ECE 4213/5213

MODULE 8

Index

1. Intro to Digital Filter Structures 8.1
2. Basic Block Diagram Elements 8.2
3. Block Diagram Analysis ....8.3 – 8.5
4. Delay-Free Loops...8.5 – 8.7
5. Canonical, Direct, and Equivalent Structures 8.8
6. Transpose of a Block Diagram..8.9 – 8.11
7. FIR Direct Form 8.12
8. FIR Cascade Form..8.13 – 8.14
9. FIR Polyphase Realizations...8.14 – 8.17
10. Linear Phase FIR Structures 8.18
11. IIR Direct Form I ....8.19 – 8.22
12. IIR Direct Form II ....8.22 – 8.25
13. Cascade and Parallel IIR Structures ....8.26 – 8.30
14. IIR All-Pass Structures ....8.31 – 8.33
15. Lattice Filters...8.34 – 8.65
16. Introduction to Lattice Structures ....8.34 – 8.37
17. FIR Lattice Structures ....8.38 – 8.43
18. Schur-Cohn Stability Test 8.44
19. All-Pole IIR Lattice Structures...8.45 – 8.51
20. General IIR Lattice Structures (Gray-Markel Method).......................8.51 – 8.65
21. Matlab Routines for Lattice Structures..8.65

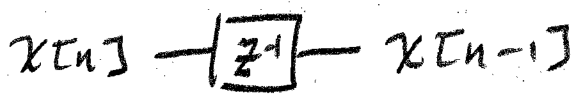
Structures

- "Structure" or "Form" refers to the order of operations when a discrete-time system is implemented.
  - i.e., the order of the multiplies and adds.
- Implementation could be directly in hardware or could be in software on a general purpose CPU or on a DSP processor.
- Different structures (different forms) will generally have different characteristics when implemented in finite precision arithmetic.
  - Different hardware complexity (number of registers, multipliers, adders required)
  - Different computation delay (minimum latency)
  - Different roundoff & quantization effects
  - Different coefficient sensitivities.

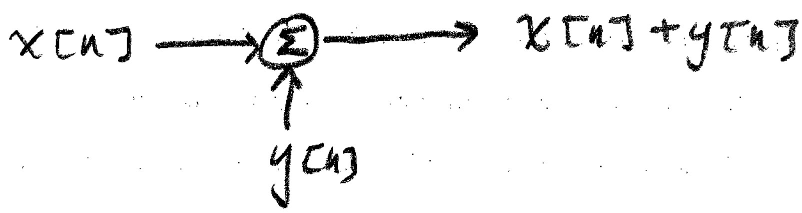
NOTE: Fixed point arithmetic is usually used. CPU clock speeds have recently become fast enough to support floating point, but it is expensive compared to fixed point.

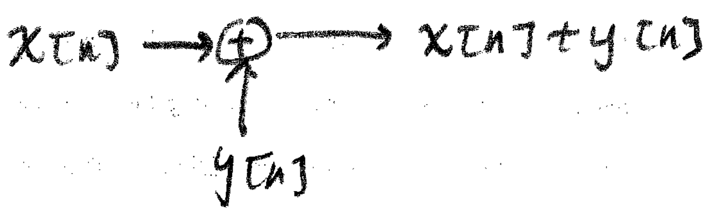
Basic elements:

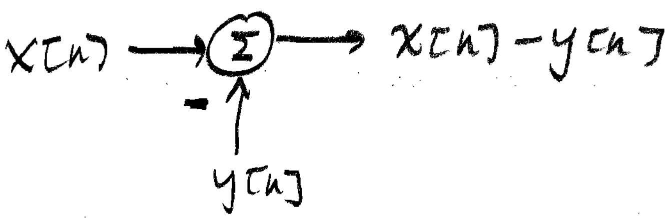
- unit delay:



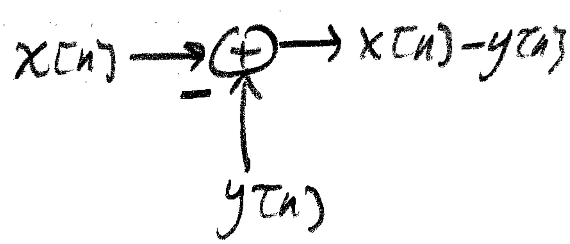
- adder:



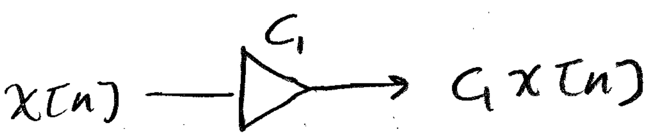




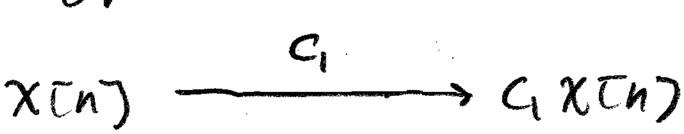
or



- multiplier:



- or -



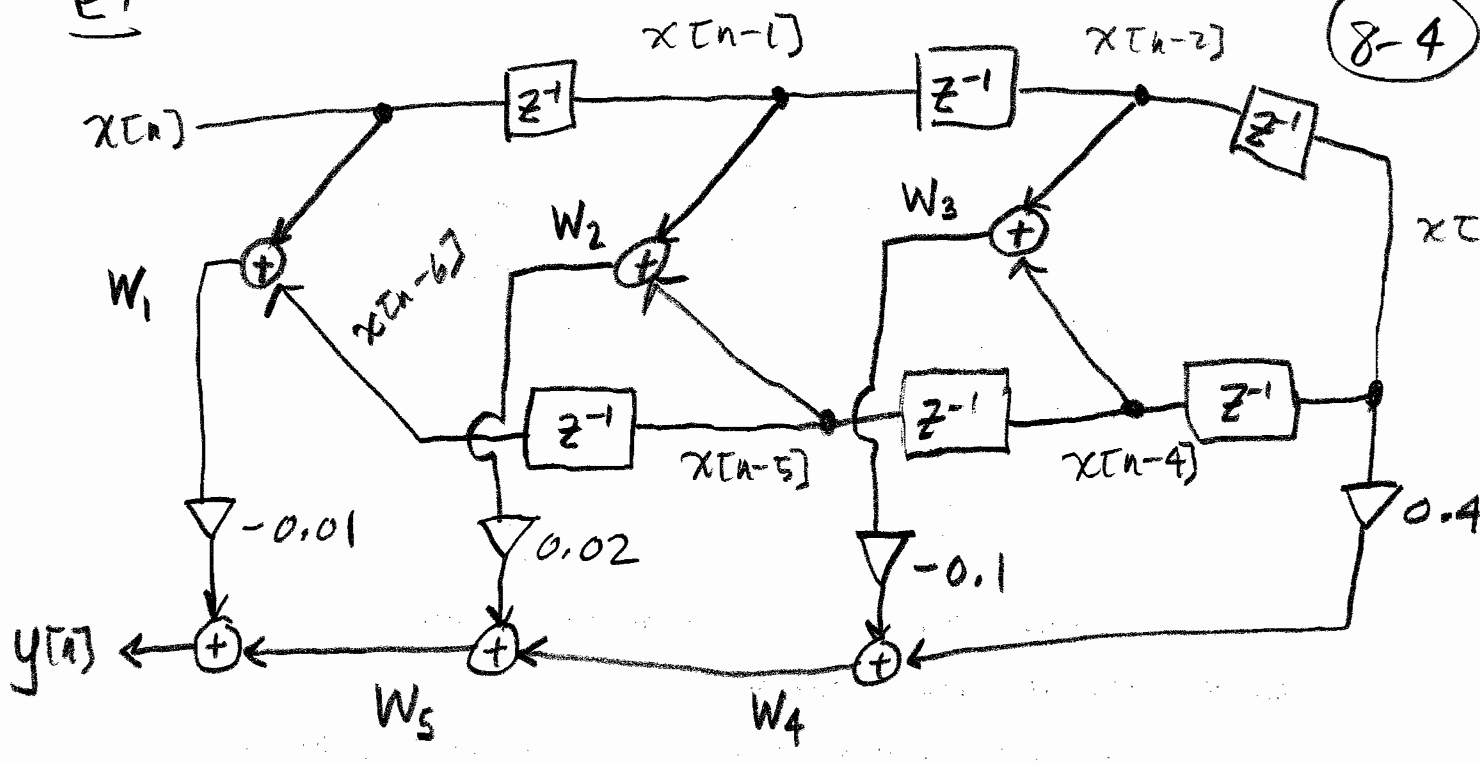
Block Diagram Analysis:

- Given a block diagram or signal flow graph, you may be asked to find the frequency response, transfer function, impulse response, or I/O relation.

- The general procedure is to
  1. Assign an intermediate variable name to the output of each adder.
     → This results in a system of simultaneous equations.
  2. Combine equations to eliminate the intermediate variables.

Ex

8-4



$$W_1 = x[n] + x[n-6] \quad (1)$$

$$W_2 = x[n-1] + x[n-5] \quad (2)$$

$$W_3 = x[n-2] + x[n-4] \quad (3)$$

$$W_4 = -0.1 W_3 + 0.4 x[n-3] \quad (4)$$

$$W_5 = 0.02 W_2 + W_4 \quad (5)$$

$$y[n] = -0.01 W_1 + W_5 \quad (6)$$

(3) → (4) : $W_4 = -0.1 x[n-2] - 0.1 x[n-4] + 0.4 x[n-3] \quad (7)$

(2), (7) → (5) : $W_5 = 0.02 x[n-1] + 0.02 x[n-5] - 0.1 x[n-2] - 0.1 x[n-4] + 0.4 x[n-3] \quad (8)$

(1), (8) → (6) :

$$y[n] = -0.01 x[n] - 0.01 x[n-6] + 0.02 x[n-1] + 0.02 x[n-5] - 0.1 x[n-2] - 0.1 x[n-4] + 0.4 x[n-3]$$

$$= -0.01 x[n] + 0.02 x[n-1] - 0.1 x[n-2] + 0.4 x[n-3] - 0.1 x[n-4] + 0.02 x[n-5] - 0.01 x[n-6]$$

$$h[n] = -0.01\delta[n] + 0.02\delta[n-1] - 0.1\delta[n-2] + 0.4\delta[n-3] - 0.1\delta[n-4] + 0.02\delta[n-5] - 0.01\delta[n-6]$$

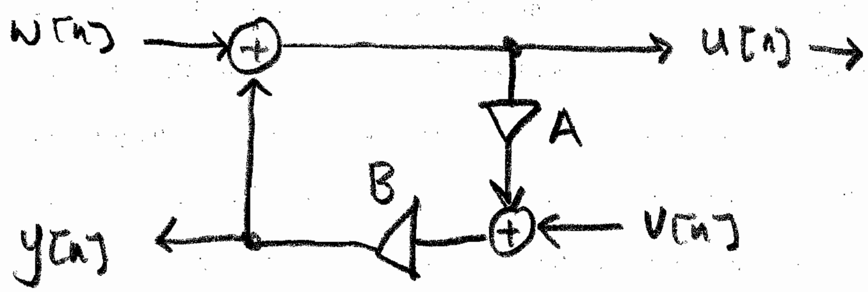
Symmetry

$$= [-0.01 \quad 0.02 \quad -0.1 \quad 0.4 \quad -0.1 \quad 0.02 \quad -0.01]$$

$N = 6$ (order)
Length $= 7$

$\rightarrow$ N even, even symmetry : Type I FIR.

Delay-Free Loops:



$$y[n] = B\big(v[n] + A(w[n] + y[n])\big)$$

- This can't be implemented directly because the value "y[n]" is needed in order to compute y[n].

- If a block diagram or signal flow graph contains delay-free loops, they must be removed before the algorithm can be implemented in the form given by the block diagram.

- In this example, we can multiply out to get

$$y[n] = Bv[n] + ABw[n] + ABy[n]$$

$$(1-AB)\,y[n] = Bv[n] + ABw[n]$$

$$y[n] = \frac{B}{1-AB}v[n] + \frac{AB}{1-AB}w[n].$$

- The value $u[n]$ from the original diagram may be needed to compute $v[n]$.

$$u[n] = w[n] + y[n]$$

$$= w[n] + \frac{AB}{1-AB}w[n] + \frac{B}{1-AB}v[n] \quad (*)$$

So $\quad y[n] = u[n] - w[n] \qquad (**)$

$\longrightarrow$

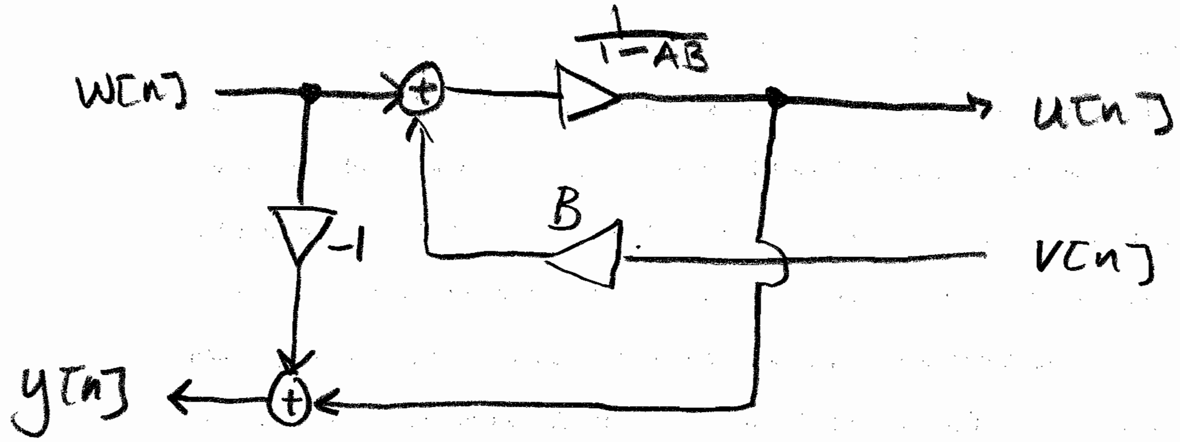
8-7

(*): $u[n] = \frac{1-AB}{1-AB} w[n] + \frac{AB}{1-AB} w[n] + \frac{B}{1-AB} v[n]$

$= \frac{1-AB+AB}{1-AB} w[n] + \frac{B}{1-AB} v[n]$

$= \frac{1}{1-AB} w[n] + \frac{B}{1-AB} v[n] \quad (***)$

- Now (***) and (**) can be used to redraw the algorithm without the delay-free loop:



(Form)
Canonical Structure : the number of delays in the diagram is equal to the order of the difference equation.

Direct Form : the multiplier gains in the diagram are equal to the coefficients in the numerator and denominator polynomials of $H(z)$.

Equivalent structures : have the same transfer function.

Transpose of a diagram:

- An equivalent structure can be obtained by:

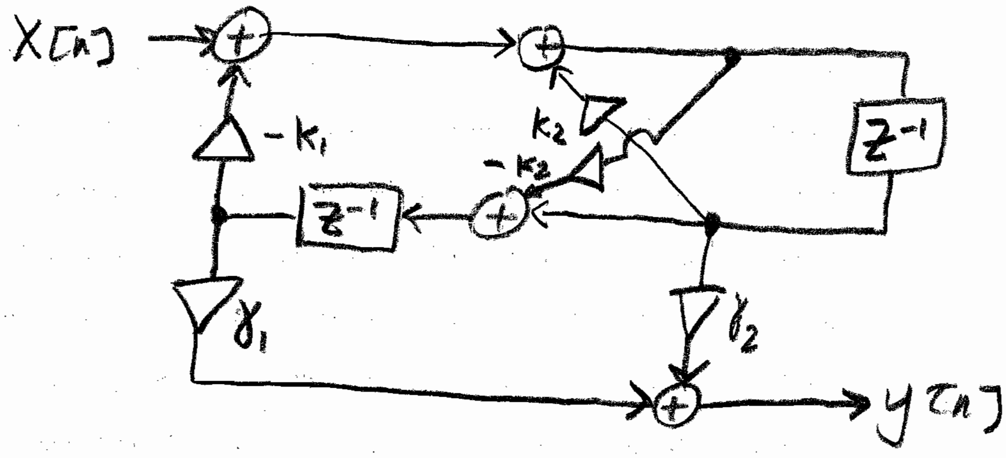
1. reverse the direction of signal flow in all paths.

2. - Change pickoff nodes (taps) to adders
   - Change adders to pickoff nodes.

