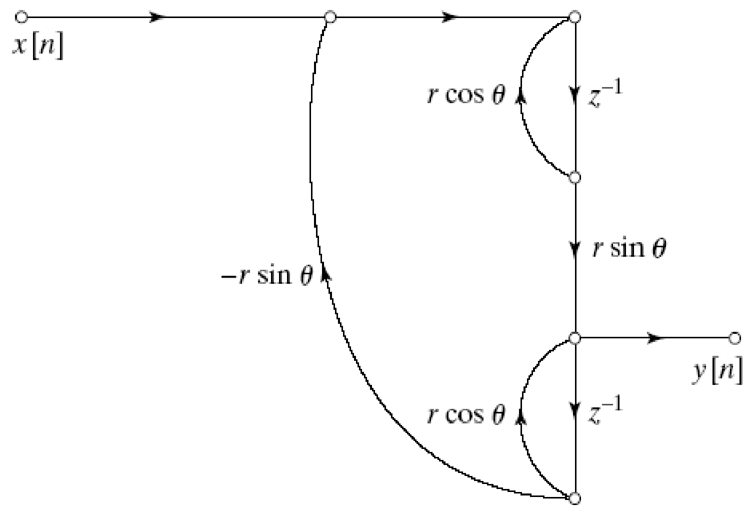
← 3-bits

7-bits →

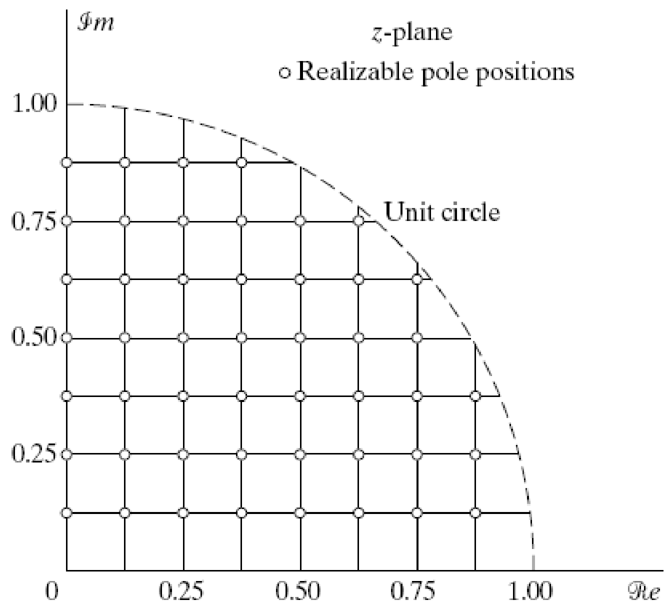


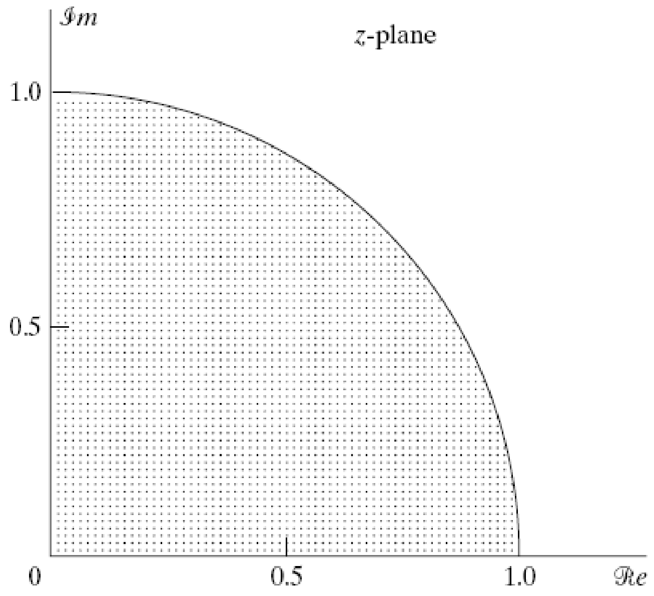
Coupled-Form Implementation of Complex-Conjugate Pair

