Chapter 8

Digital Filter Structures

清大電機系林嘉文
cwlin@ee.nthu.edu.tw
03-5731152

Original PowerPoint slides prepared by S. K. Mitra

Block Diagram Representation

- The convolution sum description of an LTI discrete-time system can, in principle, be used to implement the system
- Here the input-output relation involves a finite sum of products:

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

- On the other hand, an FIR system can be implemented using the convolution sum which is a finite sum of products:

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

- The implementation of an LTI digital filter can be either in software or hardware form, depending on applications
- In either case, the signal variables and the filter coefficients cannot be represented with infinite precision

Block Diagram Representation

- A structural representation using interconnected basic building blocks is the first step in the hardware or software implementation of an LTI digital filter
- In the time domain, the input-output relations of an LTI digital filter is given by the convolution sum

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

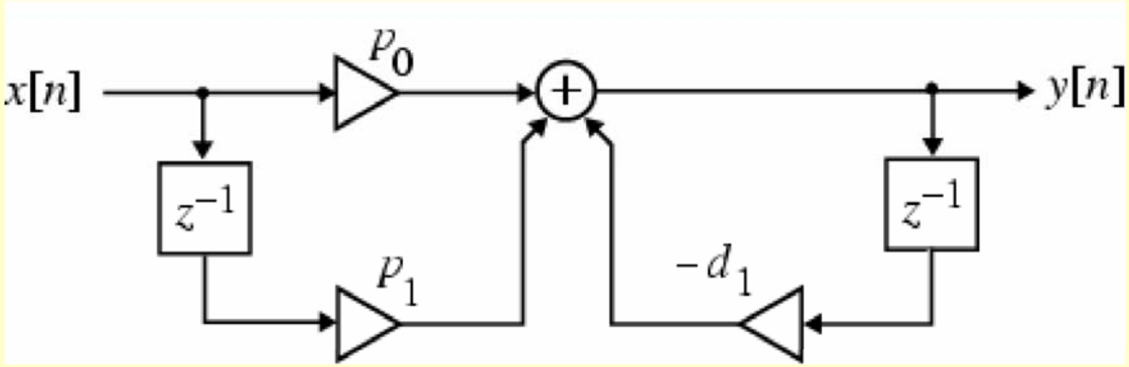
or, by the linear constant coefficient difference equation

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid computational algorithm

Block Diagram Representation

- Consider the causal first-order LTI digital filter shown below



- The filter is described by the difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

- Using the above equation we can compute $y[n]$ for $n \geq 0$ knowing $y[-1]$ and the input $x[n]$ for $n \geq -1$

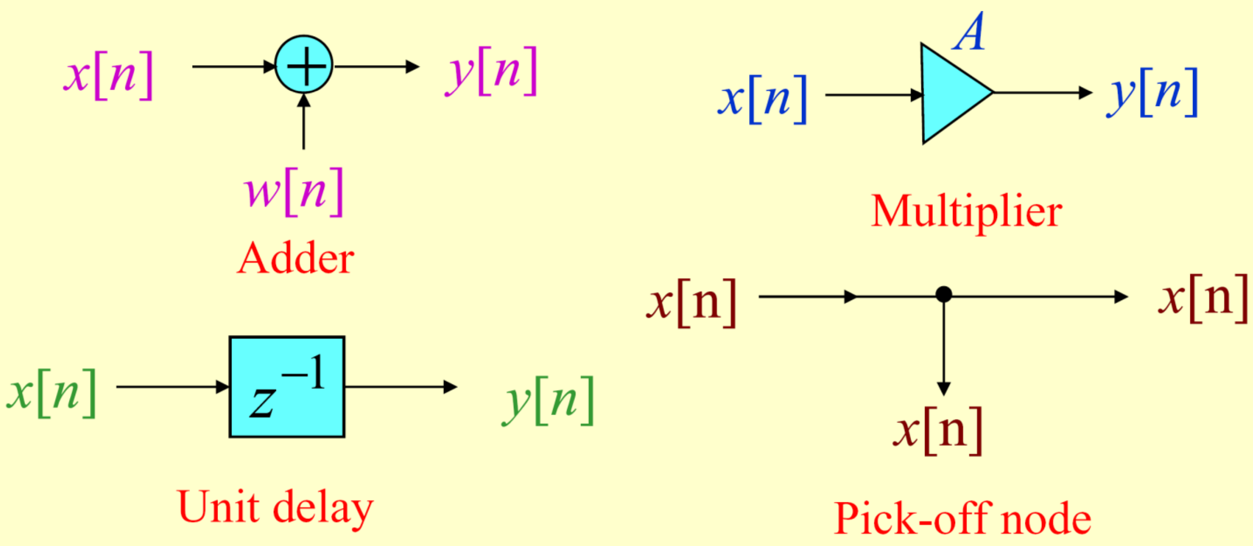
$$y[0] = -d_1 y[-1] + p_0 x[0] + p_1 x[-1]$$

$$y[1] = -d_1 y[0] + p_0 x[1] + p_1 x[0]$$

$$\cdots$$

Basic Building Blocks

- The computational algorithm of an LTI digital filter can be conveniently represented in block diagram form using the basic building blocks shown below



Basic Building Blocks

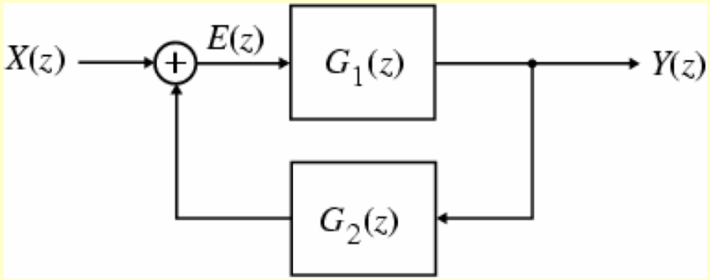
- Advantages of block diagram representation
  - Easy to write down the computational algorithm by inspection
  - Easy to analyze the block diagram to determine the explicit relation between the output and input
  - Easy to manipulate a block diagram to derive other “equivalent” block diagrams yielding different computational algorithms
  - Easy to determine the hardware requirements
  - Easier to develop block diagram representations from the transfer function directly

Analysis of Block Diagrams

- Write down the expressions for the output signals of each adder as a sum of its input signals, and develop a set of equations relating the filter input and output signals in terms of all internal signals
- Eliminate the unwanted internal variables to obtain the expression for the output signal as a function of the input signal and the filter parameters (multiplier coefficients)

Analysis of Block Diagrams

- Example - Consider the following single-loop feedback structure



- The output $E(z)$ of the adder is

$$E(z) = X(z) + G_2(z)Y(z)$$

- But from the figure

$$Y(z) = G_1(z)E(z)$$

- Eliminating $E(z)$ from the previous equations we arrive at

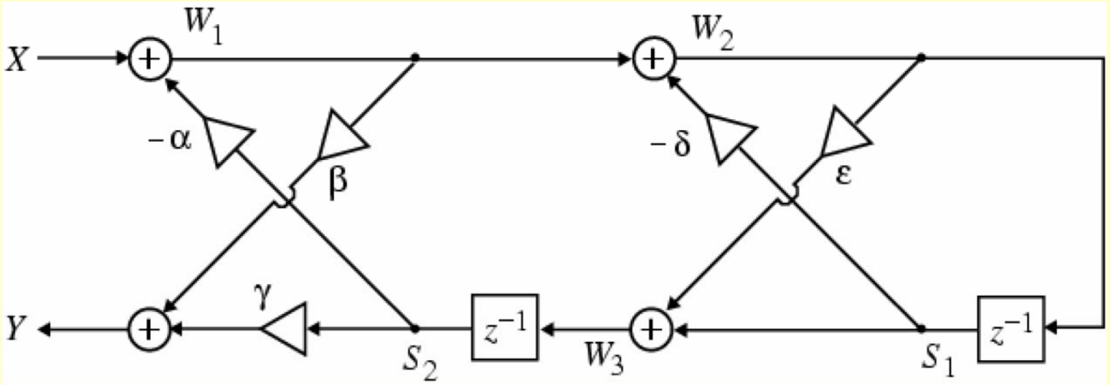
$$[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$$

which leads to

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

Analysis of Block Diagrams

- Example – Analyze the following cascade lattice structure



- The output signals of the four adders are given by

$$W_1 = X - \alpha S_2$$

$$W_2 = W_1 - \delta S_1$$

$$W_3 = S_1 + \varepsilon W_2$$

$$Y = \beta W_1 + \gamma S_2$$

Original PowerPoint slides prepared by S. K. Mitra

Analysis of Block Diagrams

- From the figure we observe

$$S_1 = z^{-1}W_2$$
$$S_2 = z^{-1}W_3$$

- Substituting the last two relations in the first four equations we get

$$W_1 = X - \alpha z^{-1}W_3$$
$$W_2 = W_1 - \delta z^{-1}W_2$$
$$W_3 = z^{-1}W_2 + \varepsilon W_2$$
$$Y = \beta W_1 + \gamma z^{-1}W_3$$

- From the second equation we get $W_2 = W_1/(1 + \delta z^{-1})$ and from the third equation we get $W_3 = (\varepsilon + z^{-1})W_2$

Original PowerPoint slides prepared by S. K. Mitra

Analysis of Block Diagrams

- Combining the last two equations we get

$$W_3 = \frac{\varepsilon + z^{-1}}{1 + \delta z^{-1}} W_1$$

- Substituting the above equation in

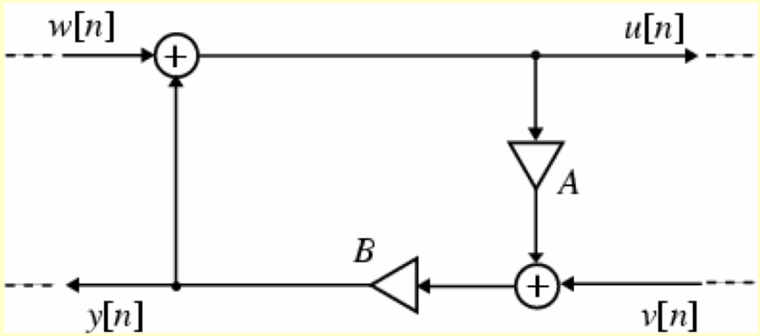
$$W_1 = X - \alpha z^{-1} W_3$$
$$Y = \beta W_1 + \gamma z^{-1} W_3$$

we finally arrive at

$$H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon) z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon) z^{-1} + \alpha z^{-2}}$$

The Delay-Free Loop Problem

- For physical realizability of the digital filter structure, it is necessary that the block diagram representation contains no delay-free loops
- To illustrate the delay-free loop problem consider the structure below