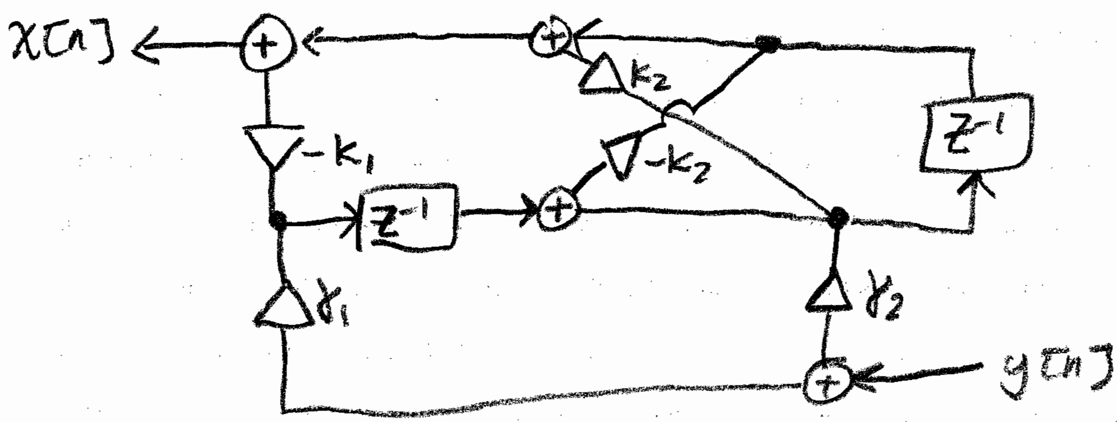
3. Switch the names of the input and output.

Transpose Example:

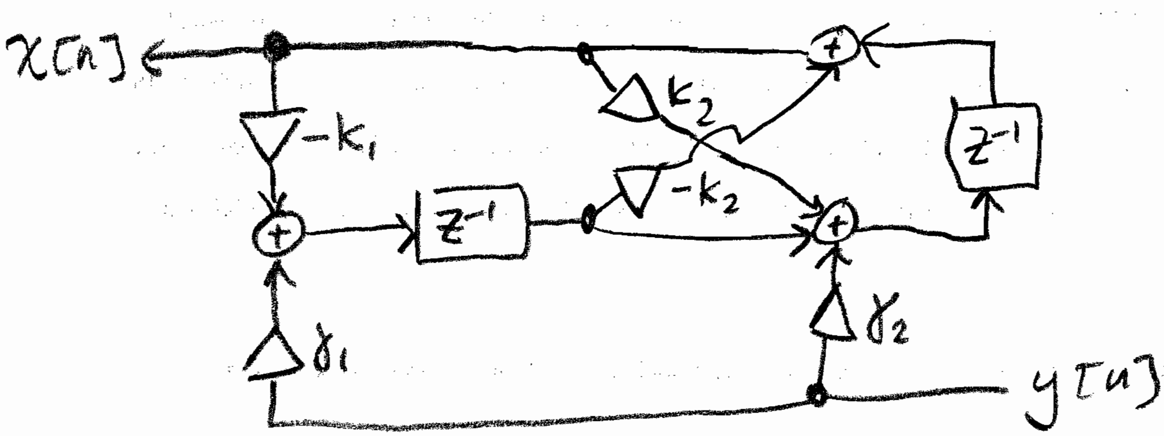
Original:



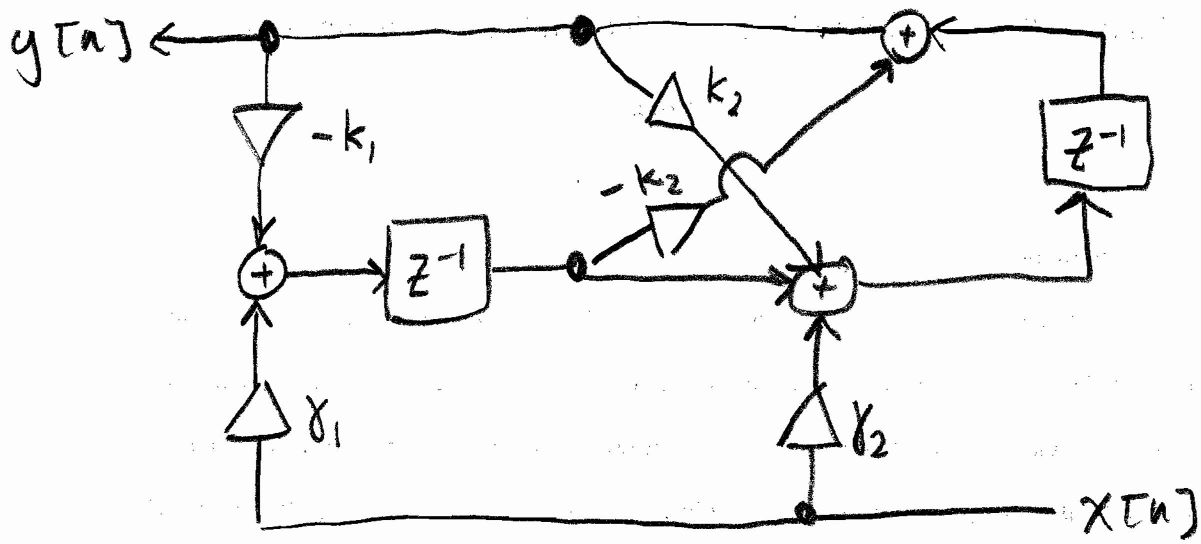
① Reverse flow; the diagram no longer makes sense at this point.



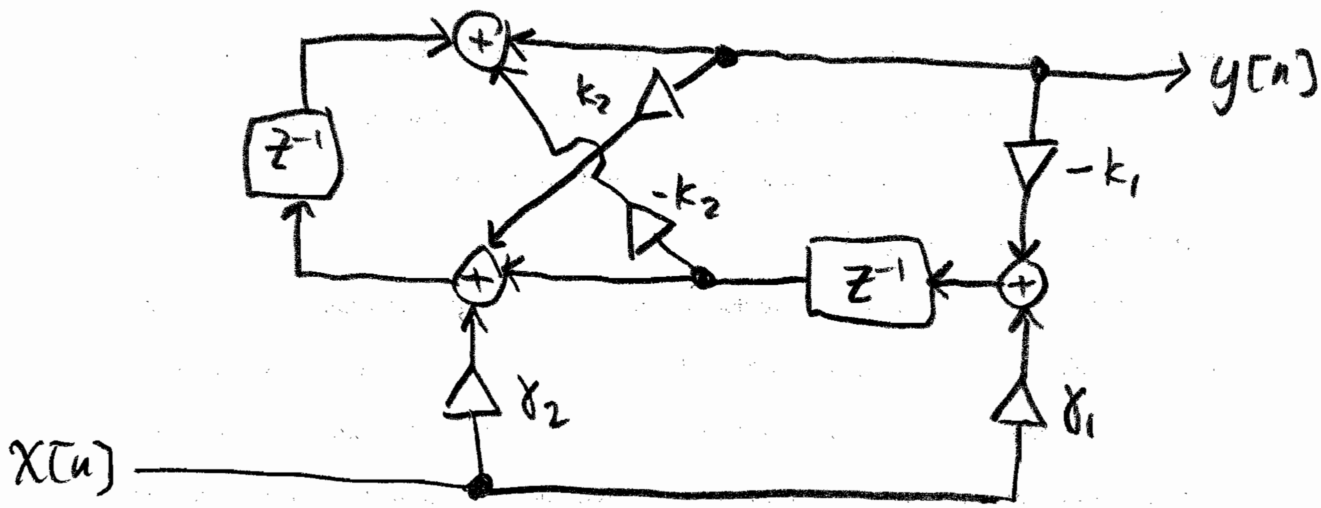
② pickoffs to adders, adders to pickoffs:



③ Change Names:



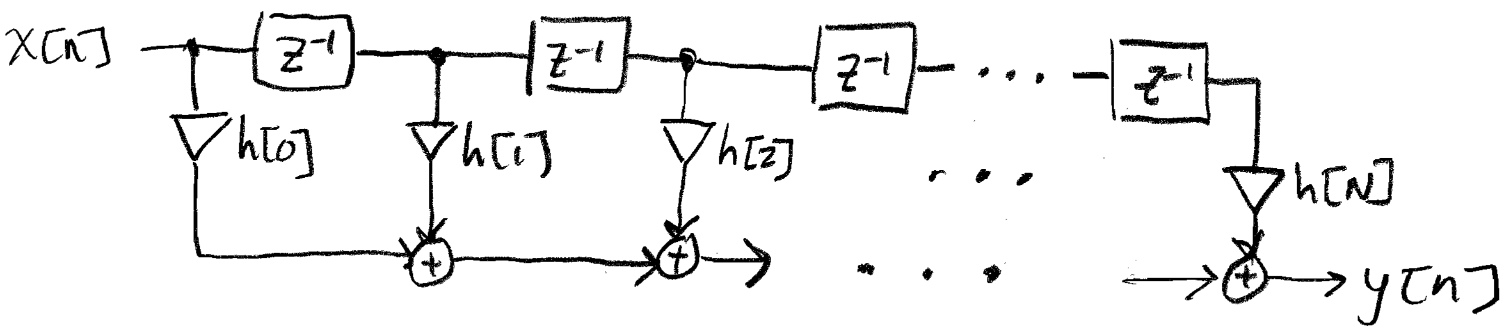
- For clarity, you usually redraw by flipping (transposing) left and right:



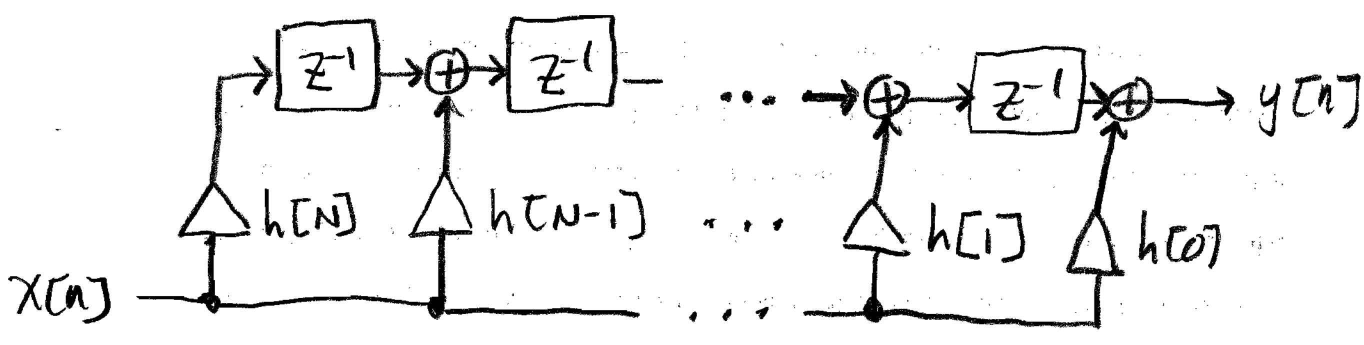
FIR Direct Form:

$$y[n] = \sum_{k=0}^{N} h[k]\, x[n-k]$$

$$= h[0]x[n] + h[1]x[n-1] + \dots + h[N]x[n-N]$$



Transpose:



<u>NOTE:</u> these forms are both direct and canonical.

FIR Cascade Form:

- Any FIR filter $H(z)$ can be factored into a product of first order "zero terms":

$$H(z) = h[0] \prod_{k=1}^{N} (1 - \alpha_k z^{-1})$$

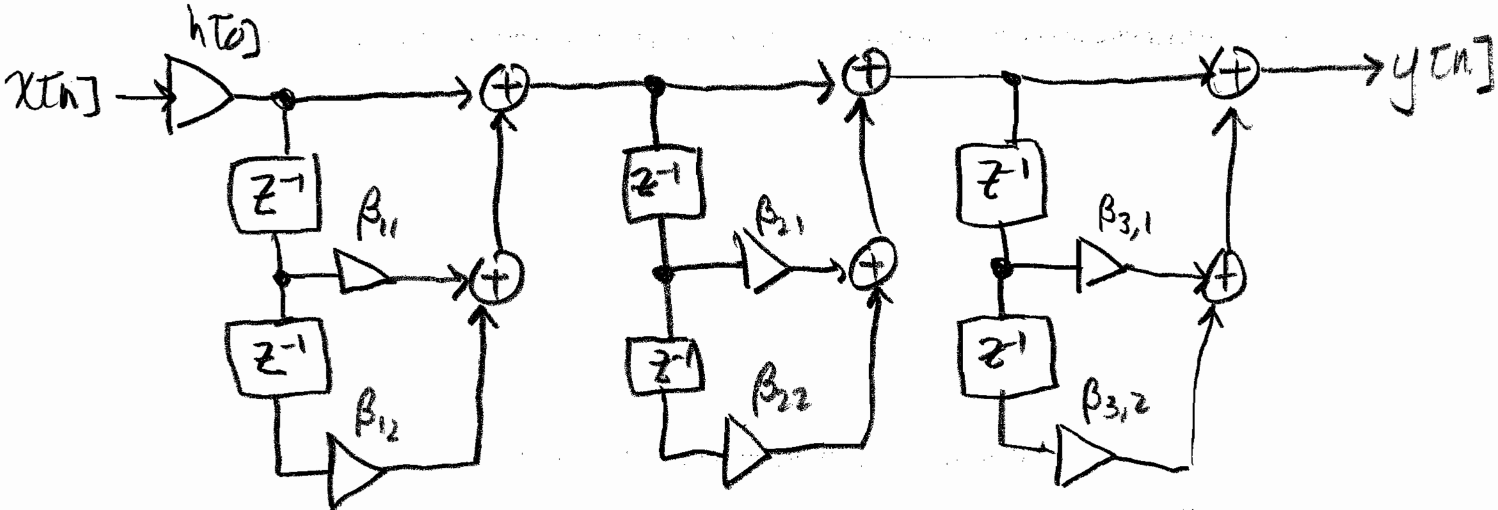
- This could be implemented directly as a cascade of N first order sections.
- But what is more commonly done is to group pairs of zeros together and implement as a cascade of 2nd-order sections:

$$H(z) = h[0] \prod_{k=0}^{K} (1 + \beta_{1,k} z^{-1} + \beta_{2,k} z^{-2}) \qquad (*)$$

NOTE: for a real-valued impulse response, complex zeros occur in conjugate pairs and it is common to group them together.

NOTE: in (*) above, one or more of the sections may be of first order, in which case $\beta_{2,k} = 0$ for those sections

EX: 6^{th}-order cascade form FIR structure:



Polyphase FIR Realization

- The idea is that we divide the impulse response coefficients into groups or "phases" and $H(z)$ accordingly.

EX: two phases; $N=7$, length $=8$

$$h[n] = h[0]\delta[n] + h[1]\delta[n-1] + h[2]\delta[n-2] + h[3]\delta[n-3] + h[4]\delta[n-4] + h[5]\delta[n-5] + h[6]\delta[n-6] + h[7]\delta[n-7].$$

$\longrightarrow$

8-15

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7}$$

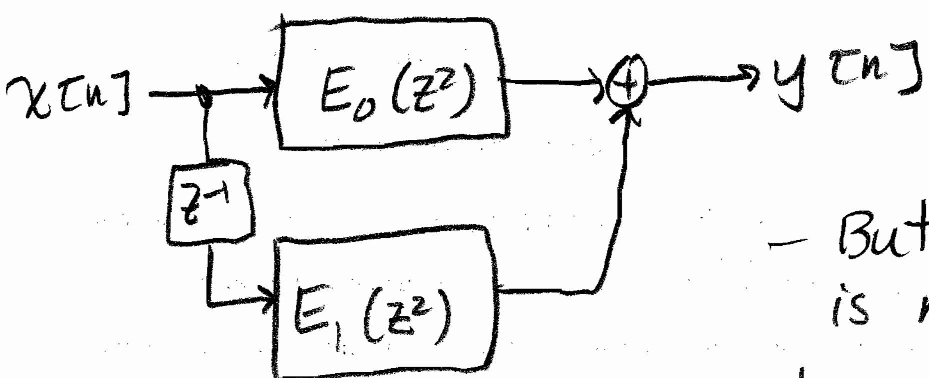
$$= \left(h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6}\right) + \left(h[1]z^{-1} + h[3]z^{-3} + h[5]z^{-5} + h[7]z^{-7}\right)$$

$$= \left(h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6}\right) + z^{-1}\left(h[1] + h[3]z^{-2} + h[5]z^{-4} + h[7]z^{-6}\right)$$