- Equivalent implementation of the second order system



- But the quantization grid this time is





Effects of Coefficient Quantization in FIR Systems

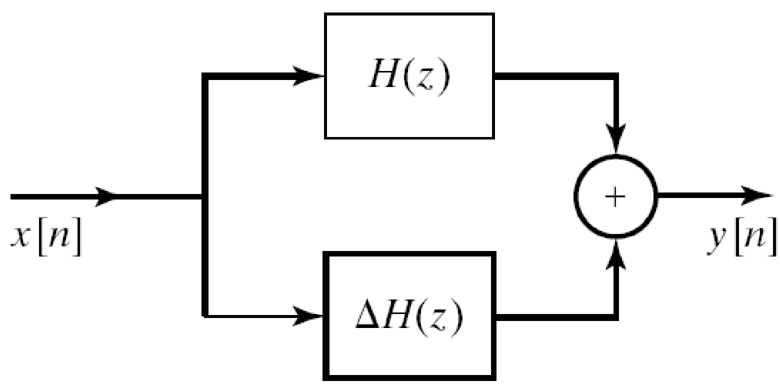
- No poles to worry about only zeros
- Direct form is commonly used for FIR systems

$$H(z) = \sum_{n=0}^{M} h[n] z^{-n}$$

- Suppose the coefficients are quantized

$$\hat{H}(z) = \sum_{n=0}^{M} \hat{h}[n] z^{-n} = H(z) + \Delta H(z) \qquad \Delta H(z) = \sum_{n=0}^{M} \Delta h[n] z^{-n}$$

- Quantized system is linearly related to the quantization error



- Again quantization noise is higher for clustered zeros
- However, most FIR filters have spread zeros

OUTLINE

Structures For Discrete Time Systems:

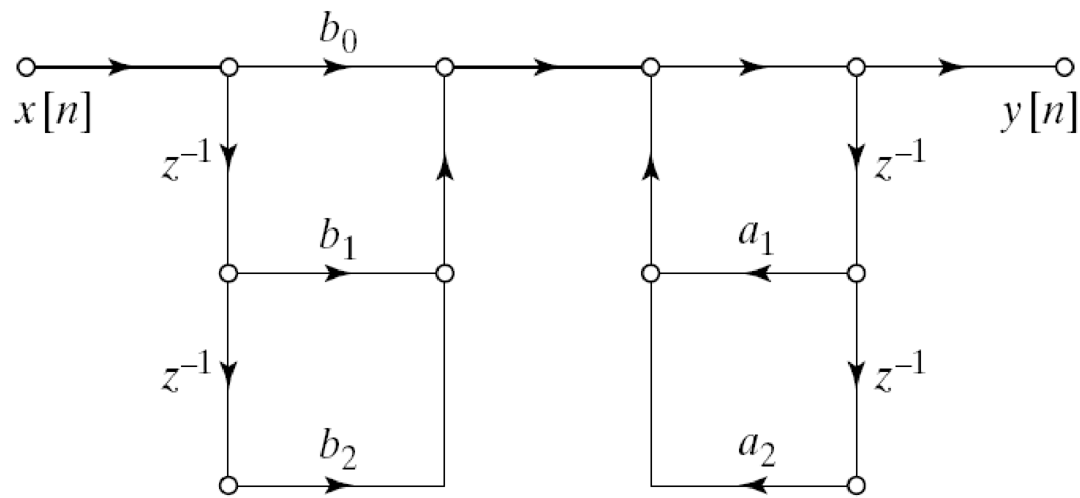
- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o Transposed Forms
- o Basic Structures For FIR Systems