- Analysis of this structure yields

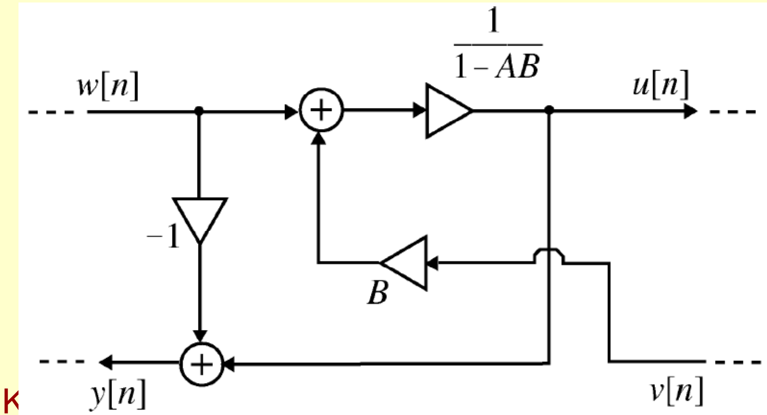
$$u[n] = w[n] + y[n]$$

$$y[n] = B(v[n] + Au[n])$$

- As a result, $y[n] = B(v[n] + A(w[n] + y[n]))$

The Delay-Free Loop Problem

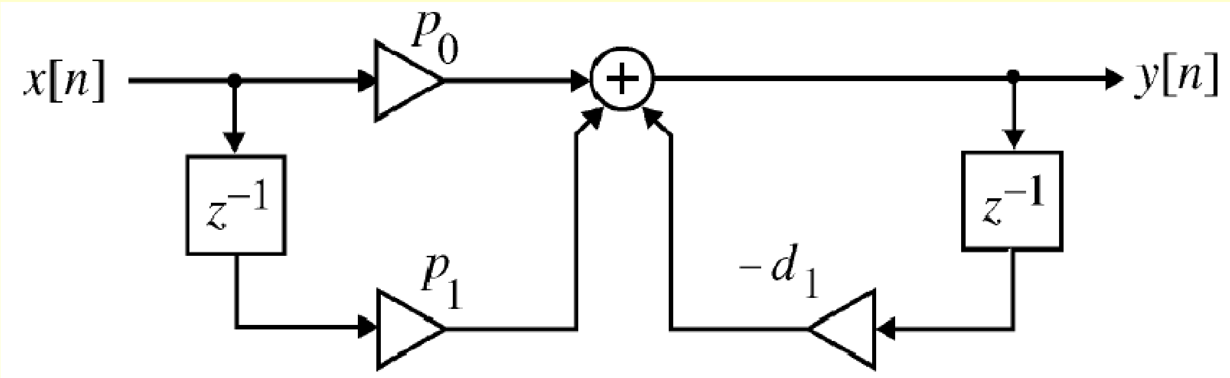
- The determination of the current value of $y[n]$ requires the knowledge of the same value
- However, this is physically impossible to achieve due to the finite time required to carry out all arithmetic operations on a digital machine
- **Solution**: Replace the portion of the overall structure containing the delay-free loops by an equivalent realization with no delay-free loops



Canonic & Noncanonic Structures

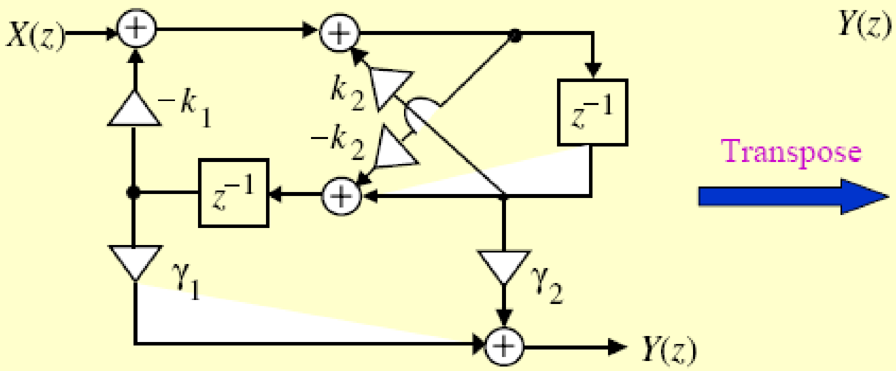
- A digital filter structure is said to be **canonic** if the number of delays in the block diagram representation is equal to the order of the transfer function
- Otherwise, it is a **noncanonic** structure
- The structure shown below is noncanonic as it employs two delays to realize a first-order difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

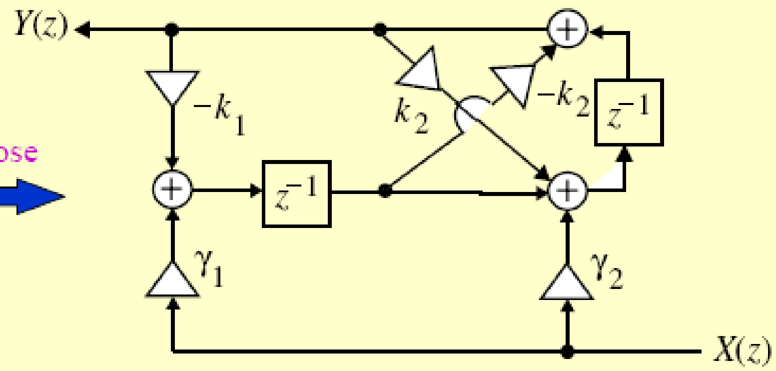


Equivalent Structures

- Two digital filter structures are defined to be **equivalent** if they have the same transfer function
- A simple way to generate an equivalent structure from a given realization is via the **transpose operation:**
  - Reverse all paths
  - Replace pick-off nodes by adders, and vice versa
  - Interchange the input and output nodes



Original structure

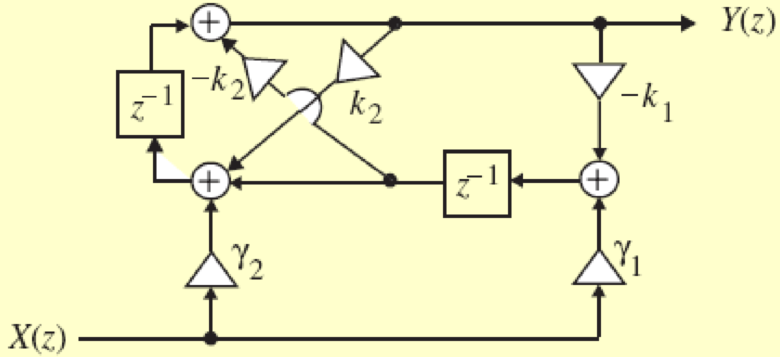


Transposed structure

Original PowerPoint slides prepared by S. K. Mitra

Equivalent Structures

- A redrawn transposed structure is shown below



- All other methods for developing equivalent structures are based on a specific algorithm for each structure
- There are literally an infinite number of equivalent structures realizing the same transfer function
- Under infinite precision arithmetic any given realization of a digital filter behaves identically to any other equivalent structure

Equivalent Structures

- However, in practice, due to the finite word-length limitations, a specific realization behaves totally differently from its other equivalent realizations
- Hence, it is important to choose a structure that has the least quantization effects when implemented using finite precision arithmetic
- One way to arrive at such a structure is to determine a large number of equivalent structures, analyze the finite word-length effects in each case, and select the one showing the least effects
- In certain cases, it is possible to develop a structure that by construction has the least quantization effects

Original PowerPoint slides prepared by S. K. Mitra

Basic FIR Digital Filter Structures

- A causal FIR filter of order *N* is characterized by a transfer function *H*(*z*) given by

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

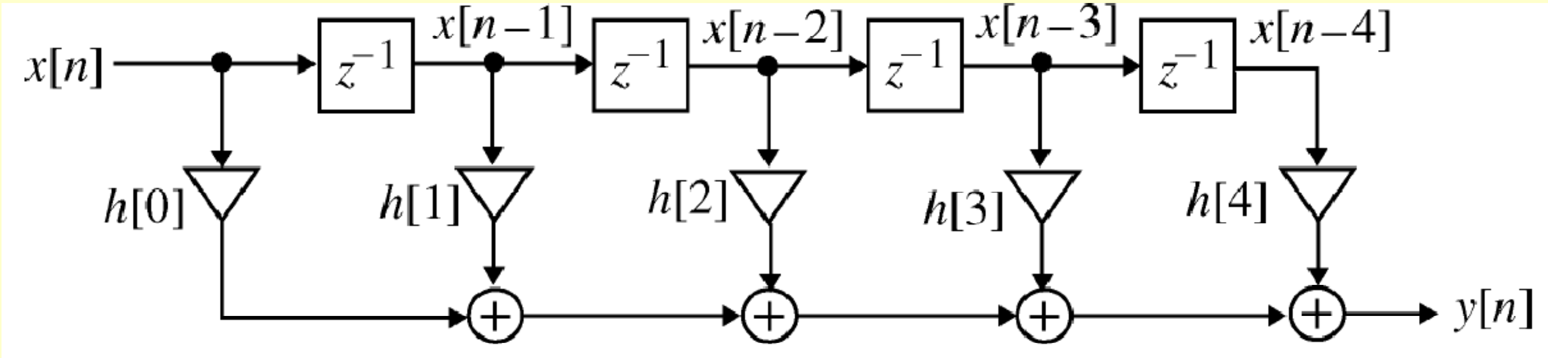
- In the time-domain the input-output relation of the above FIR filter is given by

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

- An FIR filter of order *N* is characterized by *N*+1 coefficients and, in general, require *N*+1 multipliers and *N* two-input adders
- Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called **direct form** structures

Direct Form FIR Filter Structures

- A direct form realization of an FIR filter can be readily developed from the convolution sum description as indicated below for $N = 4$



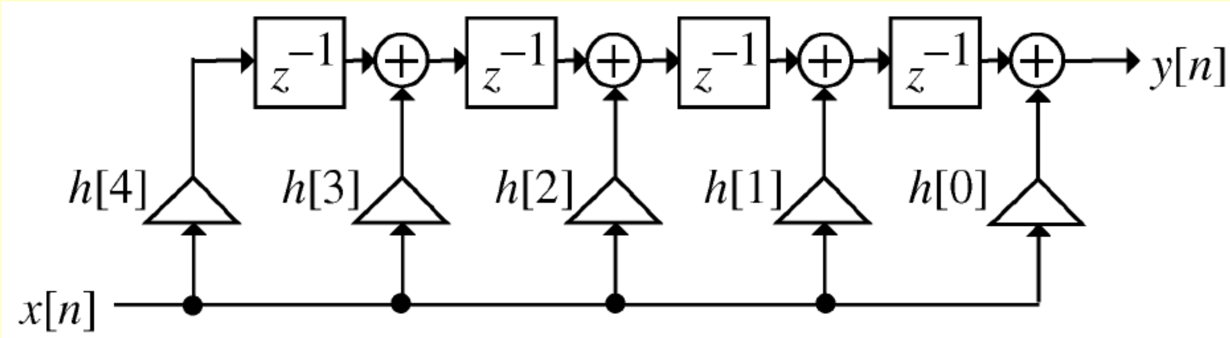
- An analysis of this structure yields

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] + h[4]x[n-4]$$

- The direct form structure is also known as a **transversal filter**

Direct Form FIR Filter Structures

- The transpose of the direct form structure shown earlier is indicated below



- Both direct form structures are canonic with respect to delays

Cascade Form FIR Filter Structures

- A higher-order FIR transfer function can also be realized as a cascade of second-order FIR sections and possibly a first-order section
- To this end we express $H(z)$ as

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

- Both direct form structures are canonic with respect to delays where $K = N/2$ if N is even, and $K = (N+1)/2$ if N is odd, with $\beta_{2K} = 0$