- Let $E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3}$

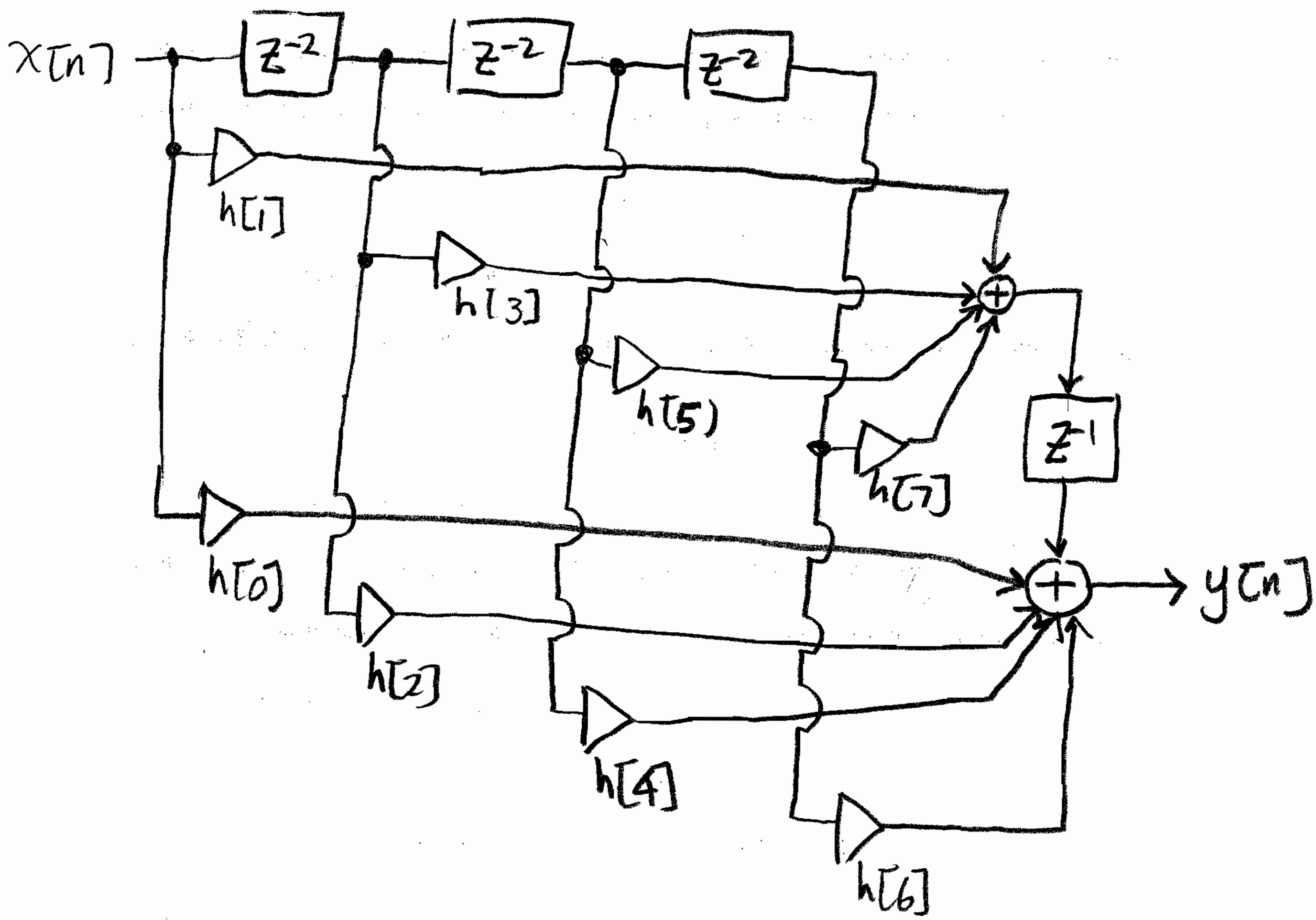
  $E_1(z) = h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3}$

- Then $H(z) = E_0(z^2) + z^{-1}E_1(z^2)$



- But this structure is not canonical because it requires six unit delays in each phase (12 overall).

- To make a cancnical structure, it is necessary to share unit delays between the phases:



Order = N = 7

Number of unit delays = 7 ✓ canonical

Coefficients = multiplier gains ✓ Direct

<u>EX</u>: three phases; $N=7$; length $=8$

$$H(z) = \left(h[0] + h[3]z^{-3} + h[6]z^{-6}\right)$$

$$+ z^{-1}\left(h[1] + h[4]z^{-3} + h[7]z^{-6}\right)$$

$$+ z^{-2}\left(h[2] + h[5]z^{-3}\right)$$

- Let $E_0(z) = h[0] + h[3]z^{-1} + h[6]z^{-2}$

  $E_1(z) = h[1] + h[4]z^{-1} + h[7]z^{-2}$

  $E_2(z) = h[2] + h[5]z^{-1}$

- Then $H(z) = E_0(z^3) + z^{-1}E_1(z^3) + z^{-2}E_2(z^3)$

Linear Phase FIR Structures

Type I, Type II: $h[n] = h[N-n]$

Type III, Type IV: $h[n] = -h[N-n]$

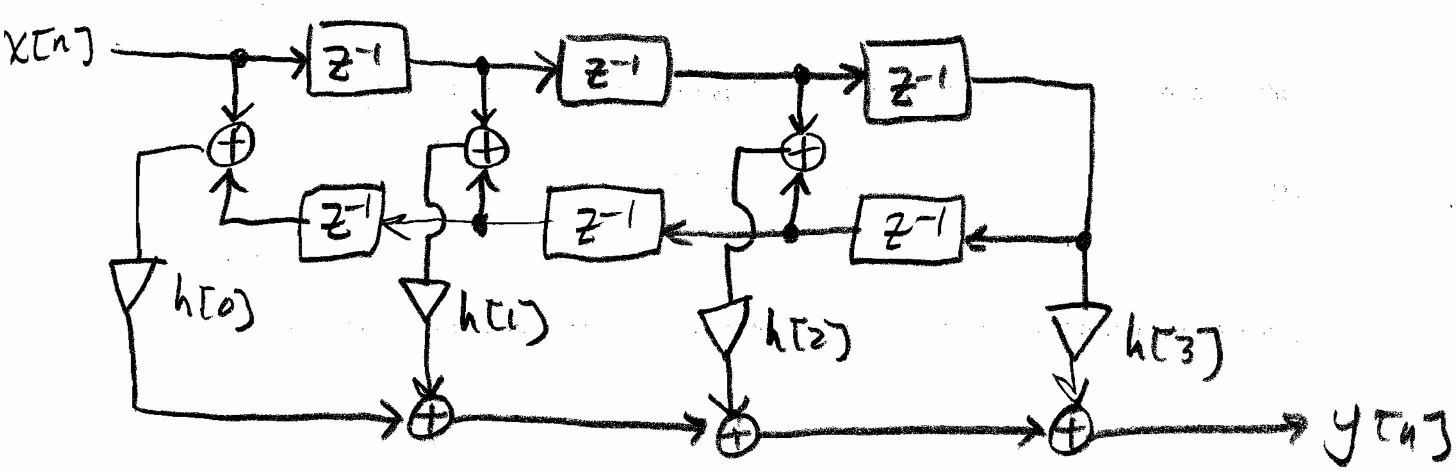
- This implies that multipliers can be shared.

EX: Type I; N = order = 6; length = 7

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6}$$

$$= h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[2]z^{-4} + h[1]z^{-5} + h[0]z^{-6}$$

$$= h[0](1+z^{-6}) + h[1](z^{-1}+z^{-5}) + h[2](z^{-2}+z^{-4}) + h[3]z^{-3}$$



IIR Direct Form I

- $H(z)$ is a ratio of polynomials in z^{-1}:
  - the numerator comes from the "$x[n]$" terms of the difference equation and gives the zeros of $H(z)$.
  - the denominator comes from the "$y[n]$" terms of the difference equation and gives the poles of $H(z)$.
- The main idea of Direct Form I is to implement the poles and zeros separately in a pair of cascaded systems.

EX:

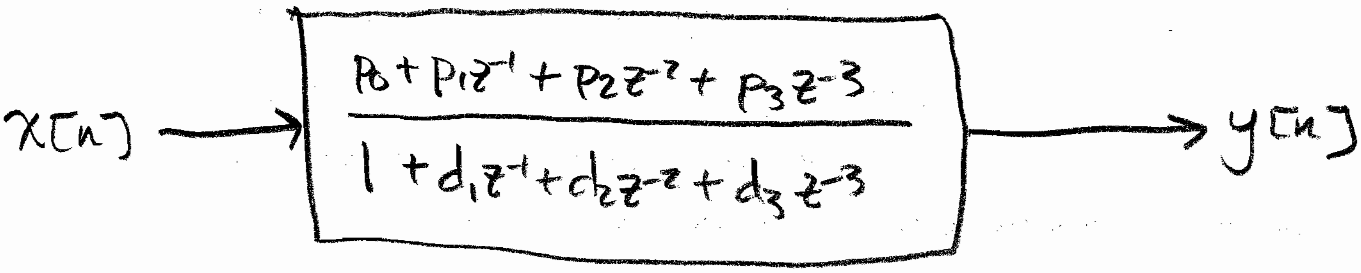
$$y[n] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3] - d_1 y[n-1] - d_2 y[n-2] - d_3 y[n-3]$$

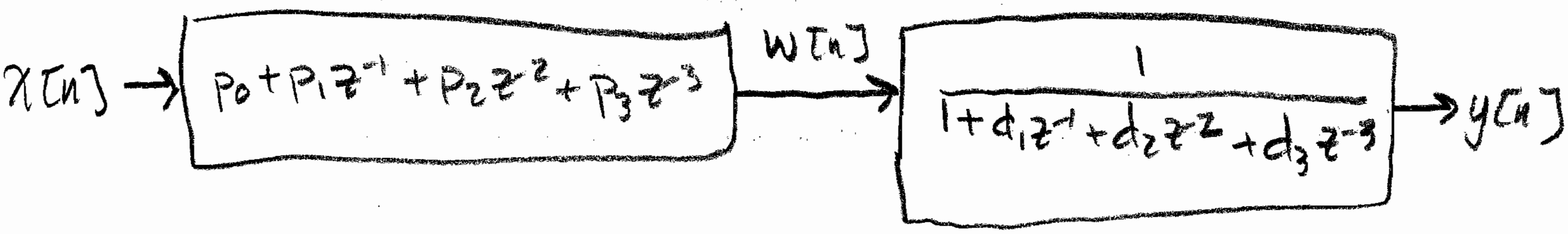
$$y[n] + d_1 y[n-1] + d_2 y[n-2] + d_3 y[n-3] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3]$$

$$Y(z)\left[1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}\right] = X(z)\left[p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}\right]$$

$\longrightarrow$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$ 8-20





$$w[n] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3] \quad (*)$$

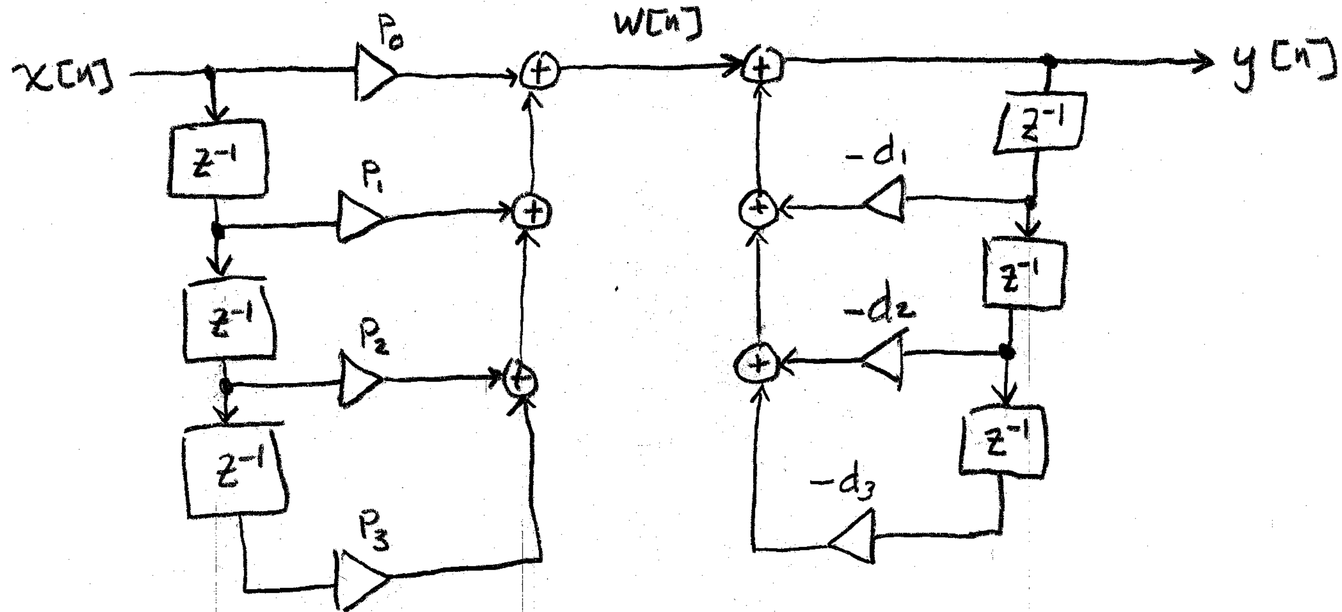
$$\frac{Y(z)}{W(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

$$W(z) = Y(z) + d_1 z^{-1} Y(z) + d_2 z^{-2} Y(z) + d_3 z^{-3} Y(z)$$

$$w[n] = y[n] + d_1 y[n-1] + d_2 y[n-2] + d_3 y[n-3]$$

$$y[n] = w[n] - d_1 y[n-1] - d_2 y[n-2] - d_3 y[n-3] \quad (**)$$

Direct form I comes from (*) and (**): 8-21



- Note that this form is direct but not canonical.

- By transposing this structure and sharing delays, one obtains Direct form II.

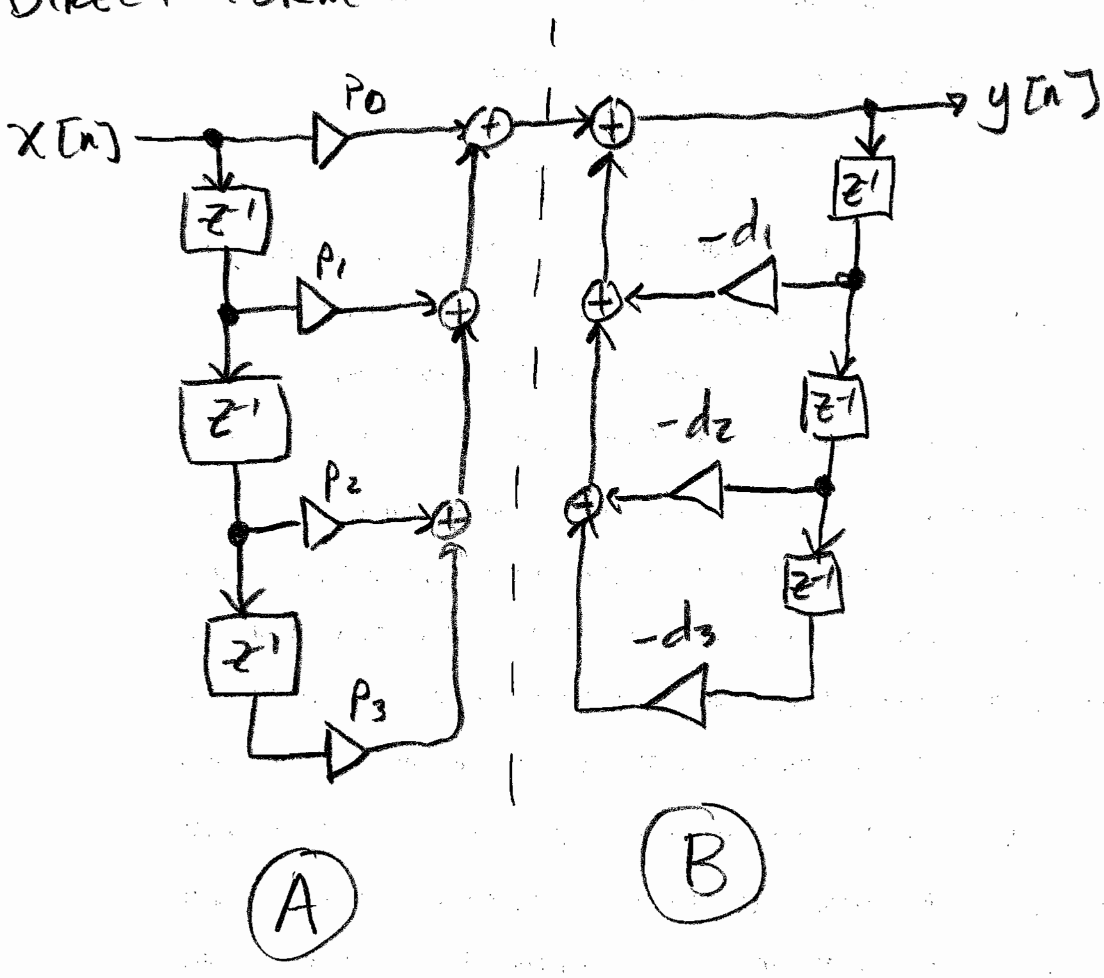
- Direct form II is both direct and canonical.