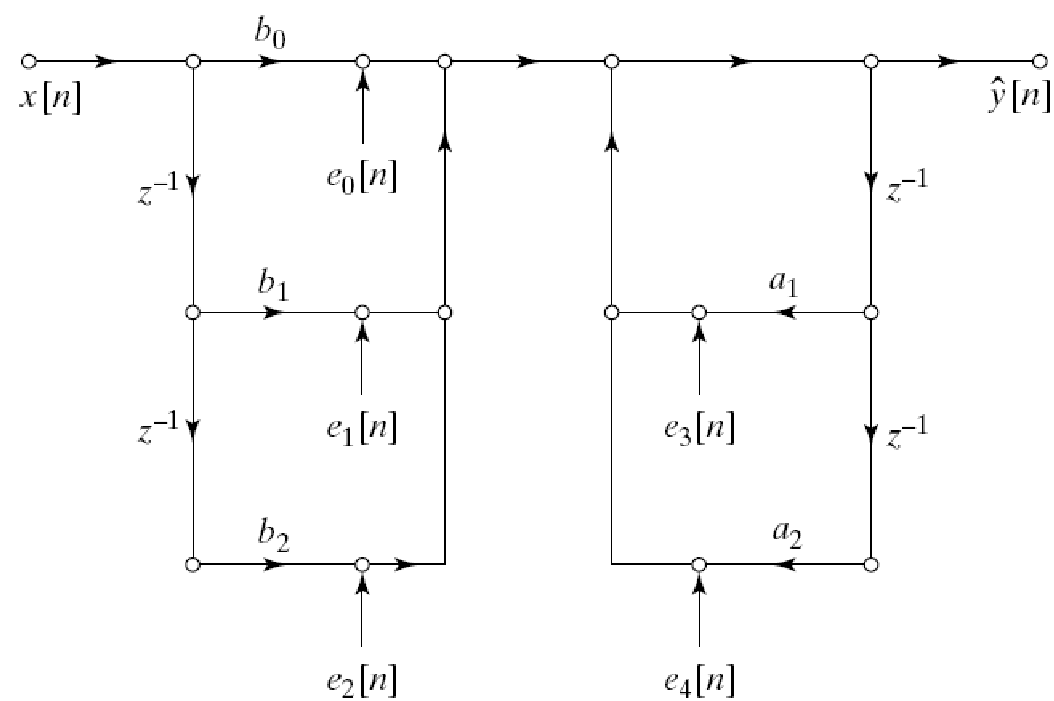
Finite Precision Numerical Effects:

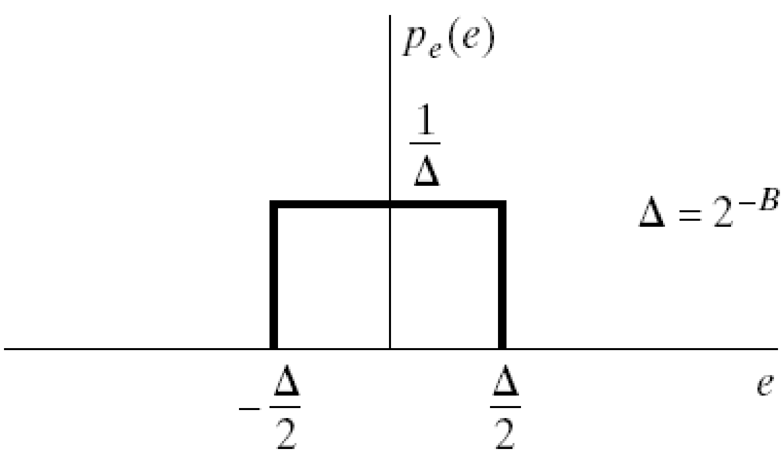
- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

Round-Off Noise in Digital Filters

- Difference equations implemented with finite-precision arithmetic are non-linear systems
- Second order direct form I system
- Model with quantization effect
- Density function error terms for rounding