Cascade Form FIR Filter Structures

- A cascade realization for $N = 6$ is shown below
- To this end we express $H(z)$ as

- Each second-order section in the above structure can also be realized in the transposed direct form

Polyphase FIR Structures

- The polyphase decomposition of $H(z)$ leads to a parallel form structure
- To illustrate this approach, consider a causal FIR transfer function $H(z)$ with $N = 8$:

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8}$$

- $H(z)$ can be expressed as a sum of two terms, with one term containing the even-indexed coefficients and the other containing the odd-indexed coefficients

$$H(z) = (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6} + h[8]z^{-8}) + (h[1]z^{-1} + h[3]z^{-3} + h[5]z^{-5} + h[7]z^{-7})$$

$$= (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6} + h[8]z^{-8}) + z^{-1}(h[1] + h[3]z^{-2} + h[5]z^{-4} + h[7]z^{-6})$$

Polyphase FIR Structures

- By using the notation

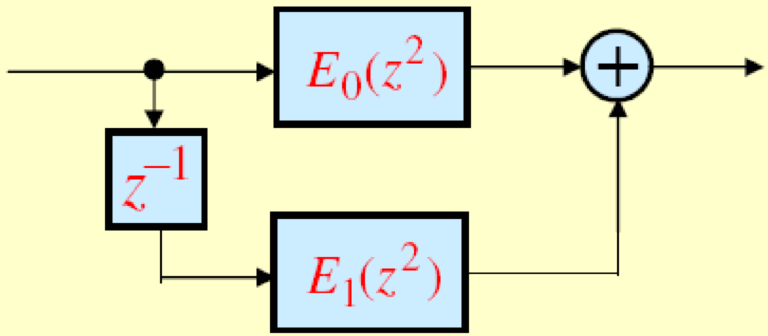
  $E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4}$

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

  we can express $H(z)$ as

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

- The above decomposition is more commonly known as the 2-branch **polyphase decomposition**



Polyphase FIR Structures

- In a similar manner, by grouping the terms in the original expression for $H(z)$, we can reexpress it in the form

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

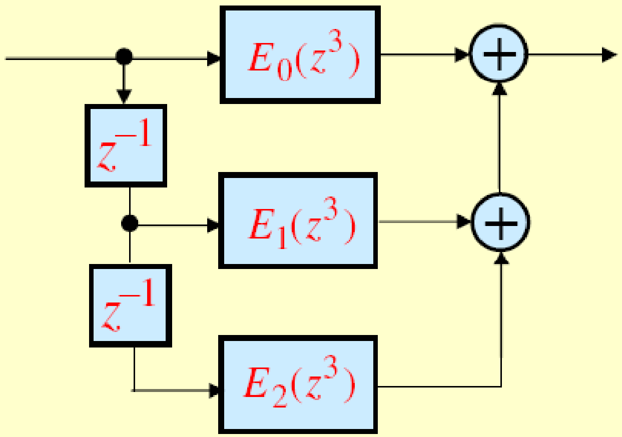
  where now

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

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

  $E_2(z) = h[2] + h[5]z^{-1} + h[8]z^{-2}$

- The 3-branch **polyphase decomposition** is shown below



Polyphase FIR Structures

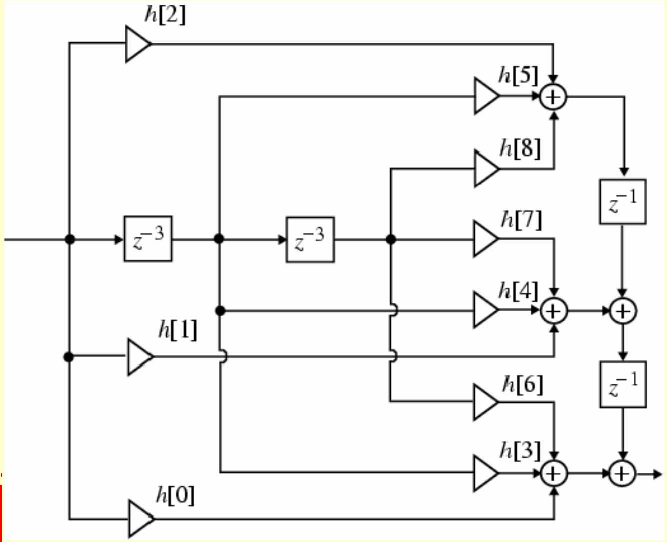
- In the general case, an *L*-branch polyphase decomposition of an FIR transfer function of order *N* is of the form

$$H(z) = \sum_{m=0}^{L-1} z^{-m} E_m(z^L)$$

where

$$E_m(z) = \sum_{n=0}^{\lfloor (N+1)/L \rfloor} h[Ln+m] z^{-m}$$ with $h[n] = 0$ for $n > N$

- The subfilters $E_m(z^L)$ are also FIR filters



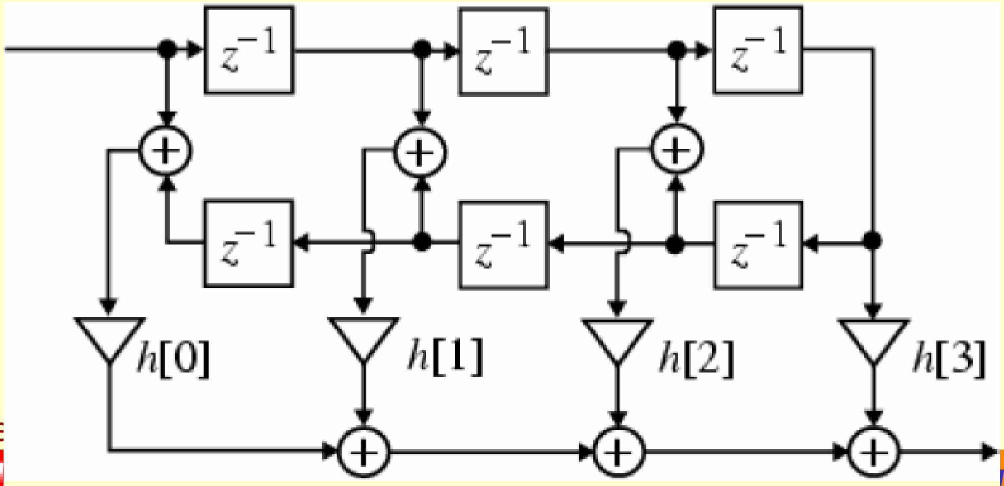
Linear Phase FIR Structures

- The symmetry (or antisymmetry) property of a linear-phase FIR filter can be exploited to reduce the number of multipliers into almost half of that in the direct form
- Consider a length-7 Type 1 FIR transfer function with a symmetric impulse response:

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

- Rewriting $H(z)$ in the form

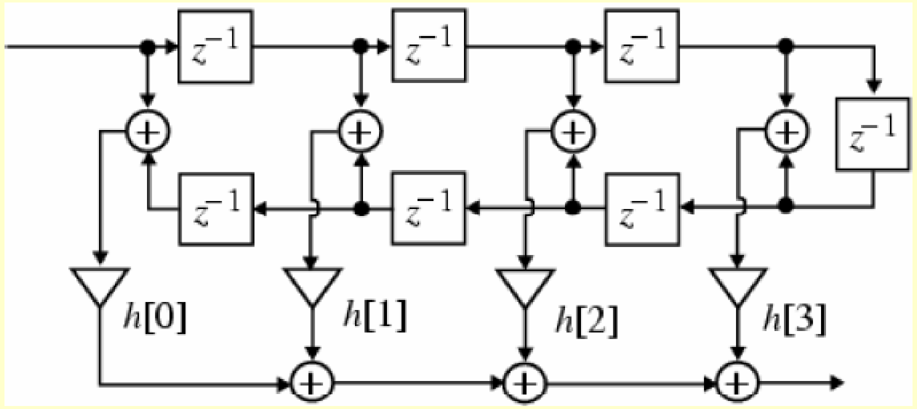
$$H(z) = h[0](1 + z^{-6}) + h[1](z^{-1} + z^{-5}) + h[2](z^{-2} + z^{-4}) + h[3]z^{-3}$$



Linear Phase FIR Structures

- A similar decomposition can be applied to a Type 2 FIR transfer function
- For example, a length-8 Type 2 FIR transfer function can be expressed as :

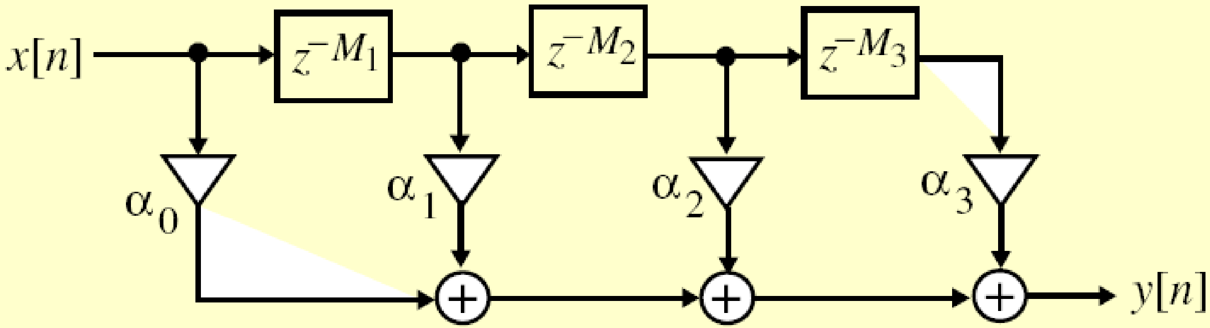
$$H(z) = h[0](1 + z^{-7}) + h[1](z^{-1} + z^{-6}) + h[2](z^{-2} + z^{-5}) + h[3](z^{-3} + z^{-4})$$



- Note: The Type 1 linear-phase structure for a length-7 FIR filter requires 4 multipliers, whereas a direct form realization requires 7 multipliers

Tapped Delay Lines

- In some applications, such as musical and sound processing, FIR filter structures of the form shown below are employed



- The structure consists of a chain of $M_1 + M_2 + M_3$ unit delays with taps at the input, at the end of first M_1 delays, at the end of next M_2 delays, and at the output
- Signals at these taps are then multiplied by constants α_0, α_1, α_2, and α_3 and added to form the output
- The structure is referred to as the tapped delay line

Original PowerPoint slides prepared by S. K. Mitra

Basic IIR Digital Filter Structures

- We concern about causal IIR digital filters characterized by a real rational transfer function of z^{-1} or, equivalently by a constant coefficient difference equation
- The realization of the causal IIR digital filters requires some form of feedback
- An N-th order IIR digital transfer function is characterized by $2N+1$ unique coefficients, and in general, requires $2N+1$ multipliers and $2N$ two-input adders for implementation
- **Direct form IIR filters**: Filter structures in which the multiplier coefficients are precisely the coefficients of the transfer function

Direct Form IIR Filter Structures

- Consider for simplicity a 3rd-order IIR filter with a transfer function

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

- We can implement *H*(*z*) as a cascade of two filter sections as shown below

where

$$H_1(z) = \frac{W(z)}{X(z)} = P(z) = p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}$$

$$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{D(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