Direct Form II:

- We'll use a third-order system as an example to show how direct form II is obtained from the (more intuitive) direct form I.

$$H(z) = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

DIRECT FORM I:

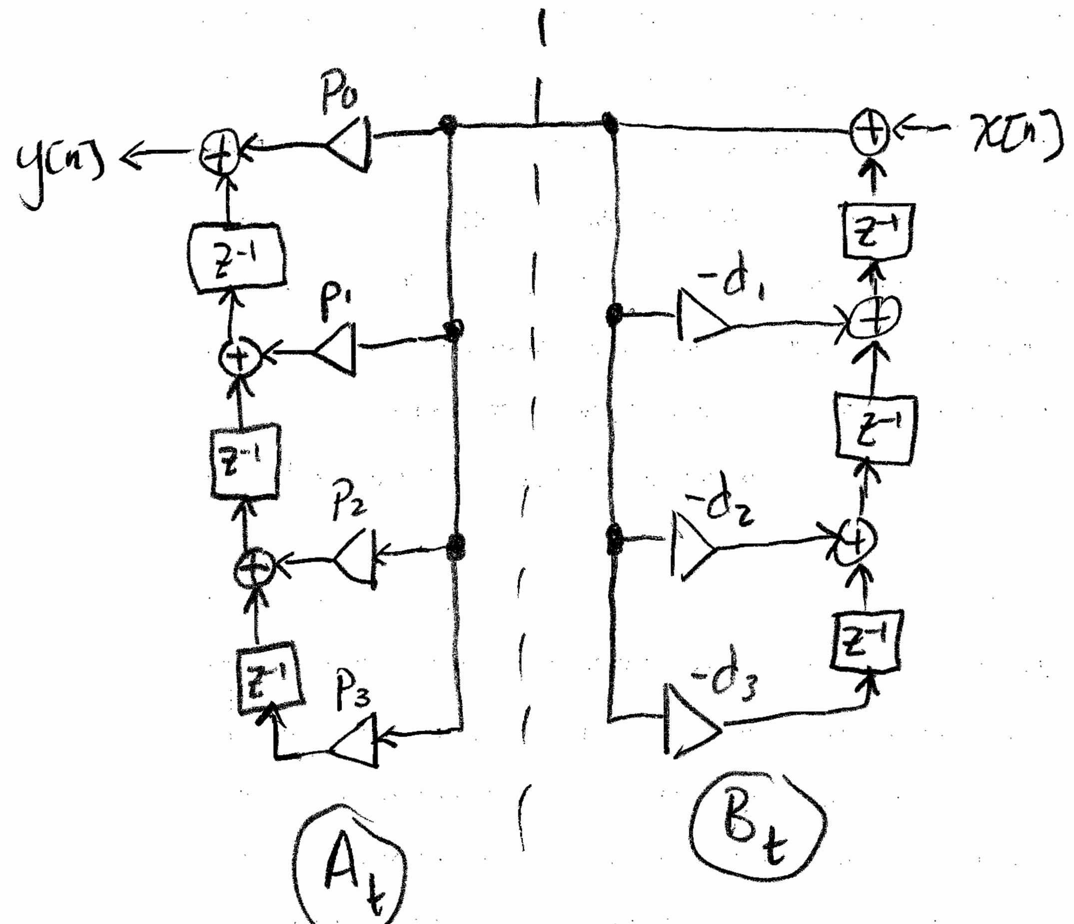


- Notice that this can be considered as a series connection of two systems A and B

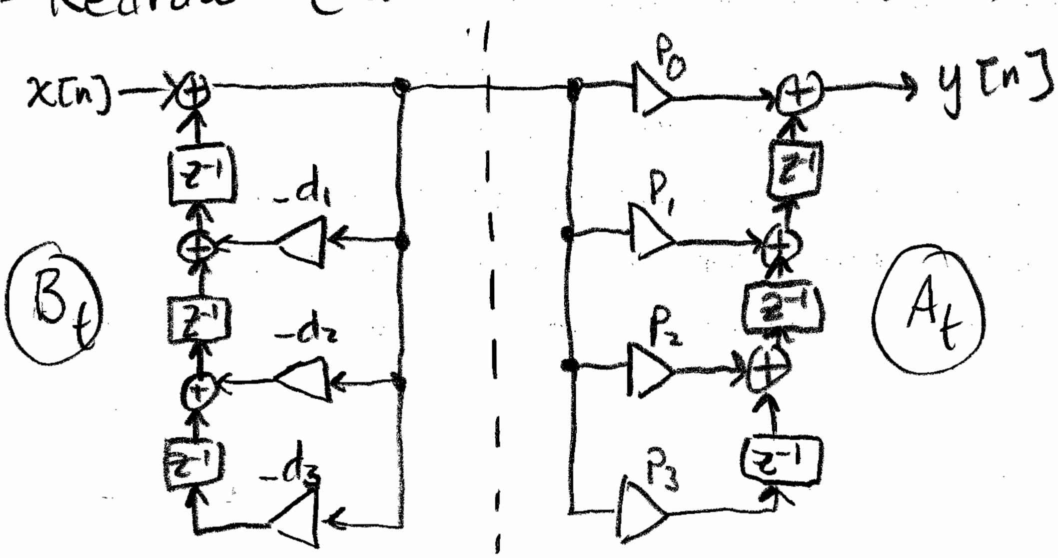
- Transpose the Direct Form I realization:

8-23

1. Reverse all paths
2. Change pickoffs to adders and adders to pickoffs
3. Interchange the input and output

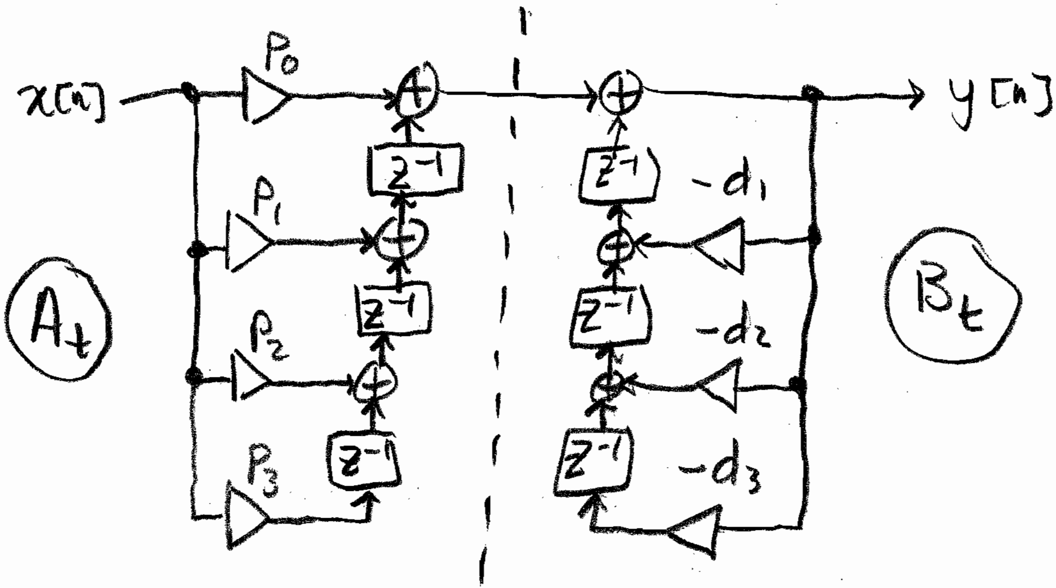


- Redraw (mirror left & right):



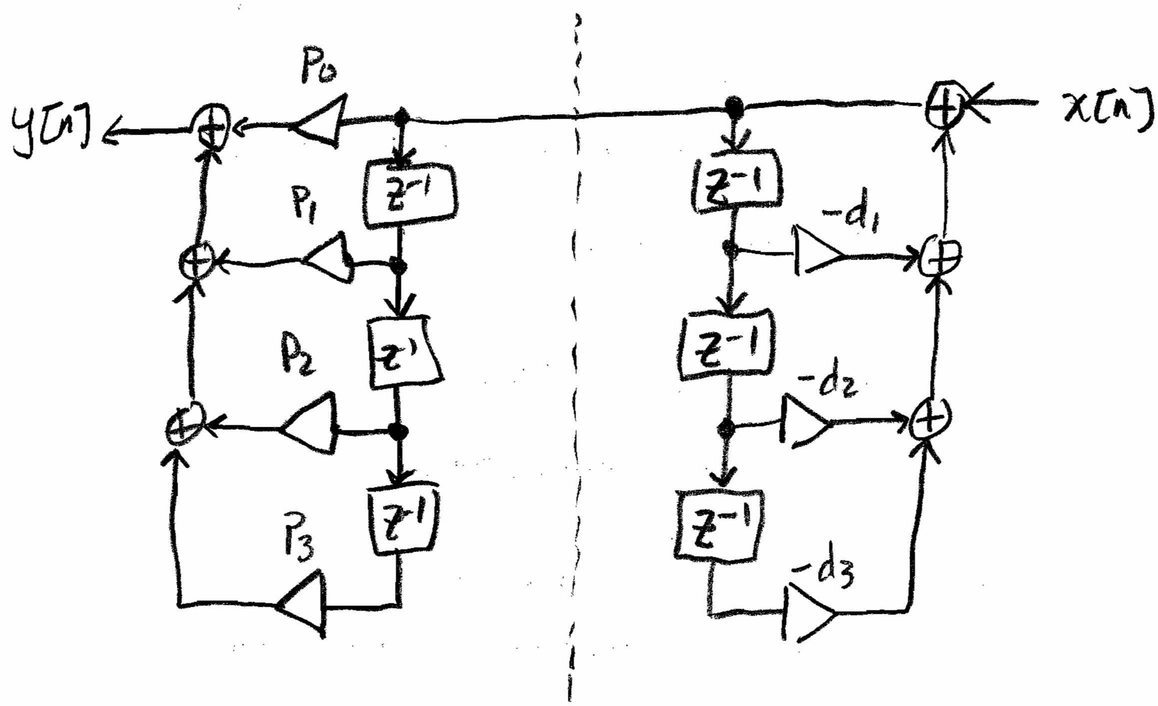
- Now switch the order of the systems (8-24)

B_t and A_t in this series connection:

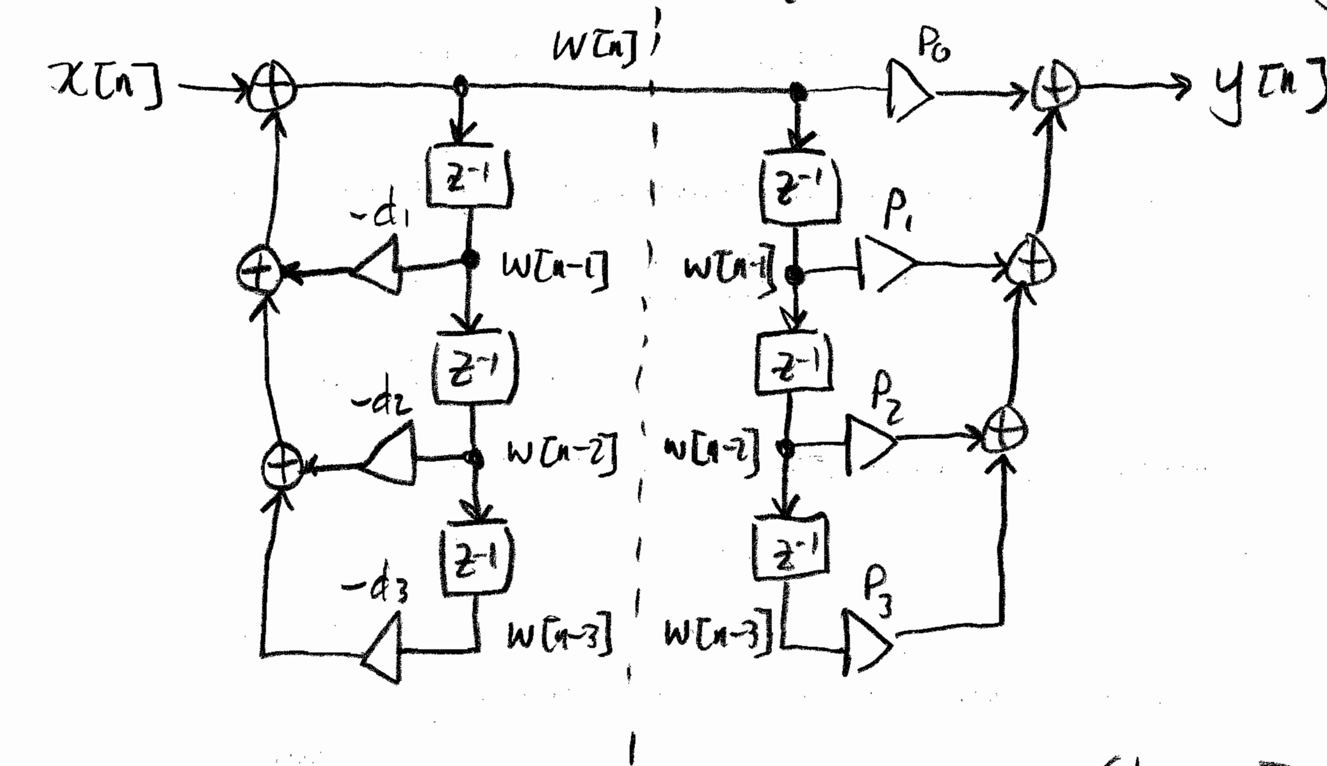


- Transpose Again

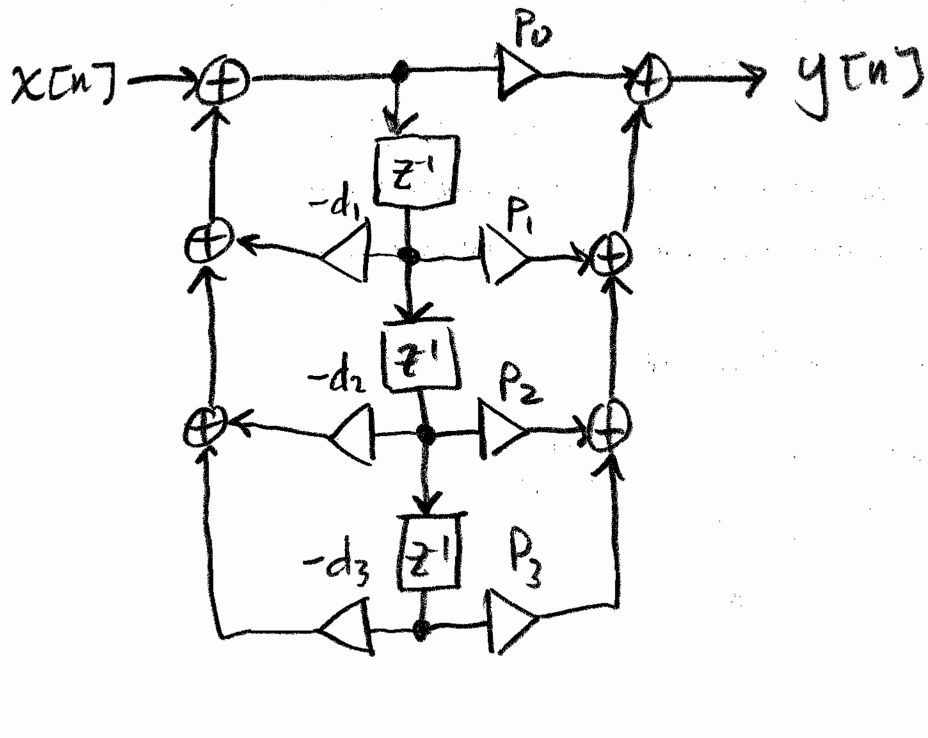
1. Reverse all paths
2. Change pickoffs to adders; adders to pickoffs
3. Interchange input & output



- Redraw (flip left & right)



- Notice that the signals entering the $\left\{\begin{matrix}\text{top}\\ \text{middle}\\ \text{bottom}\end{matrix}\right\}$ pair of delays are identical.
- These delays can therefore be combined. This gives the Direct Form II realization:



- Canonical

- Direct

Cascade Realizations:

- Often, a higher-order $H(z)$ is implemented as a series connection of 1st/ and/or 2nd/ order sections.
  - Factor the numerator and denominator into products of first-order zero terms (numerator) and first-order pole terms (denominator).
  - For a real-valued $h[n]$, complex-valued poles and zeros occur in complex conjugate pairs.
  - Group the poles and zeros together into first- and second-order sections.
  - Typically, conjugate pole pairs and/or zero pairs are grouped together in second-order sections.
- The first & second-order sections can be realized using Direct Form I, Direct Form II, or transposes of these.

Parallel Realizations:

- Alternatively, partial fractions can be used to express a higher-order $H(z)$ as a <u>sum</u> of first- and second-order sections instead of a product.

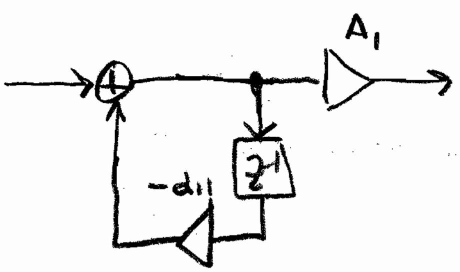
- This gives parallel realizations of the overall $H(z)$.