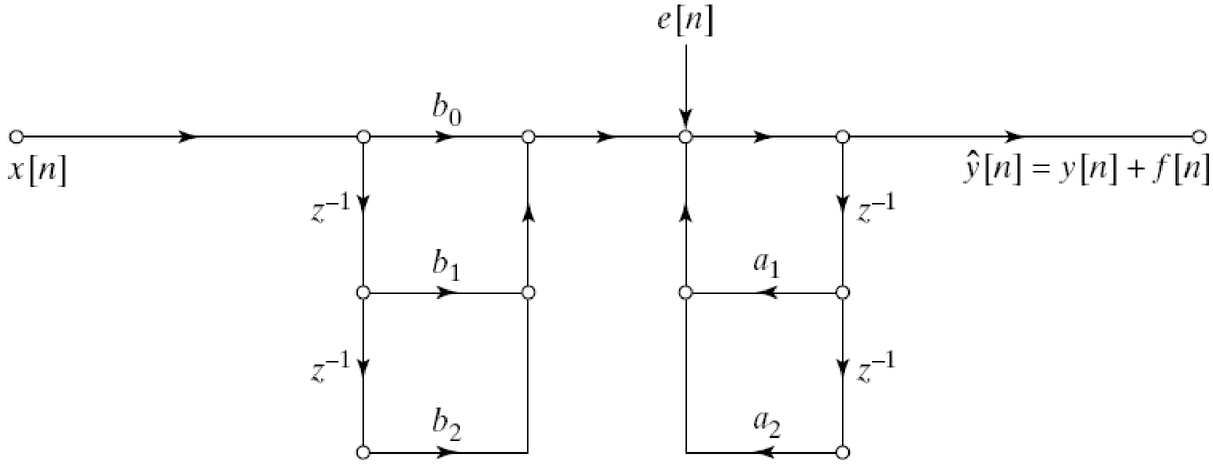




Analysis of Quantization Error

- Combine all error terms to single location to get

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



- The variance of e[n] in the general case is

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

- The contribution of e[n] to the output is

$$f[n] = \sum_{k=1}^{N} a_k f[n-k] + e[n]$$

- The variance of the output error term f[n] is

$$\sigma_f^2 = (M + 1 + N)\frac{2^{-2B}}{12}\sum_{n=-\infty}^{\infty}|h_{ef}[n]|^2$$

$$H_{ef}(z) = 1 / A(z)$$

Round-Off Noise in a First-Order System

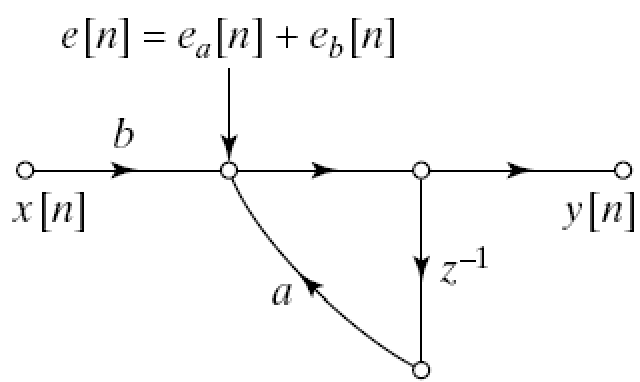
- Suppose we want to implement the following stable system

$$H(z) = \frac{b}{1 - az^{-1}} \qquad |a| < 1$$

- The quantization error noise variance is

$$\sigma_f^2 = (M + 1 + N)\frac{2^{-2B}}{12}\sum_{n=-\infty}^{\infty}|h_{ef}[n]|^2 = 2\frac{2^{-2B}}{12}\sum_{n=0}^{\infty}|a|^{2n} = 2\frac{2^{-2B}}{12}\left(\frac{1}{1-|a|^2}\right)$$

- Noise variance increases as |a| gets closer to the unit circle
- As |a| gets closer to 1 we have to use more bits to compensate for the increasing error



OUTLINE

Structures For Discrete Time Systems:

- o Description of LTI & Block Diagram Representation
- o Direct Form I & II
- o Signal Flow Graph Representation
- o Basic Structures For IIR Systems
- o Transposed Forms
- o Basic Structures For FIR Systems

Finite Precision Numerical Effects:

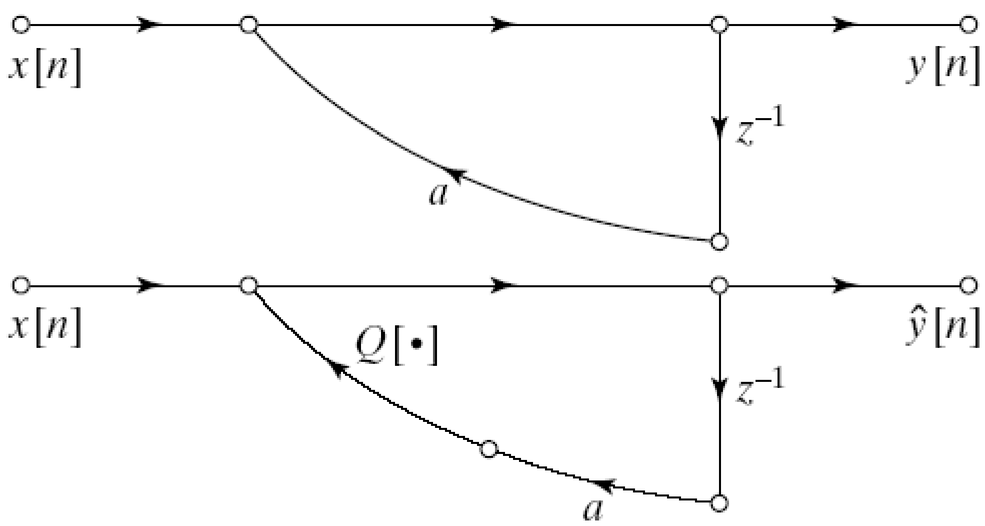
- Quantization in Implementing Systems-Effects in IIR & FIR Systems
- Round Off Noise and Analysis of Quantization Error
- Limit Cycles