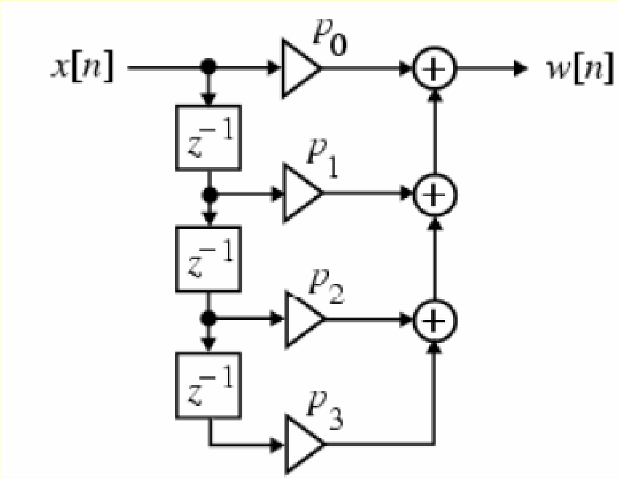
Direct Form IIR Filter Structures

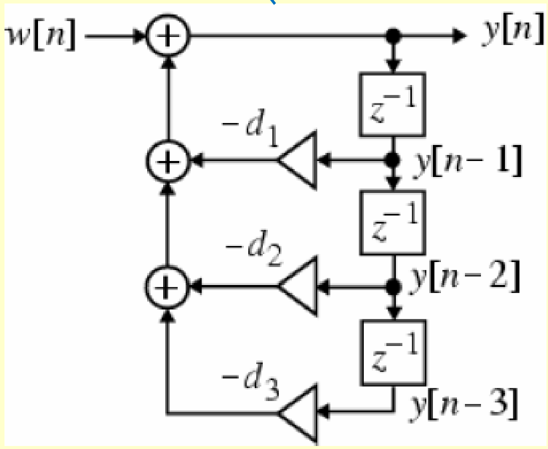
- The filter section can be seen to be an FIR filter and can be realized as shown below

$$w[n] = p_0 x[n] + p_1 x[n-1] + p_2 x[n-2] + p_3 x[n-3]$$

- The time-domain representation of is given by

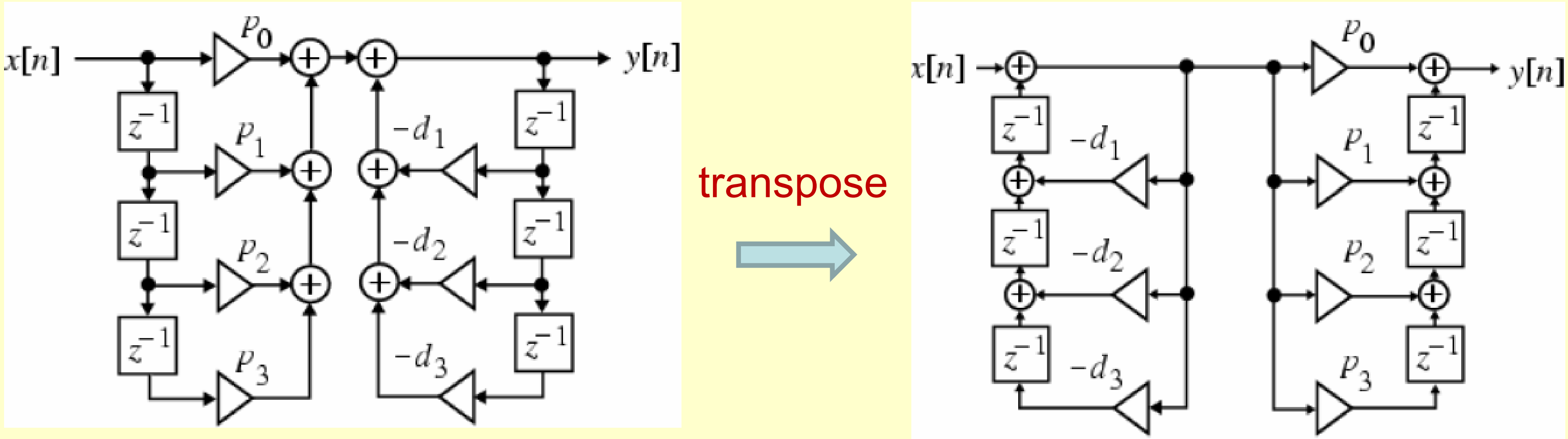
$$y[n] = w[n] - d_1 y[n-1] - d_2 y[n-2] - d_3 y[n-3]$$





Direct Form IIR Filter Structures

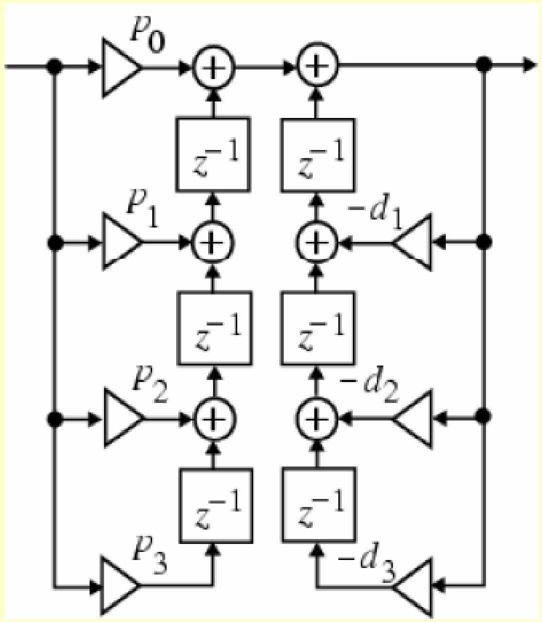
- A cascade of the two structures realizing and leads to the realization of shown below and is known as the **direct form I** structure

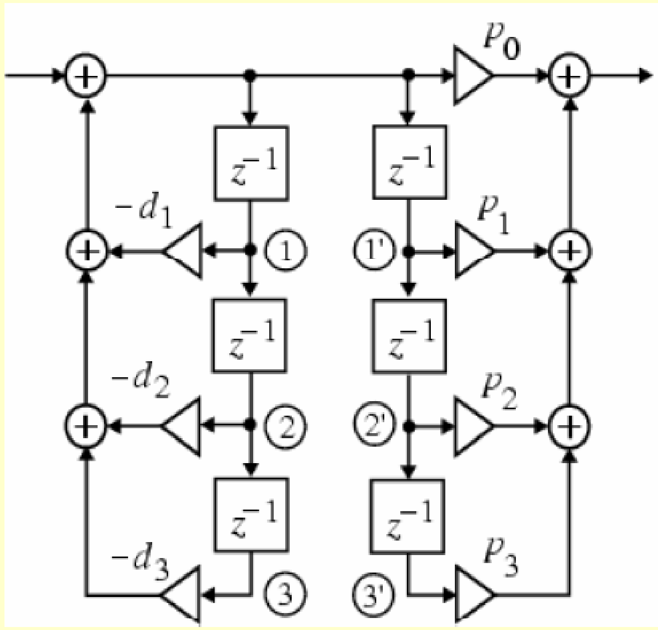


- The direct form I structure is non-canonic as it employs 6 delays to realize a 3rd-order transfer function
- A transpose of the direct form I structure is shown on the right and is called the **direct form I_t structure**

Direct Form IIR Filter Structures

- Various other non-canonic direct form structures can be derived by simple block diagram manipulations as shown below





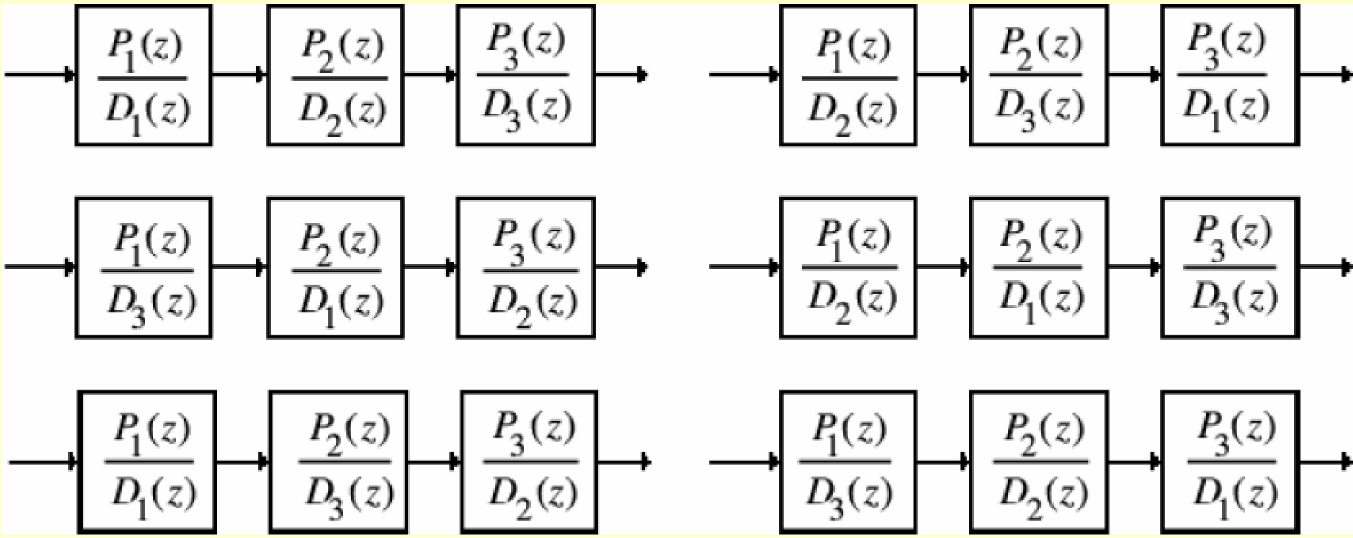
- Observe in the right-hand-side direct form structure, the signal variable at nodes (1) and (1') are the same, and hence the two top delays can be shared

Cascade Form IIR Filter Structures

- By expressing the numerator and the denominator polynomials of the transfer function as a product of polynomials of lower degree, a digital filter can be realized as a cascade of low-order filter sections
- Consider, for example, $H(z) = P(z)/D(z)$ expressed as

$$H(z) = \frac{P(z)}{D(z)} = \frac{P_1(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)}$$

- Examples of cascade:



Cascade Form IIR Filter Structures

- There are a total of 36 different cascade realizations of

$$H(z) = \frac{P(z)}{D(z)} = \frac{P_1(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)}$$

based on pole-zero-pairings and ordering

- Due to finite word-length effects, each such cascade realization behaves differently from others
- Usually, the polynomials are factored into a product of 1st-order and 2nd-order polynomials

$$H(z) = p_0 \prod_k \left(\frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