<u>EX</u>:

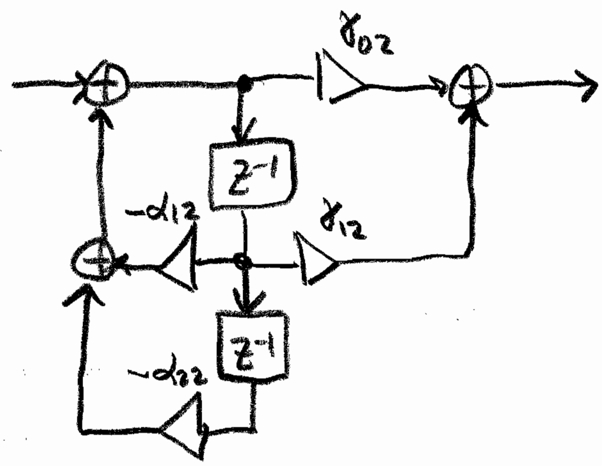
$$H(z) = \underbrace{\frac{A_1}{1+\alpha_{11}z^{-1}}}_{\text{1st order section resulting from a real pole.}} + \underbrace{\frac{\gamma_{02}+\gamma_{12}z^{-1}}{1+\alpha_{12}z^{-1}+\alpha_{22}z^{-2}} + \frac{\gamma_{03}+\gamma_{13}z^{-1}}{1+\alpha_{13}z^{-1}+\alpha_{23}z^{-2}}}_{\text{2nd order sections resulting from two pairs of complex conjugate poles.}}$$

e.g.

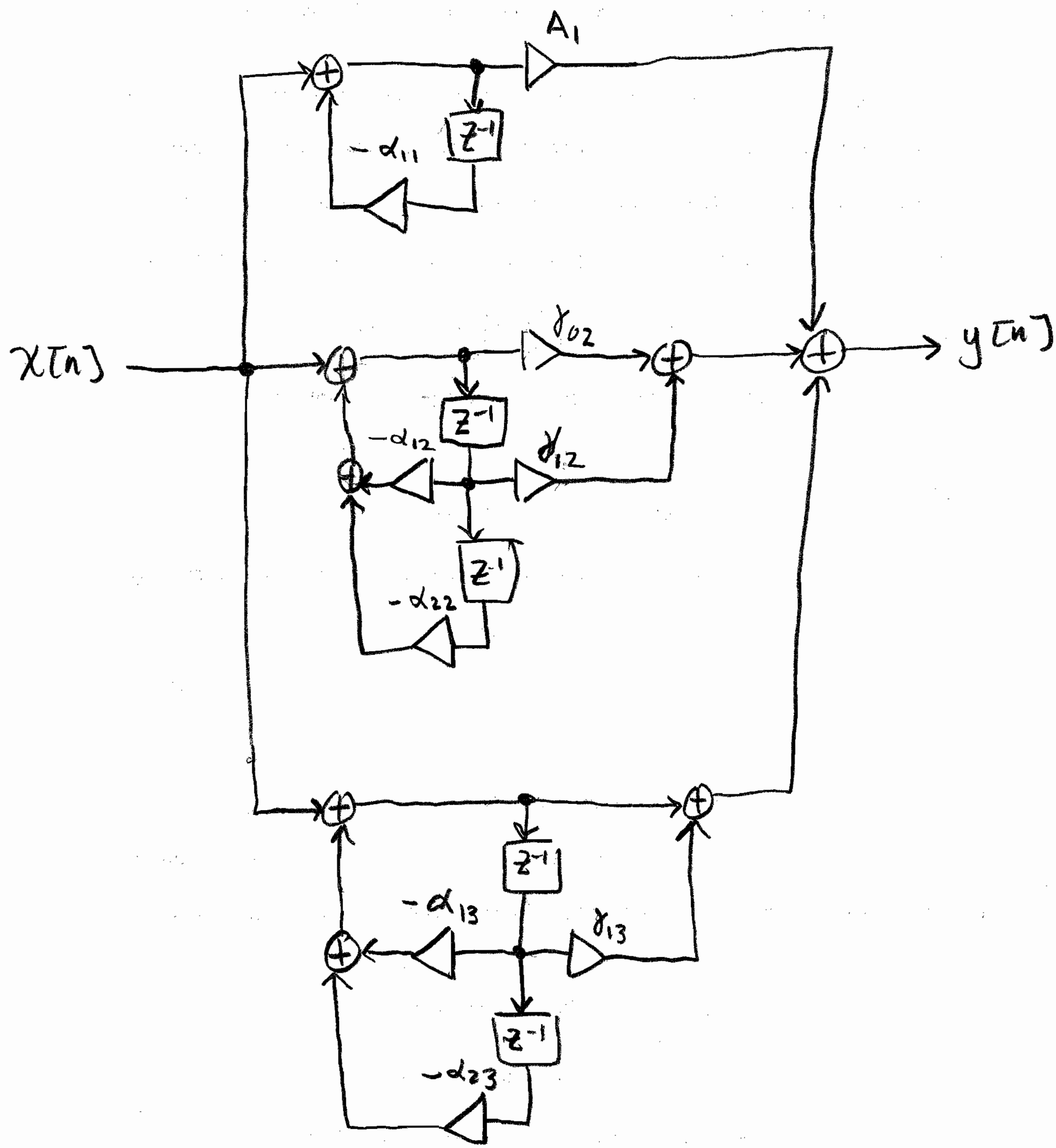
Direct Form II:



Direct Form II



Overall :



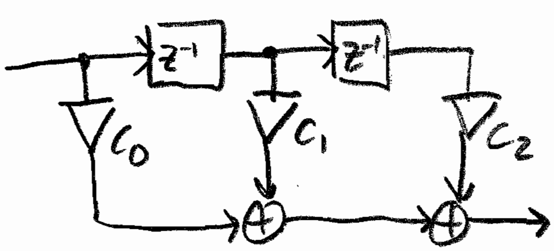
A_1
$-\alpha_{11}$
z^{-1}
$x[n]$
$y[n]$
γ_{02}
$-\alpha_{12}$
γ_{12}
z^{-1}
$-\alpha_{22}$
z^{-1}
$-\alpha_{13}$
γ_{13}
z^{-1}
$-\alpha_{23}$
z^{-1}

- If the numerator of $H(z)$ has an order greater than the denominator, then there will also be "direct transmission terms" in the partial fraction expansion.

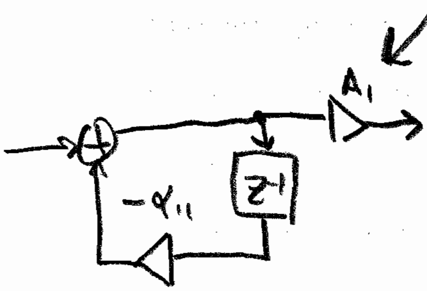
- This will lead to an FIR structure in parallel with the IIR structures in the overall parallel realization.

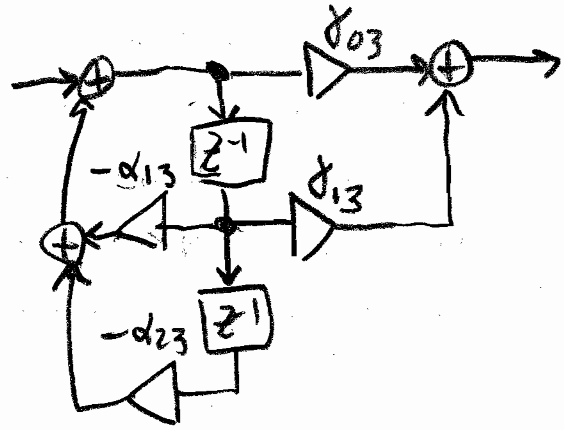
EX:

$$H(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + \frac{A_1}{1+\alpha_{11}z^{-1}} + \frac{\gamma_{02}+\gamma_{12}z^{-1}}{1+\alpha_{12}z^{-1}+\alpha_{22}z^{-2}} + \frac{\gamma_{03}+\gamma_{13}z^{-1}}{1+\alpha_{13}z^{-1}+\alpha_{23}z^{-2}}$$

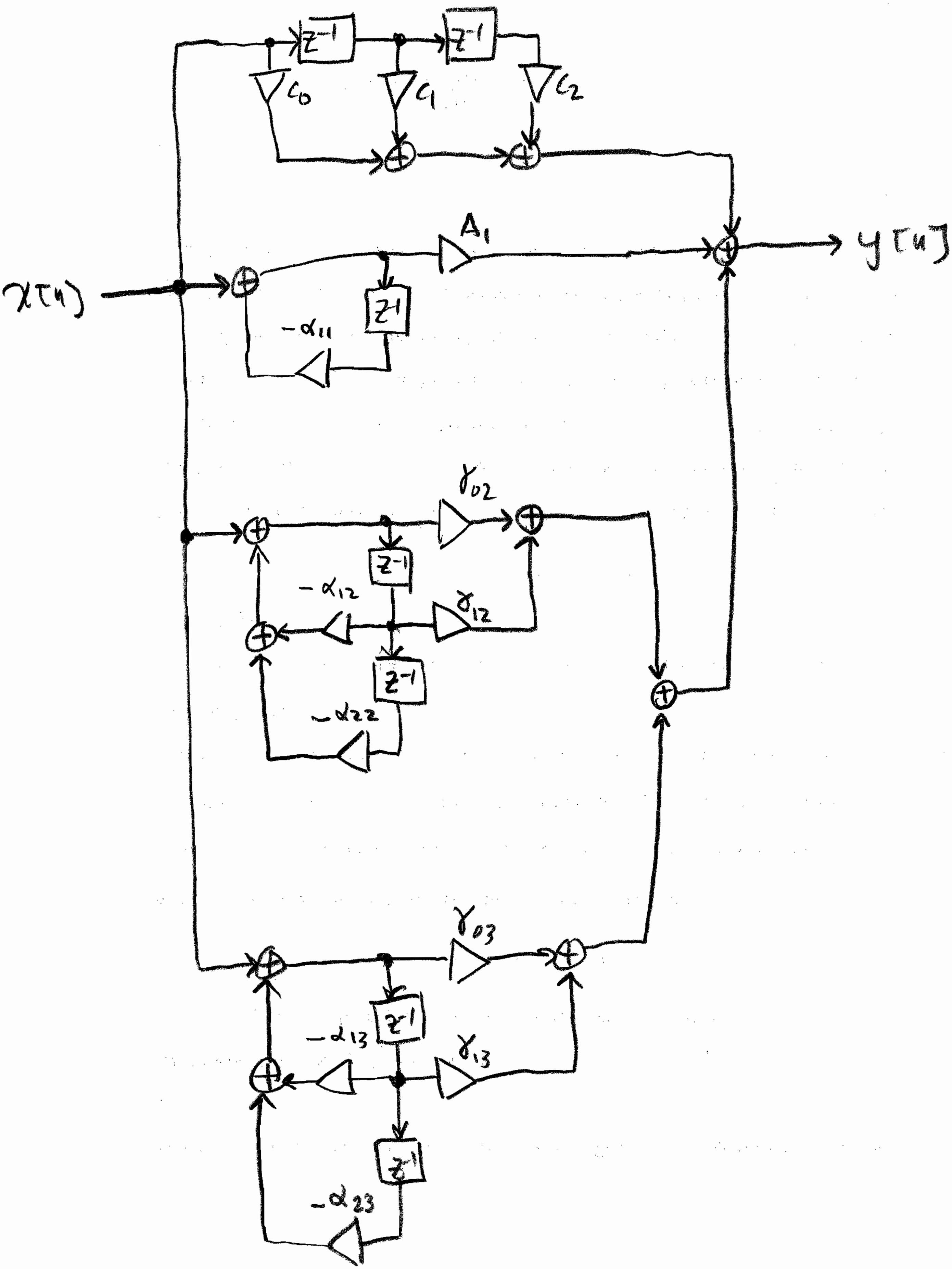


e.g.





Overall:



z^{-1}
z^{-1}
c_0
c_1
c_2
A_1
$y[n]$
$x[n]$
z^{-1}
$-\alpha_{11}$
γ_{02}
z^{-1}
$-\alpha_{12}$
γ_{12}
z^{-1}
$-\alpha_{22}$
γ_{03}
z^{-1}
$-\alpha_{13}$
γ_{13}
z^{-1}
$-\alpha_{23}$

Allpass Sections

- A first-order allpass section has the form

$$A(z) = \frac{d_1^* + z^{-1}}{1 + d_1 z^{-1}}$$

one pole at $z = -d_1$

→ For a real $h[n]$, if d_1 is complex, it will occur with a conjugate section having a pole at $z = -d_1^*$.

→ The two conjugate poles can be implemented together as a single 2nd order allpass section having all real coefficients.

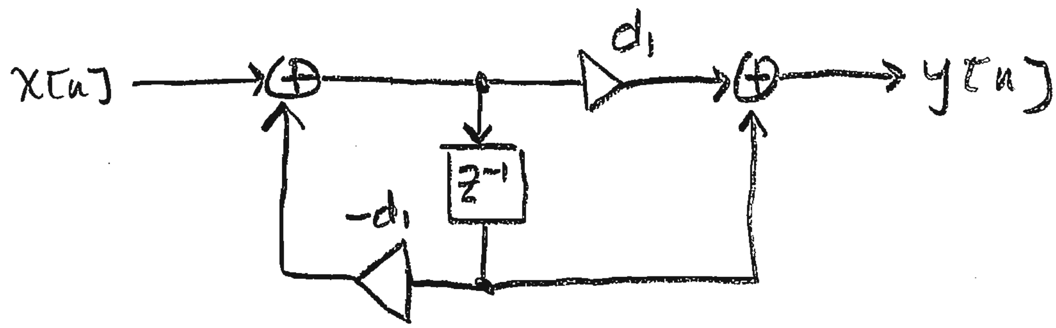
→ So let's restrict our attention to the case where d_1 is real.

$$x[n] \longrightarrow \boxed{A(z)} \longrightarrow y[n]$$

$$A(z) = \frac{Y(z)}{X(z)} = \frac{d_1 + z^{-1}}{1 + d_1 z^{-1}}$$

$$y[n] + d_1 y[n-1] = d_1 x[n] + x[n-1] \qquad (*)$$

Direct Form II :



- There are several ways that this structure can be manipulated to yield forms that involve "two" multipliers:

  - one with a gain of d_1,
  - one with a gain of -1.

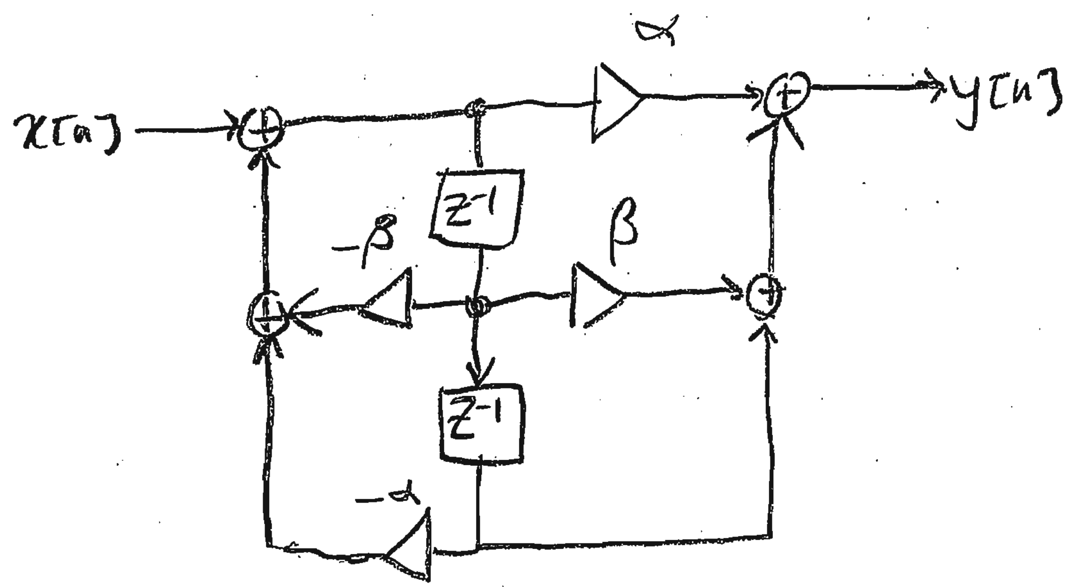
→ But a multiplier with a gain of -1 can be implemented "for free" with an adder/subtractor, which is just an adder with a two's complement circuit on one of the inputs.

→ So really these forms require just one true (nontrivial) multiplier.

- A 2nd-order allpass section with real coefficients has the form

$$A(z) = \frac{\alpha + \beta z^{-1} + z^{-2}}{1 + \beta z^{-1} + \alpha z^{-2}}$$

Direct Form II:



- Again, there are several techniques that can be used for manipulating this structure to obtain forms using only two nontrivial multipliers

Lattice Structures

- An LTI digital filter has a transfer function $H(z)$ that is a ratio of two polynomials in z^{-1}.
- The roots of the numerator polynomial are the zeros of the filter.
- The roots of the denominator are the poles
  - Actually, the poles/zeros are the values of z that make the denominator/numerator equal to zero (not the values of z^{-1}).

- For an FIR filter, the denominator is equal to 1... meaning that there are nontrivial zeros, but all the poles are at $z=0$.

NOTE: an IIR filter generally has both a numerator polynomial and a denominator polynomial, giving nontrivial zeros and nontrivial poles.