Zero-Input Limit Cycles in Fixed-Point Realization of IIR Filters

- For stable IIR systems the output will decay to zero when the input becomes zero
- A finite-precision implementation, however, may continue to oscillate indefinitely
- Nonlinear behavior very difficult to analyze so we sill study by example
- Example: Limit Cycle Behavior in First-Order Systems

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

- Assume x[n] and y[n-1] are implemented by 4 bit registers



Example Cont'd

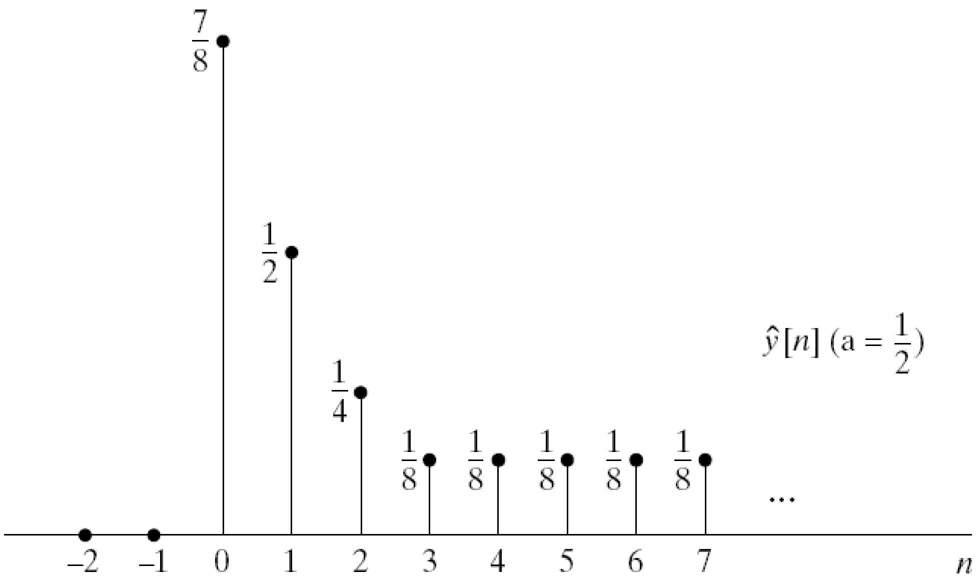
$$y[n] = ay[n-1] + x[n] \qquad |a| < 1$$

- Assume that a=1/2=0.100b and the input is

$$x[n] = \frac{7}{8}\delta[n] = (0.111b)\delta[n]$$

- If we calculate the output for values of n

n	y[n]	Q(y[n])
0	7/8=0.111b	7/8=0.111b
1	7/16=0.011100b	1/2=0.100b
2	1/4=0.010000b	1/4=0.010b
3	1/8=0.001000b	1/8=0.001b
4	1/16=0.00010b	1/8=0.001b



- A finite input caused an oscillation with period 1

Example: Limit Cycles due to Overflow

- Consider a second-order system realized by

$$\hat{y}[n] = x[n] + Q(a_1\hat{y}[n-1]) + Q(a_2\hat{y}[n-2])$$

  - Where Q() represents two's complement rounding
  - Word length is chosen to be 4 bits
- Assume a_1=3/4=0.110b and a_2=-3/4=1.010b
- Also assume

$$\hat{y}[-1] = 3/4 = 0.110b \text{ and } \hat{y}[-2] = -3/4 = 1.010b$$

- The output at sample n=0 is

$$\begin{aligned}\hat{y}[0] &= 0.110b \times 0.110\text{b} + 1.010b \times 1.010\text{b} \\ &= 0.100100\text{b} + 0.100100\text{b}\end{aligned}$$

- After rounding up we get

$$\hat{y}[0] = 0.101\text{b} + 0.101\text{b} = 1.010\text{b} = \text{-}3/4$$

- Binary carry overflows into the sign bit changing the sign
- When repeated for n=1

$$\hat{y}[0] = 1.010\text{b} + 1.010\text{b} = 0.110 = 3/4$$

Avoiding Limite Cycles

- Desirable to get zero output for zero input: Avoid limit-cycles
- Generally adding more bits would avoid overflow
- Using double-length accumulators at addition points would decrease likelihood of limit cycles
- Trade-off between limit-cycle avoidance and complexity
- FIR systems cannot support zero-input limit cycles