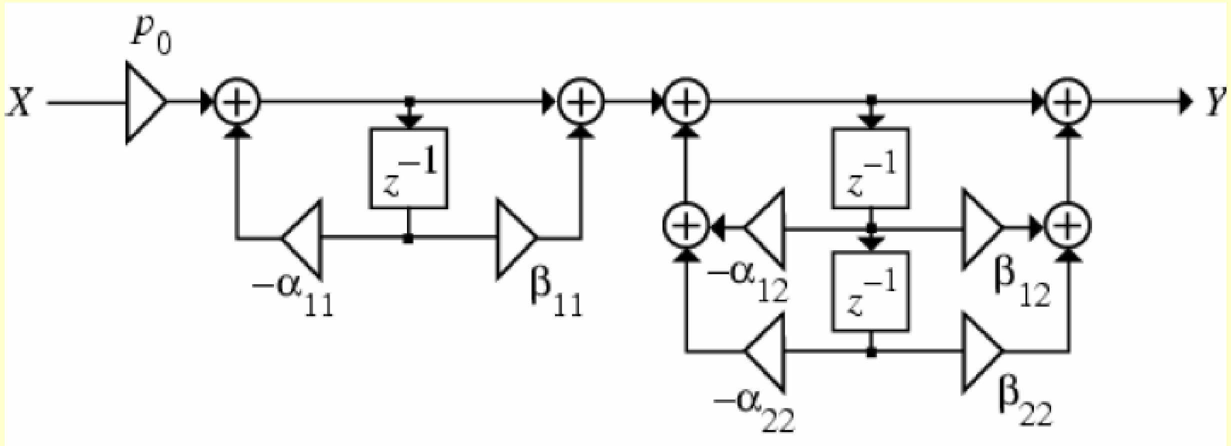
- In the above, for a first-order factor $\alpha_{2k} = \beta_{2k} = 0$

Cascade Form IIR Filter Structures

- Consider the 3rd-order transfer function

$$H(z) = p_0\left(\frac{1+\beta_{11}z^{-1}}{1+\alpha_{11}z^{-1}}\right)\left(\frac{1+\beta_{12}z^{-1}+\beta_{22}z^{-2}}{1+\alpha_{12}z^{-1}+\alpha_{22}z^{-2}}\right)$$

- One possible realization is shown below



Parallel Form IIR Filter Structures

- A partial-fraction expansion of the transfer function in z^{-1} leads to the **parallel form I** structure
- Assuming simple poles, the transfer function $H(z)$ can be expressed as

$$H(z) = \gamma_0 + \sum_k \left(\frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

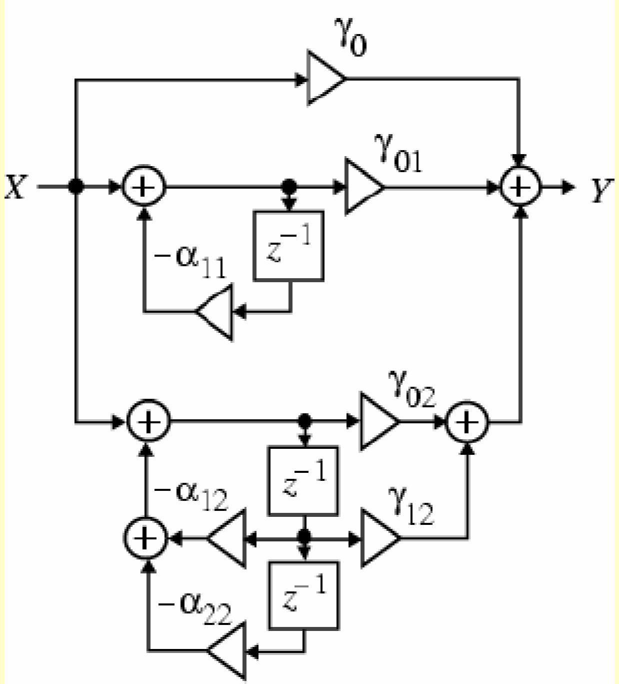
- In the above for a real pole $\alpha_{2k} = \gamma_{1k} = 0$
- A direct partial-fraction expansion of the transfer function in z leads to the **parallel form II** structure
- Assuming simple poles, the transfer function $H(z)$ can be expressed as

$$H(z) = \delta_0 + \sum_k \left(\frac{\delta_{1k} z^{-1} + \delta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

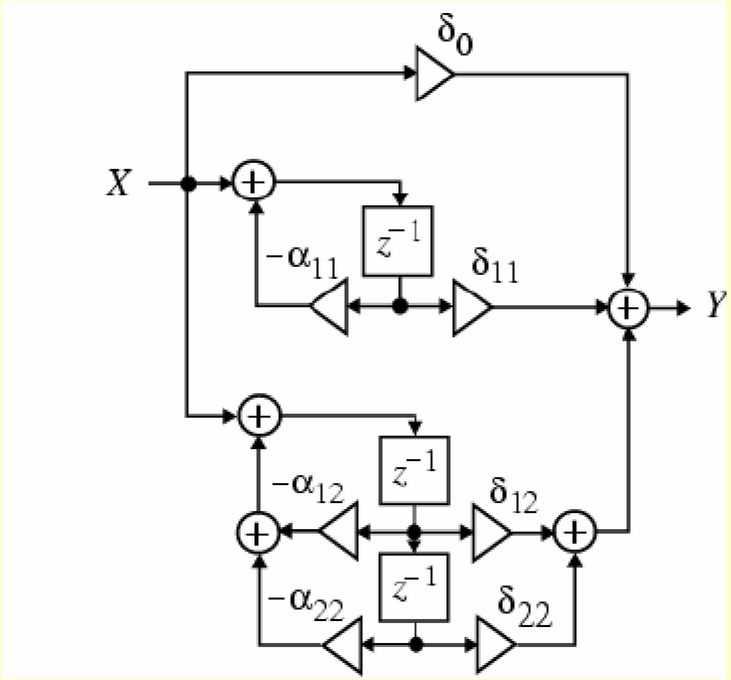
- In the above for a real pole $\alpha_{2k} = \delta_{2k} = 0$

Parallel Form IIR Filter Structures

- The two basic parallel realizations of a 3rd-order IIR transfer function are shown below



Parallel form I



Parallel form II

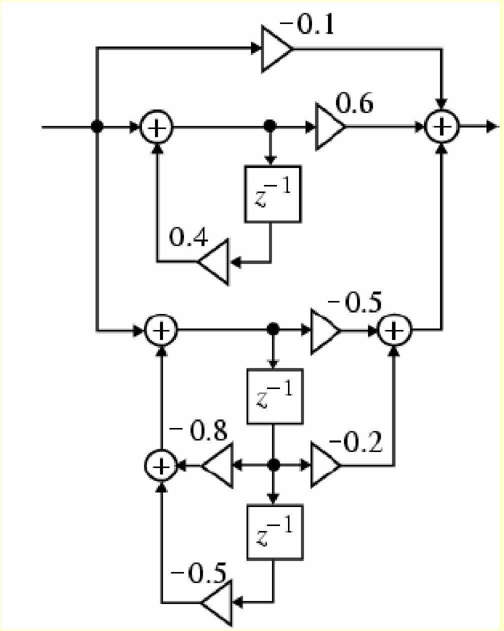
Parallel Form IIR Filter Structures

- Example - A partial-fraction expansion of

$$H(z) = \frac{0.44z^{-1} + 0.362z^{-2} + 0.02z^{-3}}{1 + 0.4z^{-1} + 0.18z^{-2} - 0.2z^{-3}}$$

in z^{-1} yields

$$H(z) = -0.1 + \frac{0.6}{1 - 0.4z^{-1}} + \frac{-0.5 - 0.2z^{-1}}{1 + 0.8z^{-1} + 0.5z^{-2}}$$

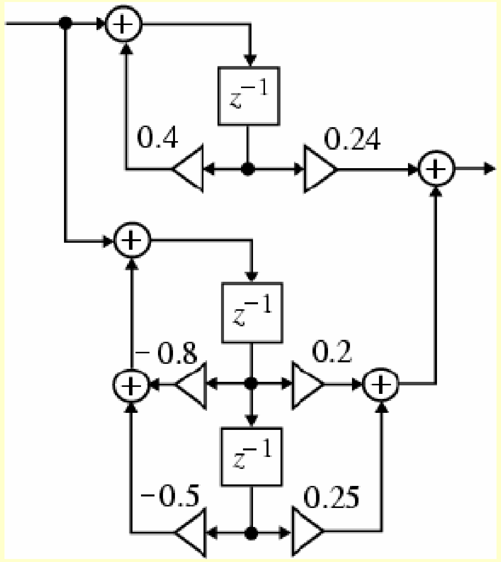


Parallel Form IIR Filter Structures

- Likewise, a partial-fraction expansion of $H(z)$ in z yields

$$H(z) = \frac{0.24z^{-1}}{1-0.4z^{-1}} + \frac{0.2z^{-1}+0.25z^{-2}}{1+0.8z^{-1}+0.5z^{-2}}$$

- The corresponding parallel form II realization is shown below

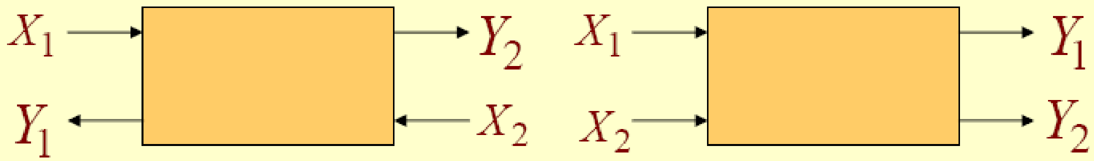


Realizations of All-Pass Filter

- An M-th order real-coefficient allpass transfer function $A_M(z)$ is characterized by M unique coefficients as here the numerator is the mirror-image polynomial of the denominator
- A direct form realization of $A_M(z)$ requires $2M$ multipliers
- Objective - Develop realizations of $A_M(z)$ requiring only M multipliers
- An arbitrary allpass transfer function can be expressed as a product of 2nd-order and/or 1st-order allpass transfer functions
- We consider first the minimum multiplier realization of a 1st-order and a 2nd-order allpass transfer functions

Digital Two-Pairs

- The LTI discrete-time systems considered so far are single-input, single-output structures characterized by a transfer function
- Often, such a system can be efficiently realized by interconnecting two-input, two-output structures, more commonly called **two-pairs**
- Figures below show two commonly used block diagram representations of a two-pair



- Here Y_1 and Y_2 denote the two outputs, and X_1 and X_2 denote the two inputs, where the dependencies on the variable *z* has been omitted for simplicity

Digital Two-Pairs

- The input-output relation of a digital two-pair is given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

- In the above relation the matrix τ given by

$$\tau = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$$

is called the **transfer matrix** of the two-pair

- It follows from the input-output relation that the transfer parameters can be found as follows:

$$t_{11} = \left.\frac{Y_1}{X_1}\right|_{X_2=0}, \qquad t_{12} = \left.\frac{Y_1}{X_2}\right|_{X_1=0}$$

$$t_{21} = \left.\frac{Y_2}{X_1}\right|_{X_2=0}, \qquad t_{22} = \left.\frac{Y_2}{X_2}\right|_{X_1=0}$$

Digital Two-Pairs

- An alternate characterization of the two-pair is in terms of its chain parameters as