- An IIR filter where the numerator is equal to 1 is called an "all-pole IIR filter"... it does not have any nontrivial zeros.

- We usually assume that the filter is causal and that the numerator and denominator polynomials of $H(z)$ have real coefficients

- This means that, for a stable design:

  1. The poles must be strictly inside the unit circle.
  2. The poles and zeros are real or occur in complex conjugate pairs.

Recall from page 8-8: a <u>direct</u> form (or structure) is one where the multiplier gains in the diagram are equal to the coefficients in the numerator and denominator polynomials of $H(z)$.

- So far, we have seen several direct structures for realizing FIR and IIR filters:
    - FIR direct forms (p. 8-12)
    - FIR polyphase realization (p. 8-16)
    - Linear phase FIR structures (p. 8-18)
    - IIR direct form I (p. 8-21)
    - IIR direct form II (p. 8-25)
  - As well as cascade and parallel realizations.

- The basic direct forms are <u>often</u> implemented as a cascade connection of second-order sections.

- This approach can have <u>problems</u> if the filter is implemented in finite precision hardware:
    - The numerator and denominator coefficients of $H(z)$ are rounded or truncated for implementation with fixed precision hardware.
    - Thus, there is an <u>error</u> or disageement between the theoretical filter and the filter that is actually built.
    - We often think of this difference as "quantization noise" or "quantization error."

Problems:

- The quantization errors can be significant, causing the practical filter to <u>not</u> behave the way it is supposed to... i.e., to <u>not</u> behave like the theoretical filter.
- For an IIR filter, quantization errors in the denominator of $H(z)$ can cause one or more poles to move <u>outside</u> the unit circle, making the filter <u>unstable</u>.
- When a high order filter is implemented as a cascade of second order sections, quantization errors introduced by the first sections in the chain get multiplied by the gains of all the following sections... they can be amplified and become very large!

When implementing a filter as a cascade of second order sections with fixed precision hardware,

- you should try to minimize these problems by:

1. Make the second order sections by grouping the poles together with the zeros that are closest to them in the z-plane.
2. Place the sections with poles that are close to the unit circle <u>last</u> in the processing chain to minimize the chance that noise introduced by following sections might move them outside the unit circle.

Nevertheless, because of these problems, many designers use this "rule of thumb": do <u>not</u> use a direct structure for a filter with order > 10.

<u>Lattice structures</u> are an alternative way to realize FIR and IIR filters that provide:

- low sensitivity to coefficient quantization effects.
- modularity
- a simple criterion for ensuring stability (the Schur-Cohn test).

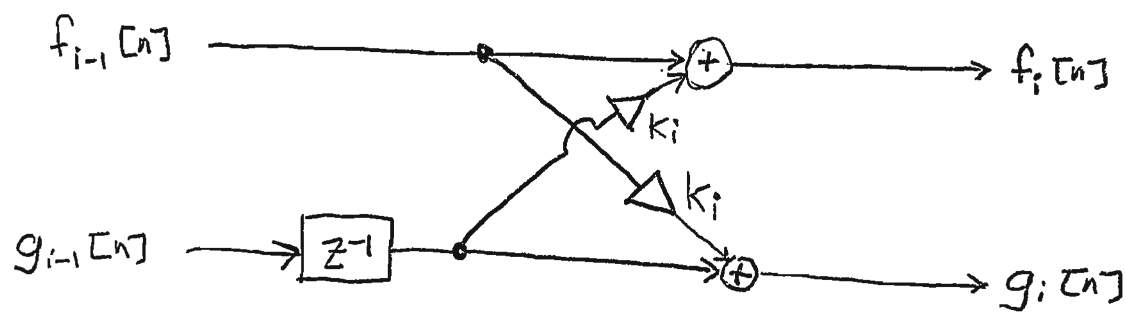
- Here, our treatment will be somewhat more restricted than what is in the textbook.
  - This will make everything substantially <u>simpler</u>.
  - But our notation will be slightly different from the book.

- Lattice structures are widely used in
  - implementation of high order transfer functions
  - speech processing
  - adaptive signal processing
  - modeling of statistical signals

FIR Lattice Structures:

- Lattice structures are made by cascading first order sections that all have the same basic diagram.
- Each section is a <u>two-port network</u>. It has two inputs and two outputs.
    - Some authors use the term "digital two-pair" instead of "two-port network."

Diagram for the i^{th} section:



- inputs: $f_{i-1}[n]$, $g_{i-1}[n]$
- outputs: $f_i[n]$, $g_i[n]$
    - Both multipliers have the same gain: k_i, called the "reflection coefficient."

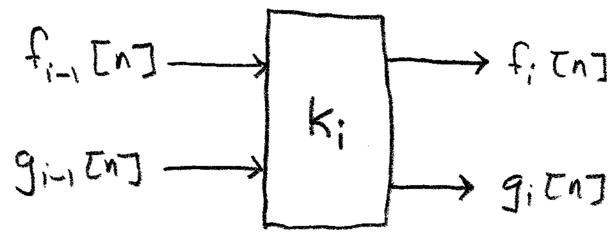
- difference equation:

$$\begin{cases} f_i[n] = f_{i-1}[n] + k_i g_{i-1}[n-1] \\ g_i[n] = g_{i-1}[n-1] + k_i f_{i-1}[n] \end{cases} \qquad (*)$$

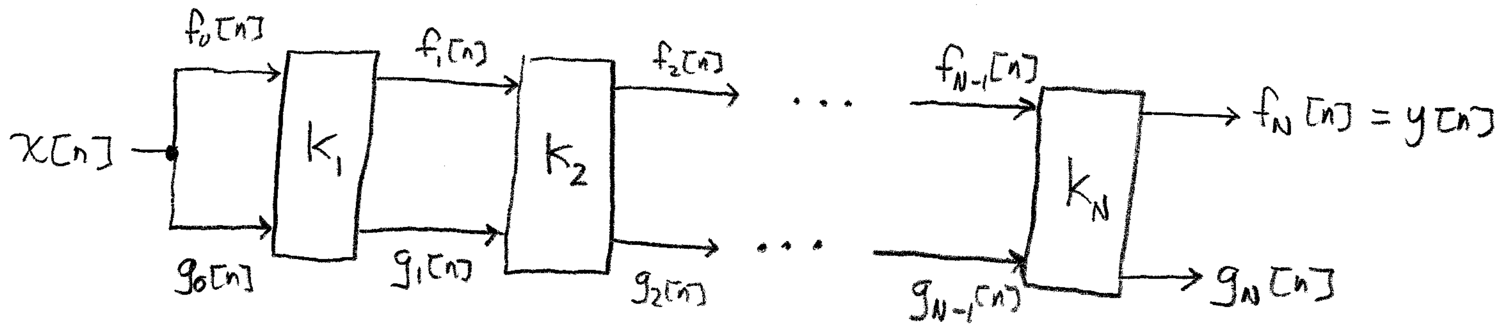
Z-transform:

$$\begin{cases} F_i(z) = F_{i-1}(z) + k_i z^{-1} G_{i-1}(z) \\ G_i(z) = z^{-1} G_{i-1}(z) + k_i F_{i-1}(z) \end{cases} \qquad (**)$$

- Simplified diagram for the i^{th} section:



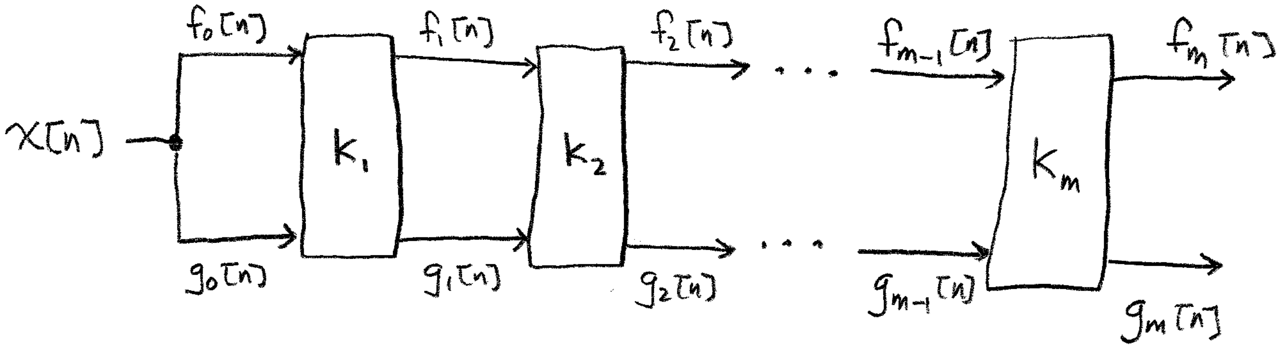
- A lattice structure to realize an N^{th} order FIR filter requires N sections. It looks like this:



input: $x[n]$

output: $y[n] = f_N[n]$

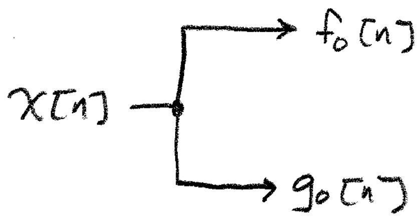
To analyze this structure, consider a smaller filter that includes only the first m sections, where $0 \le m \le N$:



- Call the transfer function from $x[n]$ to $f_m[n]$ "$A_m(z)$", so that $F_m(z) = A_m(z)\,X(z)$.

  - if $m=N$, then this is the whole filter and $A_m(z) = A_N(z) = H(z)$, the transfer function from $x[n]$ to $y[n]$.

  - if $m=0$, then the picture is



So $f_0[n] = x[n]$ and $\underline{\underline{A_0(z) = 1}}$.

- With some work, it can be shown by induction that, more generally,

$$A_m(z) = A_{m-1}(z) + k_m z^{-m} A_{m-1}(z^{-1}) .$$

- This is called the "step-up recursion."

☆ Given the reflection coefficients k_i, we can now use the step-up recursion and the fact that $A_0(z)=1$ to find $A_m(z)$ for any $1 \leq m \leq N$.

→ When $m=N$, we have $A_N(z) = H(z)$.

→ So this gives us a method to find $H(z)$ from the reflection coefficients.

<u>EX</u>: Given a second order FIR lattice filter with reflection coefficients $k_1 = \frac{1}{2}$ and $k_2 = \frac{1}{4}$, find the transfer function $H(z)$.

$$A_0(z) = 1$$

$$A_1(z) = A_0(z) + k_1 z^{-1} A_0(z^{-1})$$
$$= 1 + \tfrac{1}{2} z^{-1}$$

$$H(z) = A_2(z) = A_1(z) + k_2 z^{-2} A_1(z^{-1})$$
$$= 1 + \tfrac{1}{2} z^{-1} + \tfrac{1}{4} z^{-2}(1 + \tfrac{1}{2} z)$$
$$= 1 + \tfrac{1}{2} z^{-1} + \tfrac{1}{4} z^{-2} + \tfrac{1}{8} z^{-1}$$
$$= 1 + \tfrac{5}{8} z^{-1} + \tfrac{1}{4} z^{-2}$$

//

- There is also a "step-down recursion":

$$A_{m-1}(z) = \frac{1}{1-k_m^2}\left[A_m(z) - k_m z^{-m} A_m(z^{-1})\right]$$

FACTS: ① $H(z) = A_N(z)$

② Let $A_m(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_m z^{-m}$.

Then $k_m = a_m$.

☆ We can use these two facts with the step-down recursion to find the reflection coefficients from $H(z)$.

<u>EX</u>: Find the reflection coefficients for a lattice realization of an FIR filter with $H(z) = 1 - \frac{1}{2}z^{-2}$.

- Since $H(z)$ is second order, we have $N=2$.

- $A_2(z) = H(z) = 1 - \frac{1}{2}z^{-2}$

$a_2 = k_2 \Rightarrow k_2 = -\frac{1}{2}$ //

- $$A_1(z) = \frac{1}{1-k_2^2}\left[A_2(z) - k_2 z^{-2} A_2(z^{-1})\right]$$

$$= \frac{1}{1-\frac{1}{4}}\left[1 - \frac{1}{2}z^{-2} + \frac{1}{2}z^{-2}\left(1 - \frac{1}{2}z^{2}\right)\right]$$

$$= \frac{1}{3/4}\left[1 - \frac{1}{2}z^{-2} + \frac{1}{2}z^{-2} - \frac{1}{4}\right]$$

$$= \frac{4}{3}\left[\frac{3}{4}\right] = 1 = 1 + 0 \cdot z^{-1}$$

$a_1 = k_1 \Rightarrow k_1 = 0$ //

So, for the FIR lattice filter realization shown on page 8-39,

- we can find the reflection coefficients given $H(z)$ [p. 8-42].

- we can find $H(z)$ given the reflection coefficients [p. 8-41].

FACT: for any m with $0 \leq m \leq N$, the transfer function from $f_m[n]$ to $g_m[n]$ is given by

$$\frac{G_m(z)}{F_m(z)} = \frac{z^{-m} A_m(z^{-1})}{A_m(z)},$$

which is <u>all pass</u>, since the numerator and denominator are mirror image polynomials.

Schur-Cohn Stability Test

- generalizes the idea of the "stability triangle" to filters with order $N > 2$.

FACT: the polynomial

$$A_m(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_m z^{-m}$$

has all roots <u>inside</u> the unit circle if the reflection coefficients $k_1, k_2, \ldots, k_m$ all have magnitude strictly less than unity, i.e., if $|k_i| < 1$ for $i = 1, 2, 3, \ldots, m$.

- For a general IIR filter with transfer function

$$H(z) = \frac{B(z)}{A(z)},$$

this gives us a way to check for stability without having to factor the denominator.

- Implement the polynomial $A(z)$ in an FIR lattice structure.
- If the reflection coefficients $k_1, k_2, \ldots k_N$ are all strictly less than 1 in magnitude,

$\Rightarrow$ Then the filter is stable (provided it is causal).

All-Pole IIR Lattice Structure

- If an IIR filter has nontrivial poles, but all the zeros are located at $z=0$,
  - $\rightarrow$ then the numerator polynomial is equal to 1 and

$$H(z) = \frac{1}{A_N(z)}$$

where, as before, $A_N(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}$.

- This is an N^{th} order all-pole IIR filter.
- For IIR lattice structures, we will re-write the difference equations (*) on page 8-38 like this:

$$\begin{cases} f_{i-1}[n] = f_i[n] - k_i\, g_{i-1}[n-1] \\ g_i[n] = g_{i-1}[n-1] + k_i f_{i-1}[n] \end{cases}$$

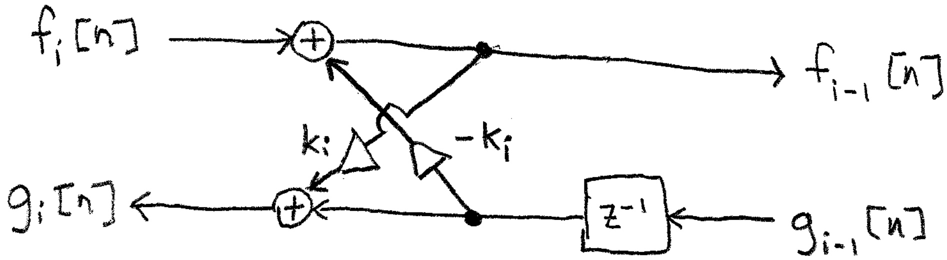
  - The only thing that's different from before is that the first equation is solved for $f_{i-1}[n]$ instead of for $f_i[n]$.

- This does <u>not</u> change any of the relationships between the signals.

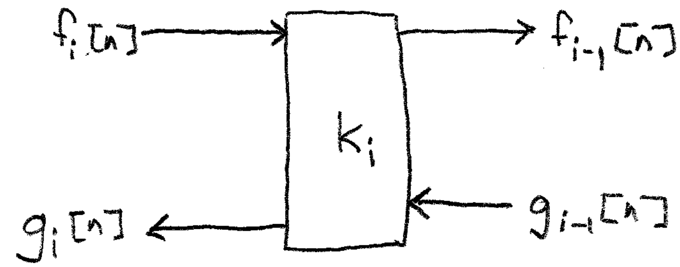
- But it enables us to draw the i^{th} section in a <u>different</u> way... where the direction of flow on the bottom path is <u>reversed</u>.