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} Y_2 \\ X_2 \end{bmatrix}$$

where the matrix Γ given by

$$\Gamma = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

is called the **chain matrix** of the two-pair

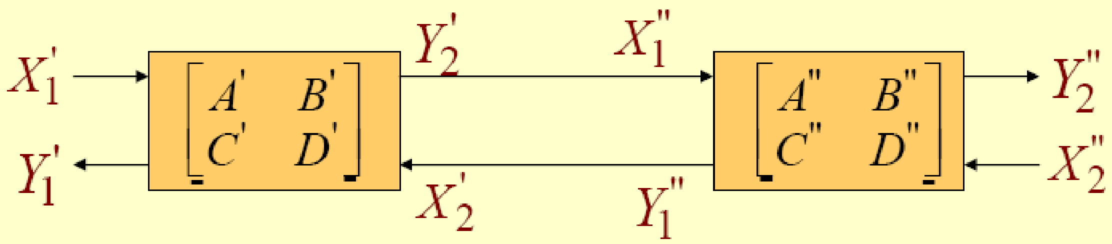
- The relation between the transfer parameters and the chain parameters are given by

$$t_{11} = \frac{C}{A},\ t_{12} = \frac{AD - BC}{A},\ t_{21} = \frac{1}{A},\ t_{22} = -\frac{C}{A}$$

$$A = \frac{1}{t_{21}},\ B = -\frac{t_{22}}{t_{21}},\ C = \frac{t_{11}}{t_{21}},\ D = \frac{t_{12}t_{21} - t_{11}t_{22}}{t_{21}}$$

Digital Two-Pairs

- **Cascade Connection** - Γ-cascade



Here

$$\begin{bmatrix} X_1' \\ Y_1' \end{bmatrix} = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} \begin{bmatrix} Y_2' \\ X_2' \end{bmatrix}$$

$$\begin{bmatrix} X_1'' \\ Y_1'' \end{bmatrix} = \begin{bmatrix} A'' & B'' \\ C'' & D'' \end{bmatrix} \begin{bmatrix} Y_2'' \\ X_2'' \end{bmatrix}$$

- As a result,

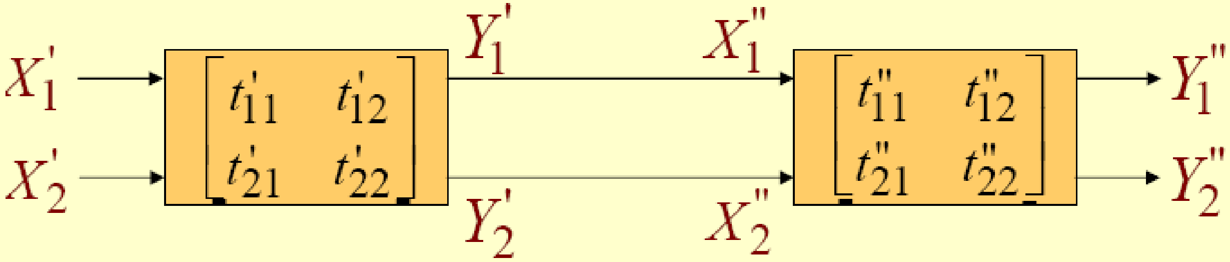
$$\begin{bmatrix} X_1' \\ Y_1' \end{bmatrix} = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} \begin{bmatrix} A'' & B'' \\ C'' & D'' \end{bmatrix} \begin{bmatrix} Y_2'' \\ X_2'' \end{bmatrix}$$

- Hence

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} \begin{bmatrix} A'' & B'' \\ C'' & D'' \end{bmatrix}$$

Digital Two-Pairs

- **Cascade Connection** - τ-cascade



Here

$$\begin{bmatrix} Y_1' \\ Y_2' \end{bmatrix} = \begin{bmatrix} t_{11}' & t_{12}' \\ t_{21}' & t_{22}' \end{bmatrix} \begin{bmatrix} X_1' \\ X_2' \end{bmatrix}$$

$$\begin{bmatrix} Y_1'' \\ Y_2'' \end{bmatrix} = \begin{bmatrix} t_{11}'' & t_{12}'' \\ t_{21}'' & t_{22}'' \end{bmatrix} \begin{bmatrix} X_1'' \\ X_2'' \end{bmatrix}$$

- As a result,

$$\begin{bmatrix} Y_1'' \\ Y_2'' \end{bmatrix} = \begin{bmatrix} t_{11}'' & t_{12}'' \\ t_{21}'' & t_{22}'' \end{bmatrix} \begin{bmatrix} t_{11}' & t_{12}' \\ t_{21}' & t_{22}' \end{bmatrix} \begin{bmatrix} X_1' \\ X_2' \end{bmatrix}$$

- Hence

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = \begin{bmatrix} t_{11}'' & t_{12}'' \\ t_{21}'' & t_{22}'' \end{bmatrix} \begin{bmatrix} t_{11}' & t_{12}' \\ t_{21}' & t_{22}' \end{bmatrix}$$

Digital Two-Pairs

- **Constrained Two-Pair**

- It can be shown that

$$H(z) = \frac{Y_1}{X_1} = \frac{C + D \cdot G(z)}{A + B \cdot G(z)}$$

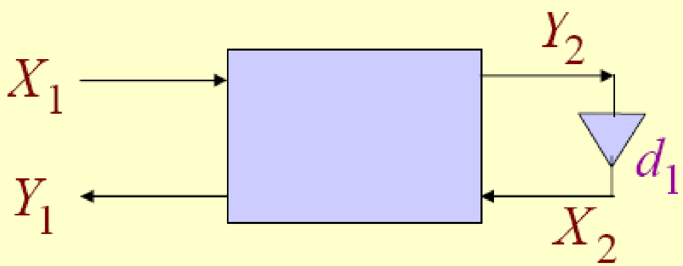
$$= t_{11} + \frac{t_{12} t_{21} G(z)}{1 - t_{22} G(z)}$$

First-Order All-Pass Filter Structures

- Consider first the 1st-order allpass transfer function given by

$$A_1(z) = \frac{d_1 + z^{-1}}{1 + d_1 z^{-1}}$$

- We shall realize the above transfer function in the form a structure containing a single multiplier d_1 as shown below



- We express the transfer function $A_1(z) = Y_1 / X_1$ in terms of the transfer parameters of the two-pair as

$$A_1(z) = t_{11} + \frac{t_{12} t_{21} d_1}{1 - d_1 t_{22}} = \frac{t_{11} - d_1 (t_{11} t_{22} - t_{12} t_{21})}{1 - d_1 t_{22}}$$

First-Order All-Pass Filter Structures

- A comparison of the above with

$$A_1(z) = \frac{d_1 + z^{-1}}{1 + d_1 z^{-1}}$$

  Yields

$$t_{11} = z^{-1},\ t_{22} = -z^{-1},\ t_{11}t_{22} - t_{12}t_{21} = -1$$

- Substituting $t_{11} = z^{-1}$ and $t_{22} = -z^{-1}$ in $t_{11}t_{22} = -1$ we get

$$t_{12}t_{21} = 1 - z^{-2}$$

- There are 4 possible solutions to the above equation:

  Type 1A: $t_{11} = z^{-1}$, $t_{22} = -z^{-1}$, $t_{12} = 1 - z^{-2}$, $t_{21} = 1$

  Type 1B: $t_{11} = z^{-1}$, $t_{22} = -z^{-1}$, $t_{12} = 1 + z^{-1}$, $t_{21} = 1 - z^{-1}$

  Type $1A_t$: $t_{11} = z^{-1}$, $t_{22} = -z^{-1}$, $t_{12} = 1$, $t_{21} = 1 - z^{-2}$

  Type $1B_t$: $t_{11} = z^{-1}$, $t_{22} = -z^{-1}$, $t_{12} = 1 - z^{-1}$, $t_{21} = 1 + z^{-1}$

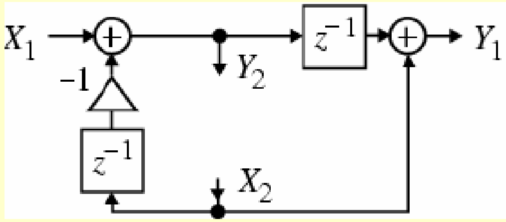
First-Order All-Pass Filter Structures

- From the transfer parameters of this allpass we arrive at the input-output relations:

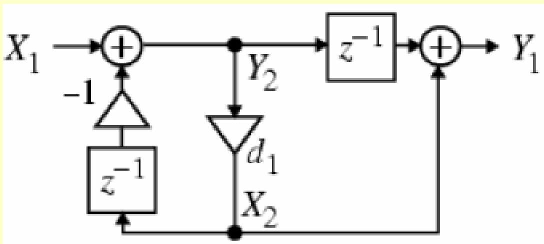
$$Y_2 = X_1 - z^{-1}X_2$$

$$Y_1 = z^{-1}X_1 + (1 - z^{2})X_2 = z^{-1}Y_2 + X_2$$

- A realization of the above two-pair is sketched below



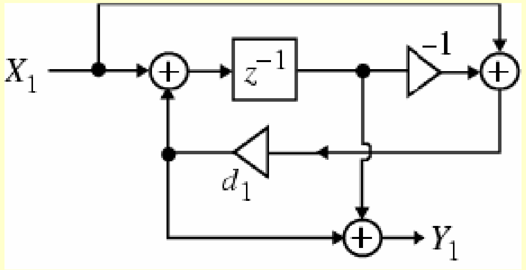
- By constraining the X_2, Y_1, terminal-pair with the multiplier d_1, we arrive at the Type 1A allpass filter structure shown below:



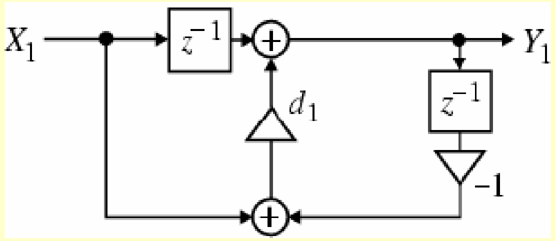
Type 1A

First-Order All-Pass Filter Structures

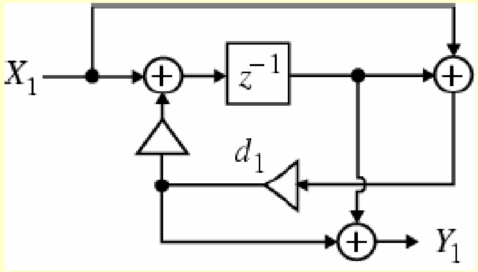
- In a similar fashion, the other three single multiplier first-order allpass filter structures can be developed as shown below



Type 1B



Type $1A_t$

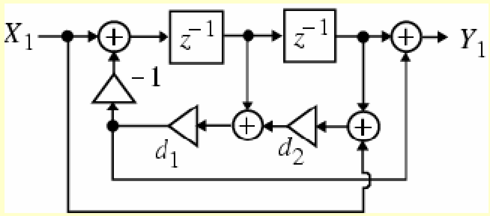


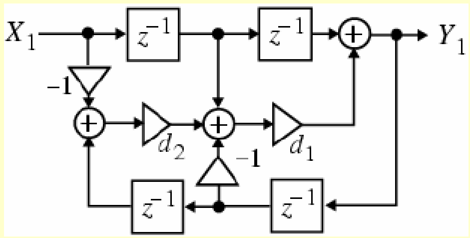
Type $1B_t$

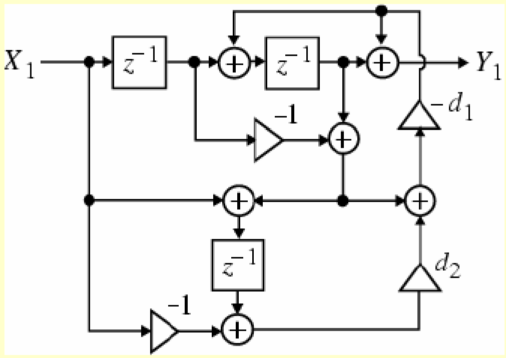
Second-Order All-Pass Structures

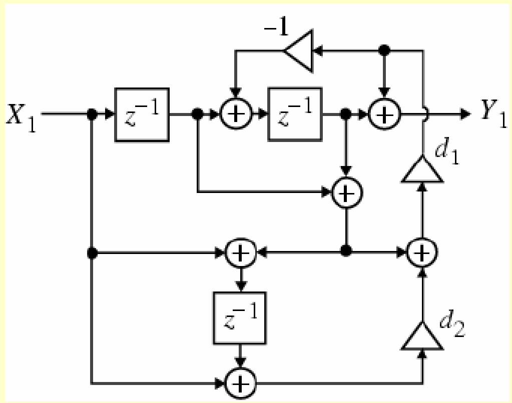
- A 2nd-order allpass transfer function is characterized by 2 unique coefficients
- Hence, it can be realized using only 2 multipliers
- Type 2 allpass transfer function:

$$A_2(z) = \frac{d_1 d_2 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_1 d_2 z^{-2}}$$





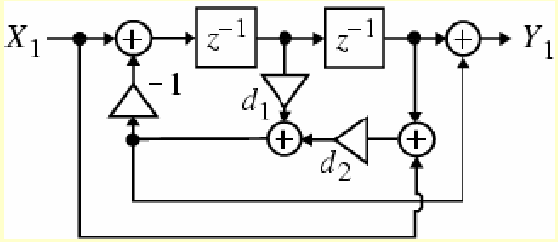


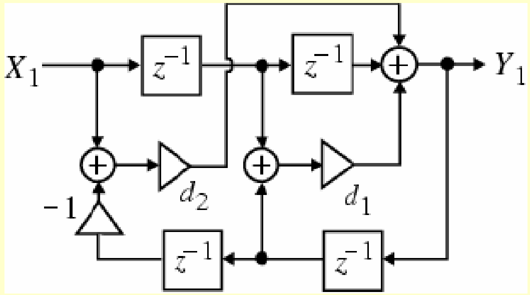


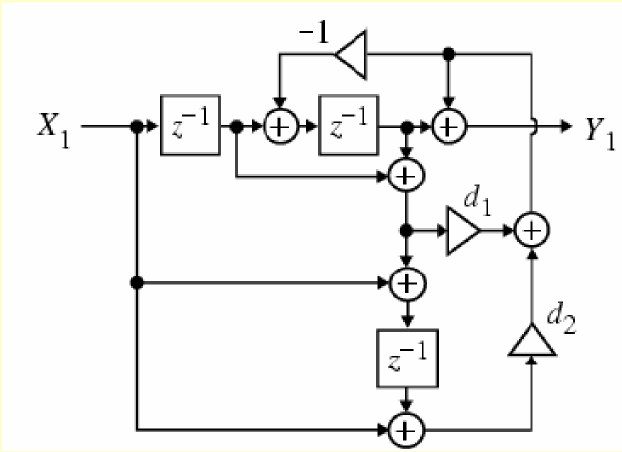
Type-3 All-Pass Structures

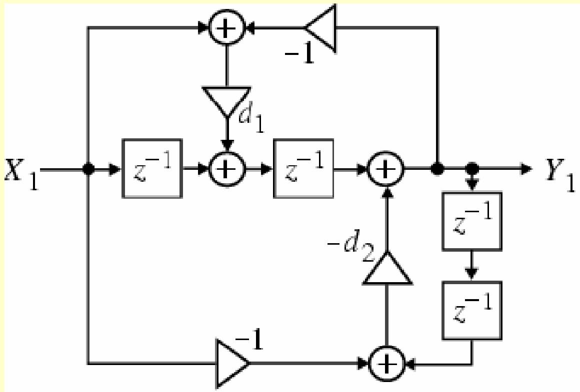
- Type 3 allpass transfer function:

$$A_3(z) = \frac{d_2 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_2 z^{-2}}$$









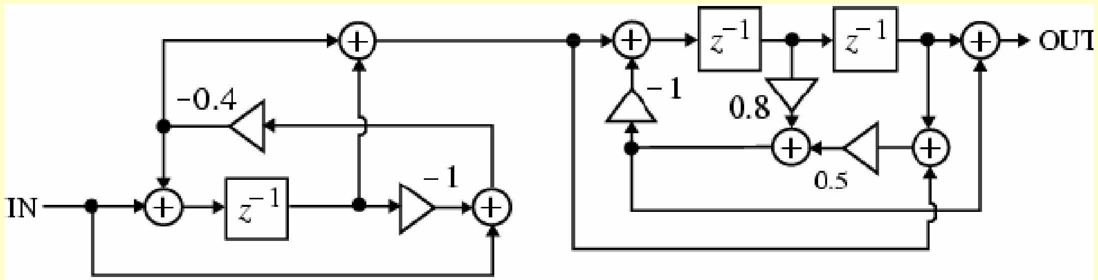
Realization Using Multiplier Extraction Approaches

- Example – Realize

$$A_3(z) = \frac{-0.2+0.18z^{-1}+0.4z^{-2}+z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

$$= \frac{(-0.4+z^{-1})(0.5+0.8z^{-1}+z^{-2})}{(1-0.4z^{-1})(1+0.8z^{-1}+0.5z^{-2})}$$

- A 3-multiplier cascade realization of the above allpass transfer function is shown below



Realization Using Two-Pair Extraction Approaches

- The algorithm is based on the development of a series of $(m-1)$th-order allpass transfer functions $A_{m-1}(z)$ from an mth-order allpass transfer function $A_m(z)$ for $m = M, M-1, \ldots, 1$
- Let
$$A_m(z) = \frac{d_m + d_{m-1}z^{-1} + d_{m-2}z^{-2} + \cdots + d_1 z^{-(m-1)} + z^{-m}}{1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{m-1} z^{-(m-1)} + d_m z^{-m}}$$
- We use the recursion
$$A_{m-1}(z) = z\left[\frac{A_m(z) - k_m}{1 - k_m A_m(z)}\right], \quad m = M, M-1, \ldots, 1$$

  where $k_m = A_m(\infty) = d_m$
- It has been shown earlier that $A_M(z)$ is stable if and only if
$$k_m^2 < 1 \quad \text{for } m = M, M-1, \ldots, 1$$

Realization Using Two-Pair Extraction Approaches

- If the allpass transfer function $A_{m-1}(z)$ is expressed in the form

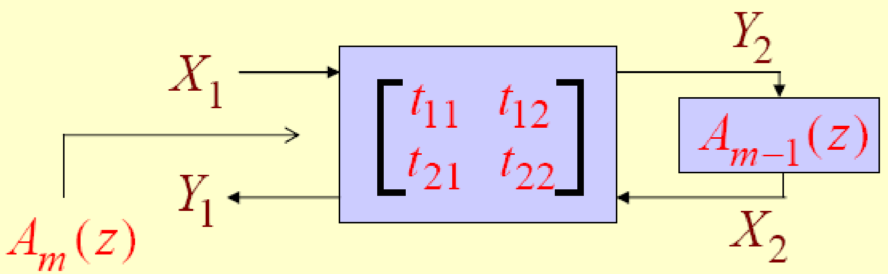
$$A_{m-1}(z) = \frac{d'_{m-1} + d'_{m-2}z^{-1} + \cdots + d'_1 z^{-(m-2)} + z^{-(m-1)}}{1 + d'_1 z^{-1} + \cdots + d'_{m-2} z^{-(m-2)} + d'_{m-1} z^{-(m-1)}}$$

- then the coefficients of $A_{m-1}(z)$ are simply related to the coefficients of $A_m(z)$ through

$$d'_i = \frac{d_i - d_m d_{m-i}}{1 - d_m^2}, \quad 1 \le i \le m-1$$

- To develop the realization method we express $A_m(z)$ in terms of $A_{m-1}(z)$

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



Realization Using Two-Pair Extraction Approaches

- The transfer function $A_m(z) = Y_1/X_1$ of the constrained two-pair can be expressed as

$$A_m(z) = \frac{t_{11} - (t_{11}t_{22} - t_{12}t_{21})A_{m-1}(z)}{1 - t_{22}A_{m-1}(z)}$$

- Comparing the above with

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

we arrive at the two-pair transfer parameters

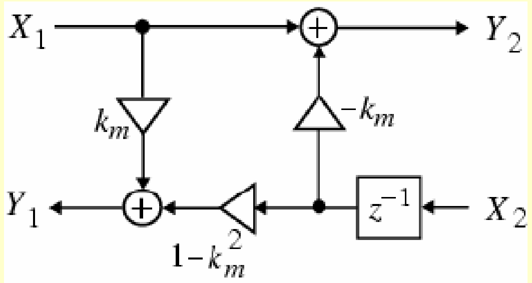
$$t_{11} = k_m,\ t_{22} = -k_m z^{-1},\ t_{12} = (1 - k_m^2)z^{-1},\ t_{21} = 1$$

- Corresponding input-output relations are

$$Y_1 = k_m X_1 + (1 - k_m^2)z^{-1}X_2$$
$$Y_2 = X_1 - k_m z^{-1}X_2$$

Realization Using Two-Pair Extraction Approaches

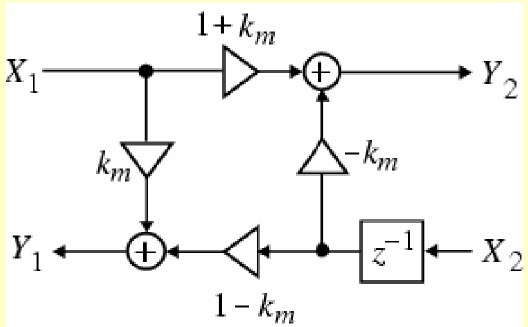
- A direct realization of the above equations leads to the following 3-multiplier two-pair