(This is shown on the next page)

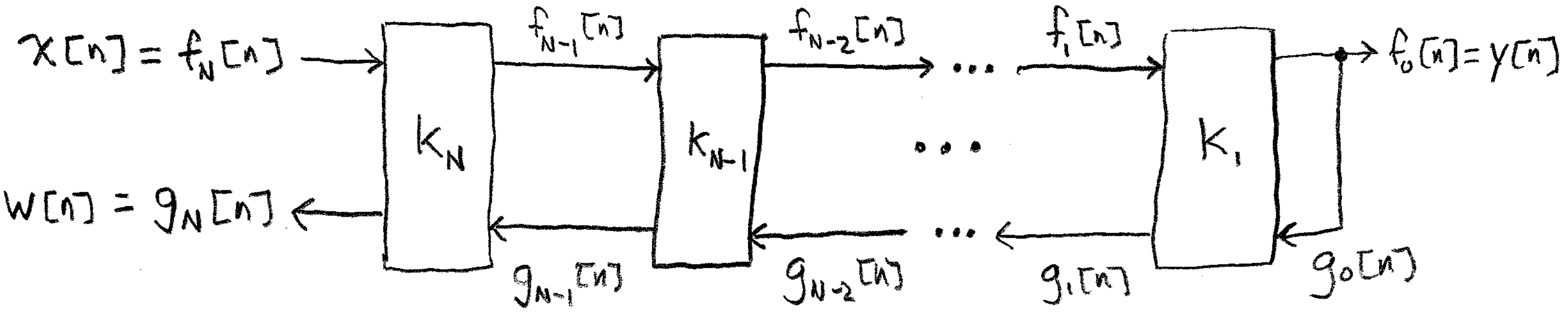
- New diagram for the i^{th} section:



- New simplified diagram for the i^{th} section:



Now, here is the lattice structure for an N^{th} order all-pole IIR filter:



input: $x[n]$
output: $y[n]$

→ $w[n]$ is an intermediate signal that we will use later

- Compare this to the FIR case shown on page 8-39:

IIR Case	FIR Case
$f_0[n] = y[n]$	$f_0[n] = x[n]$
$g_0[n] = y[n]$	$g_0[n] = x[n]$
$f_N[n] = x[n]$	$f_N[n] = y[n]$

- In the all-pole IIR structure, the relationship between $f_i[n] / g_i[n]$ and $x[n] / y[n]$ is <u>switched</u>.
  - → keeping the step-up recursion and step-down recursion the <u>same</u> as before (in the FIR case) will, for the new IIR case:
    - → flip the transfer function upside down.
    - → The relationship that we had between the reflection coefficients k_i and the <u>zeros</u> of $H(z)$ in the <u>FIR</u> case
      - will become the relationship between the reflection coefficients k_i and the <u>poles</u> of $H(z)$ in the new <u>IIR</u> case.

- So, for the IIR case, call the transfer function from $y[n]$ to $f_m[n]$ "$A_m(z)$", so that $F_m(z) = A_m(z)\,Y(z)$.

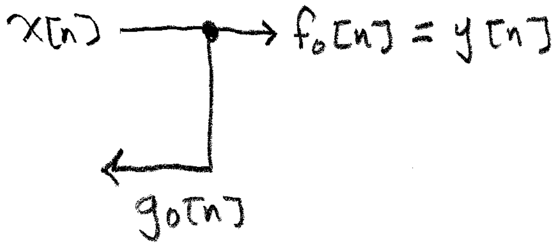
- When $m = N$, we obtain

$$F_N(z) = X(z) = A_N(z)\,Y(z),$$

so $$\frac{X(z)}{Y(z)} = A_N(z)$$

and $$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{A_N(z)}$$ as desired for an all-pole IIR filter.

- With $m = 0$, the picture is:



So $f_0[n] = y[n]$ and $A_0(z) = 1$, as before.

- The step-up recursion is <u>still</u> given by

$$A_m(z) = A_{m-1}(z) + k_m z^{-m} A_{m-1}(z^{-1}) \qquad \left(\text{as on p. 8-41}\right)$$

- The step-down recursion is <u>still</u> given by

$$A_{m-1}(z) = \frac{1}{1-k_m^2}\left[A_m(z) - k_m z^{-m} A_m(z^{-1})\right] \qquad \left(\text{as on p. 8-42}\right)$$

- As before on page 8-42, the reflection coefficient k_m for the m^{th} section is still given by the coefficient a_m of z^{-m} in $A_m(z)$:

$$A_m(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_m z^{-m}$$

$$k_m = a_m$$

- These two equations together with $A_0(z) = 1$, the step-up recursion, and the step down recursion give us a method to

① Find $H(z)$ from the reflection coefficients k_i

② Find the reflection coefficients k_i from $H(z)$

(N^{th} order all-pole IIR lattice filter)

EX: Given a 2^{nd} order all-pole IIR lattice filter with reflection coefficients $k_1 = \frac{1}{2}$ and $k_2 = \frac{1}{4}$, find the transfer function $H(z)$.

$$A_0(z) = 1$$

$$A_1(z) = A_0(z) + k_1 z^{-1} A_0(z^{-1}) = 1 + \tfrac{1}{2} z^{-1}$$

$$\begin{aligned} A_2(z) &= A_1(z) + k_2 z^{-2} A_1(z^{-1}) \\ &= (1 + \tfrac{1}{2} z^{-1}) + \tfrac{1}{4} z^{-2} (1 + \tfrac{1}{2} z) \\ &= 1 + \tfrac{1}{2} z^{-1} + \tfrac{1}{4} z^{-2} + \tfrac{1}{8} z^{-1} \\ &= 1 + \tfrac{5}{8} z^{-1} + \tfrac{1}{4} z^{-2} \end{aligned}$$

(compare to the example on p. 8-41)

$$H(z) = \frac{1}{A_2(z)} = \frac{1}{1 + \frac{5}{8} z^{-1} + \frac{1}{4} z^{-2}} \quad //$$

EX: Given a 2nd order all-pole IIR filter with transfer function $H(z) = \frac{1}{1-\frac{1}{2}z^{-2}}$, find the reflection coefficients for a lattice realization.

$N = 2$

$$H(z) = \frac{1}{1-\frac{1}{2}z^{-2}} = \frac{1}{A_2(z)} \Longrightarrow A_2(z) = 1 - \frac{1}{2}z^{-2}$$

$$k_2 = -\frac{1}{2} \; //$$

$$A_1(z) = \frac{1}{1-k_2^2}\left[A_2(z) - k_2 z^{-2} A_2(z^{-1})\right]$$

$$= \frac{1}{1-\frac{1}{4}}\left[\left(1-\frac{1}{2}z^{-2}\right) + \frac{1}{2}z^{-2}\left(1-\frac{1}{2}z^{2}\right)\right]$$

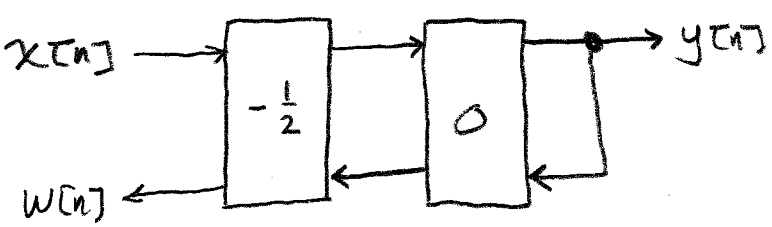
$$= \frac{1}{3/4}\left[1 - \frac{1}{2}z^{-2} + \frac{1}{2}z^{-2} - \frac{1}{4}\right]$$

$$= \frac{4}{3}\left[\frac{3}{4}\right] = 1 = 1 + 0z^{-1}$$

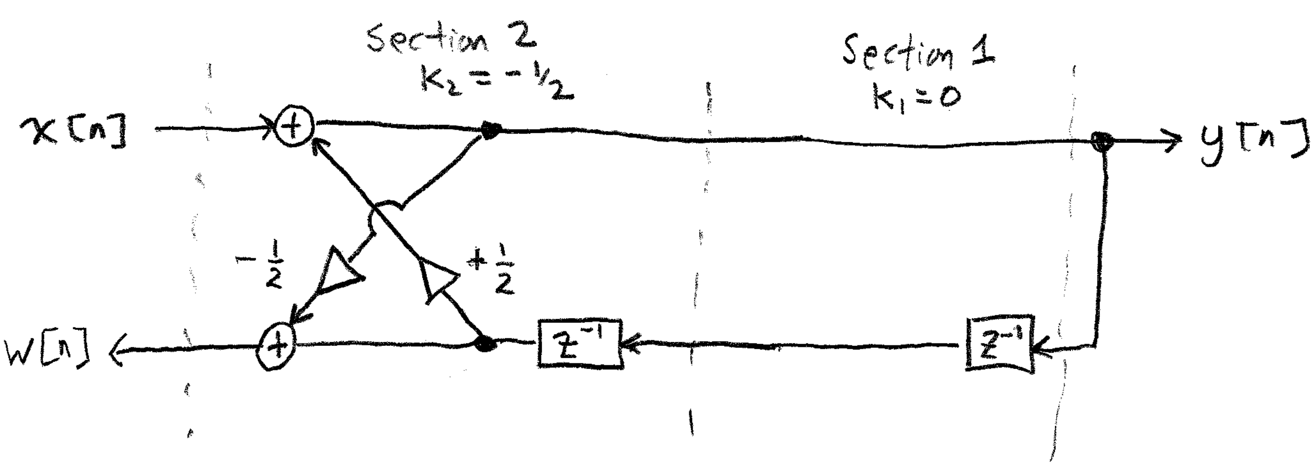
$$k_1 = 0 \; //$$

(Compare to the FIR example on p. 8-42)

- From the diagrams on p. 8-46, the lattice structure is:



- The full lattice filter block diagram is:



FACT: for the all-pole IIR lattice structure, the transfer function from $x[n]$ to $w[n]$ is allpass. It is given by

$$\frac{W(z)}{X(z)} = \frac{z^{-N} A_N(z^{-1})}{A_N(z)}.$$

General IIR Lattice Structure

- We now develop a lattice structure for a general IIR filter that has both nontrivial poles and nontrivial zeros.
- This means that both the numerator and denominator of $H(z)$ are nontrivial polynomials in z^{-1}.
- Strategy:
  1. implement the denominator of $H(z)$ as an all-pole IIR filter, just like we did on pages 8-45 through 8-50.
  2. Add an FIR structure to the lower signal path to implement the numerator of $H(z)$.
     → This will require some additional notation.
- In the book, this is called the "Gray-Markel method."
- The structure is also sometimes called a "lattice-ladder" structure.

<u>NOTE:</u> for the method to work, the order of the denominator of $H(z)$ must be $\geq$ the order of the numerator.

$\Rightarrow$ If that's <u>not</u> the case, then do long division first to factor out an FIR transfer function. Then apply the Gray-Markel method to the remainder to get an IIR lattice structure. Finally, implement the "factored out" FIR terms in a lattice and place it in parallel with the IIR structure.

<u>Additional Notation</u>

- up to now, we have made use of the polynomials

$$A_m(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_m z^{-m}$$

for $0 \leq m \leq N$.

- For our FIR lattice structure, $A_m(z)$ was the transfer function from $x[n]$ to $f_m[n]$, so that

$$F_m(z) = A_m(z) X(z) \qquad \text{(p. 8-40)}$$

- For our all-pole IIR lattice structure, $A_m(z)$ was the transfer function from $y[n]$ to $f_m[n]$, so that

$$F_m(z) = A_m(z) Y(z) \qquad \text{(p. 8-48)}$$

- Here, we are developing a lattice structure for a general IIR filter, so we continue to use the IIR convention that $A_m(z)$ is the transfer function from $y[n]$ to $f_m[n]$:

$$F_m(z) = A_m(z)\,Y(z).$$

- But up to now, we have only had to consider the polynomial $A_m(z)$ for <u>one</u> value of m at a time.
  - Because of this, the coefficients a_m needed only one index.
  - For example, if we were considering $A_3(z)$, then it was clear that "a_2" meant the coefficient a_2 from the polynomial

$$A_3(z) = 1 + a_1 z^{-1} + \underset{\substack{\uparrow \\ \text{this one!}}}{a_2} z^{-2} + a_3 z^{-3}$$

- In order to develop the structure for implementing the <u>numerator</u> of a general IIR transfer function $H(z)$, we will need to consider the polynomials $A_m(z)$ for more than one "m" at the same time.

- This will require a <u>second</u> index on the coefficients:
  - one index to tell which polynomial $A_m(z)$ we are talking about,
  - and a second index to tell which coefficient we are talking about in that polynomial.

- We will use a superscript with parentheses for the index that tells <u>which</u> <u>polynomial</u> $A_m(z)$.
- we will use a subscript for the index that tells <u>which</u> <u>coefficient</u> in the polynomial $A_m(z)$.

- So we will write $a_j^{(m)}$ to mean the coefficient of z^{-j} in the polynomial $A_m(z)$.

- With this new notation, the polynomials $A_m(z)$ will be written as:

$$m=0: \quad A_0(z) = 1$$

$$m=1: \quad A_1(z) = 1 + a_1^{(1)} z^{-1}$$

$$m=2: \quad A_2(z) = 1 + a_1^{(2)} z^{-1} + a_2^{(2)} z^{-2}$$

$$m=3: \quad A_3(z) = 1 + a_1^{(3)} z^{-1} + a_2^{(3)} z^{-2} + a_3^{(3)} z^{-3}$$

and so on...

- As before on p. 8-49 and p. 8-42, the reflection coefficients are given by

$$k_m = a_m^{(m)}, \quad 1 \leq m \leq N$$

- Now suppose we are given an N^{th} order IIR transfer function

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}}{1 + a_1^{(N)} z^{-1} + a_2^{(N)} z^{-2} + \cdots + a_N^{(N)} z^{-N}} \equiv \frac{B(z)}{A_N(z)}$$

- It is required that $q \leq N$.
- If $q > N$, then use the procedure described on page 8-52 to factor out an FIR transfer function first.

☆☆

- Here are the steps to realize $H(z)$ in a general IIR lattice structure:

Step ①: if $q < N$, then define new numerator coefficients

$$b_{q+1} = b_{q+2} = \cdots = b_N = 0$$

to make the order of the numerator equal to the order of the denominator.

For example, if $H(z) = \dfrac{1 + \frac{1}{2} z^{-1}}{1 + \frac{5}{8} z^{-1} + \frac{1}{4} z^{-2} + \frac{1}{3} z^{-3}}$,

then re-write it as

$$H(z) = \frac{1 + \frac{1}{2} z^{-1} + 0 z^{-2} + 0 z^{-3}}{1 + \frac{5}{8} z^{-1} + \frac{1}{4} z^{-2} + \frac{1}{3} z^{-3}}$$

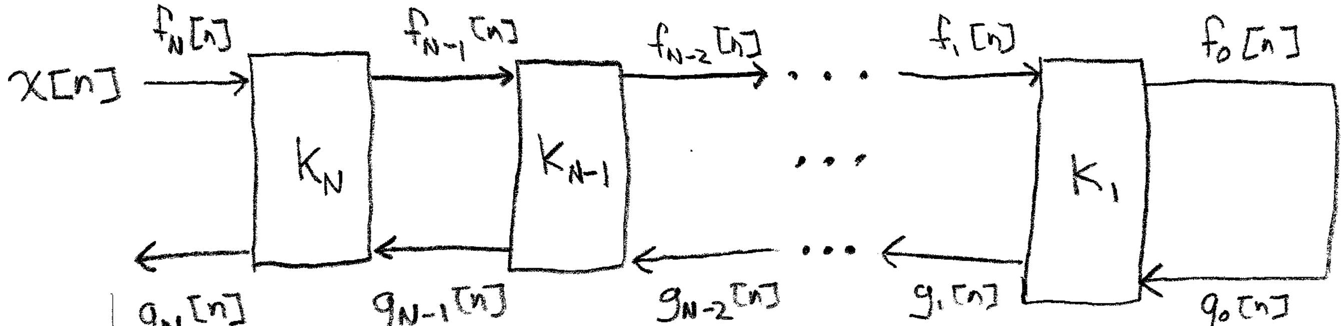
→ In this example, we have $N = 3$ and:

$b_0 = 1 \quad b_1 = 1/2 \quad b_2 = 0 \quad b_3 = 0$

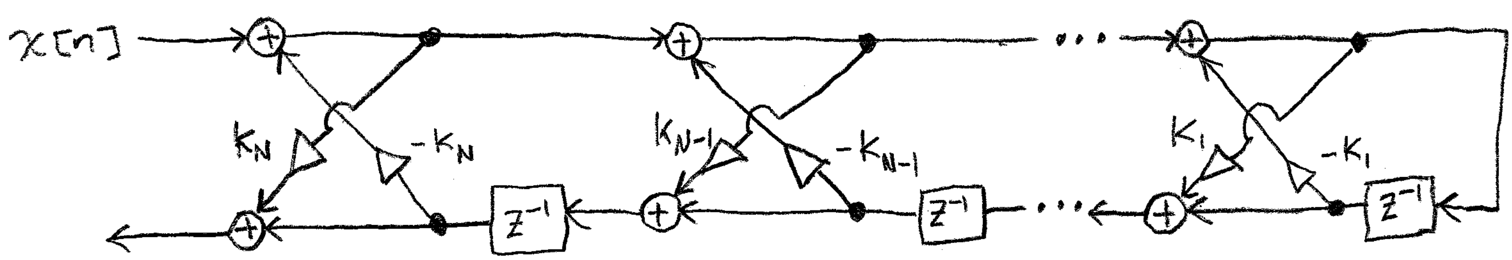
$a_1^{(3)} = 5/8 \quad a_2^{(3)} = \frac{1}{4} \quad a_3^{(3)} = \frac{1}{3}$

Step 2 : use the procedure given on pages 8-45 through 8-50 to realize the denominator $A_N(z)$ in an all-pole IIR lattice structure.

- After step 2, you will have a diagram that looks like this:

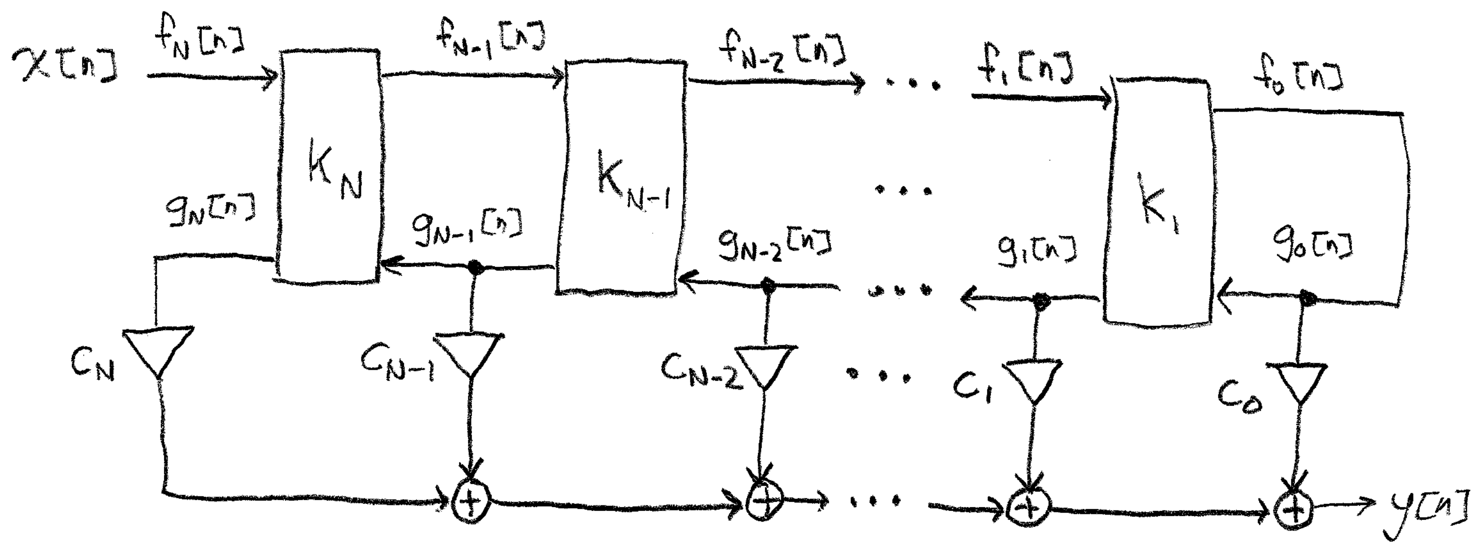


- With all the details shown, it looks like this:



Step 3 : add an FIR structure with delay line taps on the lower signal path to implement the numerator.

- The structure then looks like this:



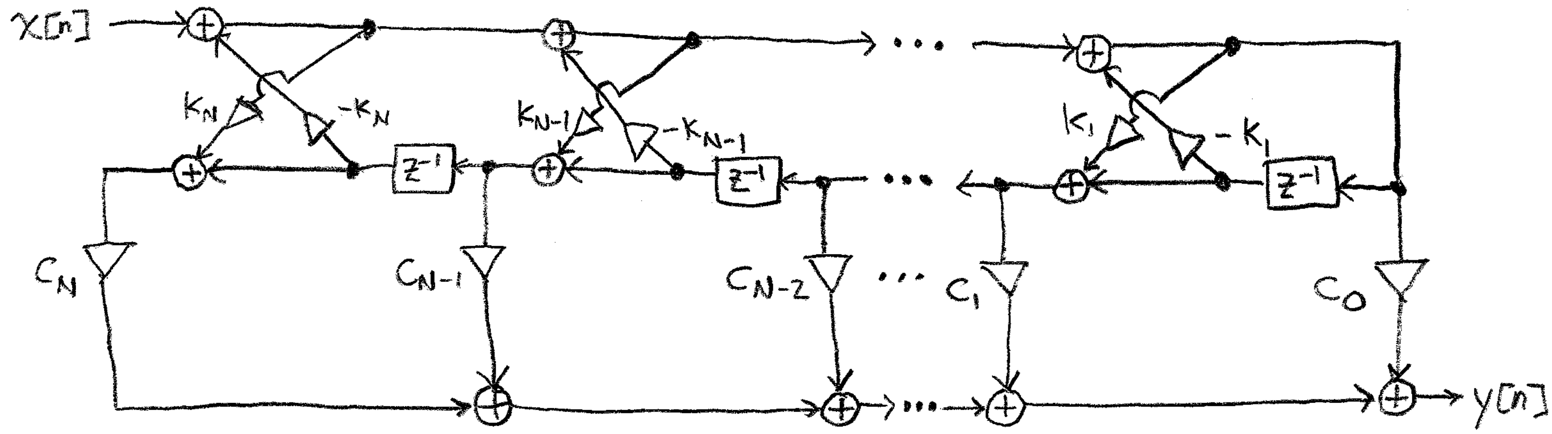
- How to find the coefficients c_m :

  - Start with $c_N = b_N$.

  - Then, for $m = N-1, N-2, \ldots, 0$, the formula is given by

$$c_m = b_m - \sum_{j=m+1}^{N} c_j a_{j-m}^{(j)}$$

- Once you have computed all the coefficients c_m for $0 \le m \le N$, the final IIR lattice structure (Gray-Markel structure) looks like this:



EX: (Example 8.5.3 from "Schaum's Outline of DSP" by Monson Hayes)

A 3rd-order low-pass digital elliptic filter with cutoff frequency $\omega_c = 0.5\pi$ rad/sample has transfer function

$$H(z) = \frac{0.2759 + 0.5121 z^{-1} + 0.5121 z^{-2} + 0.2759 z^{-3}}{1 - 0.0010 z^{-1} + 0.6546 z^{-2} - 0.0775 z^{-3}}$$

Realize this filter in an IIR lattice structure.

Step ①: nothing to do... since $q = N = 3$.

We have $B(z) = 0.2759 + 0.5121 z^{-1} + 0.5121 z^{-2} + 0.2759 z^{-3}$

$b_0 = 0.2759 \quad b_1 = 0.5121 \quad b_2 = 0.5121 \quad b_3 = 0.2759$

→

- From the denominator of $H(z)$, we have (8-59)

$$A_3(z) = 1 - 0.0010z^{-1} + 0.6546z^{-2} - 0.0775z^{-3}$$

$$a_1^{(3)} = -0.0010 \qquad a_2^{(3)} = 0.6546 \qquad a_3^{(3)} = -0.0775$$

Step ②: realize the denominator of $H(z)$ in an all-pole IIR lattice structure.

$$\Rightarrow \quad K_3 = a_3^{(3)} = -0.0775$$

Step-down recursion:

$$A_2(z) = \frac{1}{1-K_3^2}\left[A_3(z) - K_3 z^{-3} A_3(z^{-1})\right]$$

$$= \frac{1}{1-(-0.0775)^2}\Big[\left(1 - 0.0010z^{-1} + 0.6546z^{-2} - 0.0775z^{-3}\right)$$

$$- (-0.0775)z^{-3}\left(1 - 0.0010z + 0.6546z^2 - 0.0775z^3\right)\Big]$$

$$= 1.0060\Big[\left(1 - 0.0010z^{-1} + 0.6546z^{-2} - 0.0775z^{-3}\right)$$

$$+ 0.0775\left(z^{-3} - 0.0010z^{-2} + 0.6546z^{-1} - 0.0775\right)\Big]$$

$$= 1.0060\Big[\left(1 - 0.0010z^{-1} + 0.6546z^{-2} - 0.0775z^{-3}\right)$$

$$+ 0.0775z^{-3} - 77.500\times10^{-6}z^{-2} + 50.732\times10^{-3}z^{-1} - 6.0063\times10^{-3}\Big]$$

$$= 1.0060\left[993.99\times10^{-3} + 49.732\times10^{-3}z^{-1} + 654.52\times10^{-3}z^{-2} + 0z^{-3}\right]$$

$$= 1 + 0.0500z^{-1} + 0.6585z^{-2}$$

$$\Rightarrow a_1^{(2)} = 0.0500, \quad a_2^{(2)} = 0.6585, \quad K_2 = a_2^{(2)} = 0.6585$$

Step-down recursion:

$$A_1(z) = \frac{1}{1-K_2^2}\left[A_2(z) - K_2 z^{-2} A_2(z^{-1})\right]$$

$$= \frac{1}{1-(0.6585)^2}\Big[(1+0.0500z^{-1}+0.6585z^{-2})$$

$$-0.6585\,z^{-2}(1+0.0500z+0.6585z^2)\Big]$$

$$= 1+0.0301z^{-1}$$

$$\Rightarrow a_1^{(1)} = 0.0301\ , \quad K_1 = a_1^{(1)} = 0.0301$$

- Summary of coefficients calculated through step 2:

$a_1^{(3)} = -0.0010$ $\quad a_2^{(3)} = 0.6546$ $\quad a_3^{(3)} = -0.0775$

$a_1^{(2)} = 0.0500$ $\quad a_2^{(2)} = 0.6585$

$a_1^{(1)} = 0.0301$

$b_0 = 0.2759$ $\quad b_1 = 0.5121$ $\quad b_2 = 0.5121$ $\quad b_3 = 0.2759$

$K_3 = -0.0775$ $\quad K_2 = 0.6585$ $\quad K_1 = 0.0301$

Step 3: $C_3 = b_3 = 0.2759$ (8-61)

$$c_2 = b_2 - \sum_{j=3}^{3} c_j a_{j-2}^{(j)}$$

$$= b_2 - c_3 a_1^{(3)}$$

$$= 0.5121 - (0.2759)(-0.0010) = 0.5124$$

$$c_1 = b_1 - \sum_{j=2}^{3} c_j a_{j-1}^{(j)}$$

$$= b_1 - c_2 a_1^{(2)} - c_3 a_2^{(3)}$$

$$= 0.5121 - (0.5124)(0.0500) - (0.2759)(0.6546)$$

$$= 0.3062$$

$$c_0 = b_0 - \sum_{j=1}^{3} c_j a_j^{(j)}$$

$$= b_0 - c_1 a_1^{(1)} - c_2 a_2^{(2)} - c_3 a_3^{(3)}$$

$$= 0.2759 - (0.3062)(0.0301) - (0.5124)(0.6585) - (0.2759)(-0.0775)$$

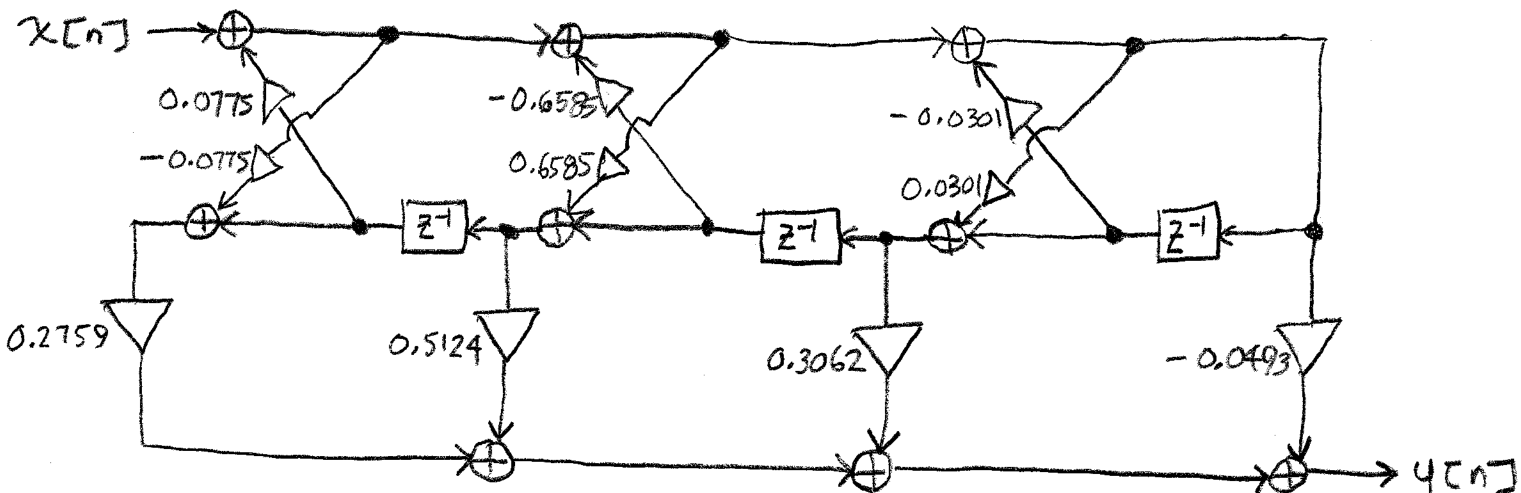
$$= -0.0493$$

- Summary of parameters calculated for the IIR lattice structure:

$K_3 = -0.0775$ $K_2 = 0.6585$ $K_1 = 0.0301$

$C_3 = 0.2759$ $C_2 = 0.5124$ $C_1 = 0.3062$ $C_0 = -0.0493$

- Here is the final IIR lattice structure for $H(z)$:



- The equation for c_m on page 8-57 can also be solved for b_m to obtain

$$b_m = c_m + \sum_{j=m+1}^{N} c_j a_{j-m}^{(j)} .$$

→ This gives us a way to find $H(z)$ from the lattice structure.

- First, use the step-up recursion (p. 8-48) to find $A_m(z)$ for $2 \le m \le N$ as on p. 8-49, starting with $A_0(z) = 1$.

- Then use the equation above to find b_m for $N-1 \geq m \geq 0$, starting with $b_N = c_N$.

EX: given the IIR lattice structure on page 8-62, find the transfer function $H(z)$.

From the diagram of the lattice structure, we have

$N=3$

$K_3 = -0.0775 \quad K_2 = 0.6585 \quad K_1 = 0.0301$

$C_3 = 0.2759 \quad C_2 = 0.5124 \quad C_1 = 0.3062 \quad C_0 = -0.0493$

- Proceeding as in the example on page 8-49, we have:

$A_0(z) = 1$

$$A_1(z) = A_0(z) + K_1 z^{-1} A_0(z^{-1})$$
$$= 1 + 0.0301 z^{-1}$$

$\Rightarrow a_1^{(1)} = 0.0301$

$$A_2(z) = A_1(z) + K_2 z^{-2} A_1(z^{-1})$$
$$= (1 + 0.0301 z^{-1}) + (0.6585) z^{-2} (1 + 0.0301 z)$$
$$= 1 + 0.0500 z^{-1} + 0.6585 z^{-2}$$

$\Rightarrow a_1^{(2)} = 0.0500 \qquad a_2^{(2)} = 0.6585$

$\longrightarrow$

$$A_3(z) = A_2(z) + k_3 z^{-3} A_2(z^{-1})$$

$$= (1 + 0.0500 z^{-1} + 0.6585 z^{-2})$$

$$+ (-0.0775) z^{-3} (1 + 0.0500 z + 0.6585 z^2)$$

$$= 1 - 0.0010 z^{-1} + 0.6623 z^{-2} - 0.0775 z^{-3}$$

There is some roundoff error in this

$$\Rightarrow \quad a_1^{(3)} = -0.0010 \quad a_2^{(3)} = 0.6623 \quad a_3^{(3)} = -0.0775$$

– Now, $b_3 = c_3 = 0.2759$

$$b_2 = c_2 + \sum_{j=3}^{3} c_j a_{j-2}^{(j)} = c_2 + c_3 a_1^{(3)}$$

$$= 0.5124 + 0.2759(-0.0010) = 0.5121$$

$$b_1 = c_1 + \sum_{j=2}^{3} c_j a_{j-1}^{(j)} = c_1 + c_2 a_1^{(2)} + c_3 a_2^{(3)}$$

$$= 0.3062 + (0.5124)(0.0500) + 0.2759(0.6623)$$

$= 0.5145$ (has some round off error)

$$b_0 = c_0 + \sum_{j=1}^{3} c_j a_j^{(j)} = c_0 + c_1 a_1^{(1)} + c_2 a_2^{(2)} + c_3 a_3^{(3)}$$

$$= -0.0493 + 0.3062(0.0301) + 0.5124(0.6585) + 0.2759(-0.0775)$$

$$= 0.2759$$

$\longrightarrow$

- The transfer function is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{A_3(z)}$$

$$H(z) = \frac{0.2759 + 0.5145 z^{-1} + 0.5121 z^{-2} + 0.2759 z^{-3}}{1 - 0.0010 z^{-1} + 0.6623 z^{-2} - 0.0775 z^{-3}}$$

→ Comparing to the original $H(z)$ on page 8-58, we see that there is some roundoff error in two of the coefficients (b_1 and $a_2^{(3)}$).

Matlab routines for lattice structures:

latc2tf : convert lattice parameters to transfer function

tf2latc : convert transfer function to lattice parameters

poly2rc : Convert coefficient vector for $A_m(z)$ into reflection coefficients