- The transfer parameters

$$t_{11} = k_m,\ t_{22} = -k_m z^{-1},\ t_{12} = (1-k_m)z^{-1},\ t_{21} = 1+k_m$$

lead to the 4-multiplier two-pair structure shown below

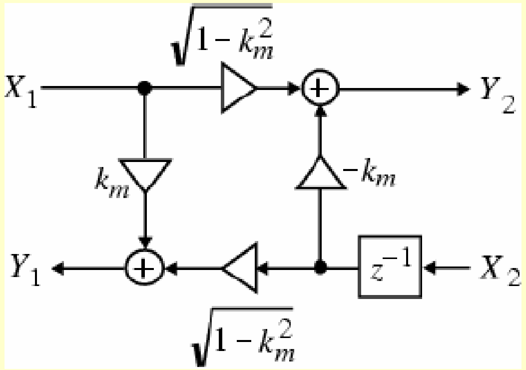


Realization Using Two-Pair Extraction Approaches

- Likewise, the transfer parameters

$$t_{11} = k_m,\ t_{22} = -k_m z^{-1},\ t_{12} = \sqrt{1-k_m^2}\, z^{-1},\ t_{21} = \sqrt{1-k_m^2}$$

lead to the 4-multiplier two-pair structure shown below



Realization Using Two-Pair Extraction Approaches

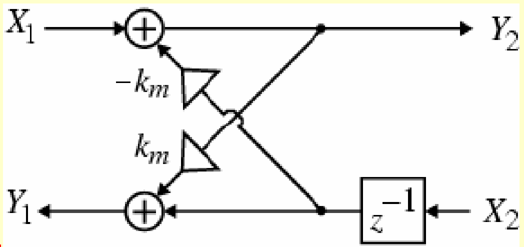
- A 2-multiplier realization can be derived by manipulating the input-output relations:

$$Y_1 = k_m X_1 + (1 - k_m^2) z^{-1} X_2$$
$$Y_2 = X_1 - k_m z^{-1} X_2$$

- Making use of the second equation, we can rewrite the equations as

$$Y_1 = k_m Y_2 + z^{-1} X_2$$
$$Y_2 = X_1 - k_m z^{-1} X_2$$

lead to the 2-multiplier two-pair structure, known as the **lattice structure**, shown below



Realization Using Two-Pair Extraction Approaches

- Consider the two-pair described by

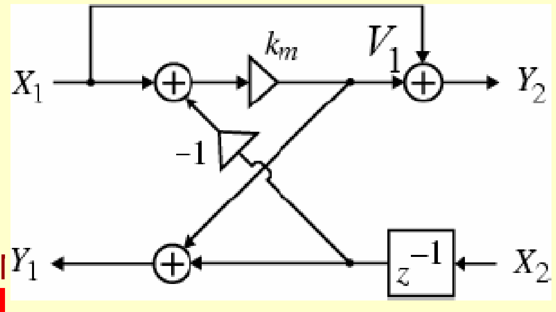
$$t_{11} = k_m,\ t_{22} = -k_m z^{-1},\ t_{12} = (1 - k_m)z^{-1},\ t_{21} = 1 + k_m$$

- Its input-output relations are given by

$$Y_1 = k_m X_1 + (1 - k_m)z^{-1}X_2$$
$$Y_2 = (1 + k_m)X_1 - k_m z^{-1}X_2$$

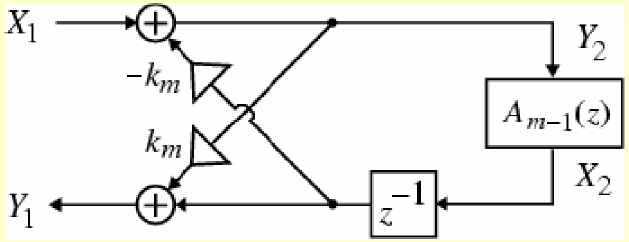
- Define $V_1 = k_m(X_1 - z^{-1}X_2)$
- We can then rewrite the input-output relations as

  $Y_1 = V_1 + z^{-1}X_2$ and $Y_2 = X_1 + V_1$, leading to the following 1-multimplier architecture

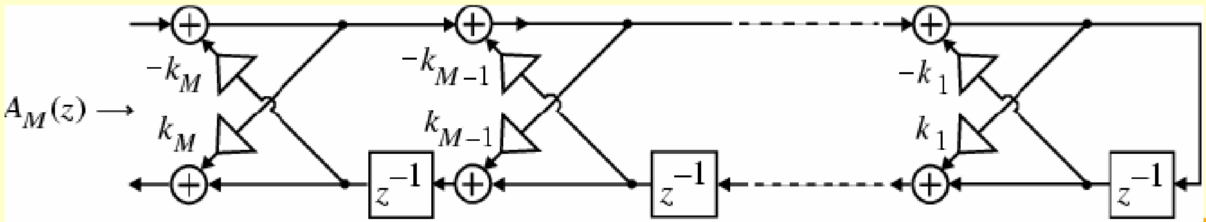


Realization Using Two-Pair Extraction Approaches

- An mth-order allpass transfer function $A_m(z)$ is then realized by constraining any one of the two-pairs of the (m-1)th-order allpass transfer function $A_{m-1}(z)$



- The process is repeated until the constraining transfer function is $A_0(z) = 1$
- The realization of $A_m(z)$ based on the extraction of the two-pair lattice is shown below



Realization Using Two-Pair Extraction Approaches

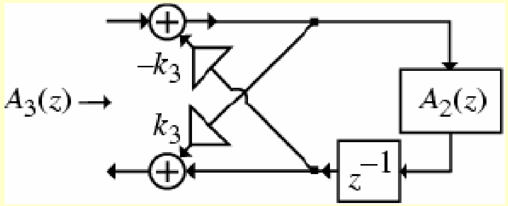
- It follows from our earlier discussion that $A_m(z)$ is stable if the magnitudes of all multiplier coefficients in the realization are less than 1, i.e., $|k_m| < 1$, for $m = M, M - 1, ..., 1$
- The cascaded lattice allpass filter structure requires $2M$ multipliers
- A realization with M multipliers is obtained if instead the single multiplier two-pair is used
- Example - Realize

$$A_3(z) = \frac{-0.2 + 0.18z^{-1} + 0.4z^{-2} + z^{-3}}{1 + 0.4z^{-1} + 0.18z^{-2} - 0.2z^{-3}}$$

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

Realization Using Two-Pair Extraction Approaches

- We first realize $A_3(z)$ in the form of a lattice two-pair characterized by the multiplier coefficient $k_3 = d_3 = -0.2$ and constrained by a 2nd-order allpass $A_2(z)$ as indicated below



$$k_3 = -0.2$$

- The allpass transfer function $A_2(z)$ is of the form

$$A_2(z) = \frac{d_2' + d_1' z^{-1} + z^{-2}}{1 + d_1' z^{-1} + d_2' z^{-2}}$$

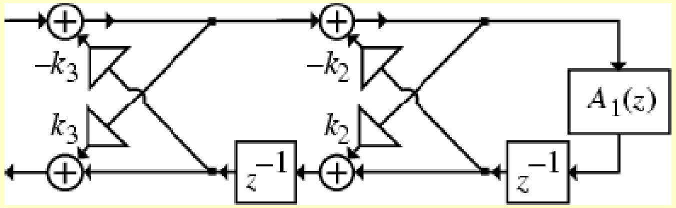
- Its coefficients are given by

$$d_1' = \frac{d_1 - d_3 d_2}{1 - d_3^2} = \frac{0.4 - (-0.2)(0.18)}{1 - (-0.2)^2} = 0.4541667$$

$$d_2' = \frac{d_2 - d_3 d_1}{1 - d_3^2} = \frac{0.18 - (-0.2)(0.4)}{1 - (-0.2)^2} = 0.2708333$$

Realization Using Two-Pair Extraction Approaches

- Next, the allpass $A_2(z)$ is realized as a lattice two-pair characterized by the multiplier coefficient $k_2 = d_2' = -0.2708$ and constrained by an allpass $A_1(z)$ as indicated below



$k_3 = -0.2, \quad k_2 = 0.2708333$

- The allpass transfer function $A_1(z)$ is of the form

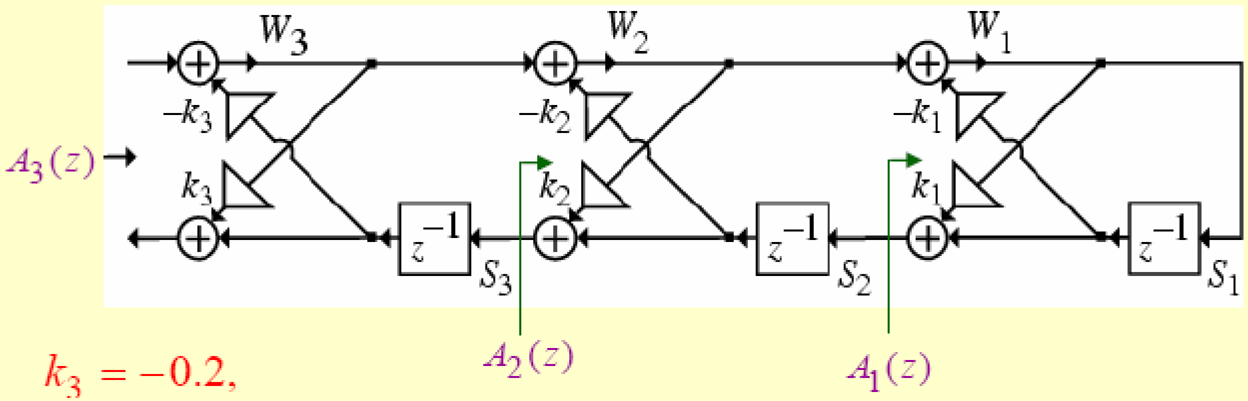
$$A_1(z) = \frac{d_1'' + z^{-1}}{1 + d_1'' z^{-1}}$$

- Its coefficients is given by

$$d_1'' = \frac{d_1' - d_2' d_1'}{1 - (d_2')^2} = \frac{d_1'}{1 + d_2'} = \frac{0.4541667}{1.2708333} = 0.3573771$$

Realization Using Two-Pair Extraction Approaches

- Finally, the allpass $A_1(z)$ is realized as a lattice two-pair characterized by the multiplier coefficient $k_1 = d_1'' = -0.3574$ and constrained by an allpass $A_0(z)$ as indicated below



$k_3 = -0.2,$

$k_2 = 0.2708333, \quad k_1 = 0.3573771$

Tunable Lowpass and Highpass Digital Filters

- We have shown earlier that the 1st-order lowpass transfer function

$$H_{LP}(z) = \frac{1-\alpha}{2}\left(\frac{1+z^{-1}}{1-\alpha z^{-1}}\right)$$

and the 1st-order highpass transfer function

$$H_{HP}(z) = \frac{1+\alpha}{2}\left(\frac{1-z^{-1}}{1-\alpha z^{-1}}\right)$$

are doubly-complementary pair

- Moreover, they can be expressed as

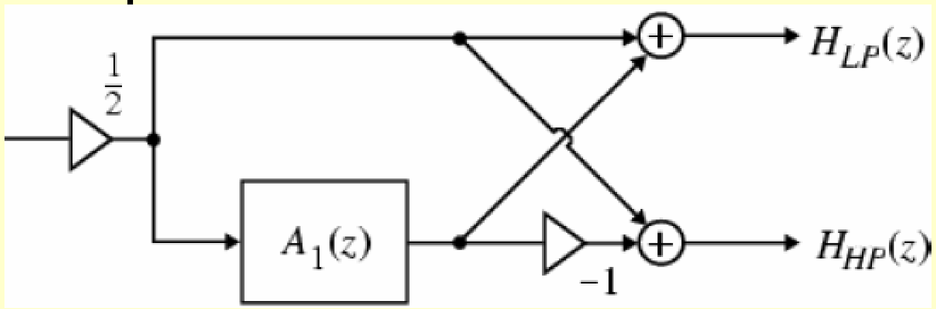
$$H_{LP}(z) = 1/2[1 + A_1(z)]$$

$$H_{HP}(z) = 1/2[1 - A_1(z)]$$

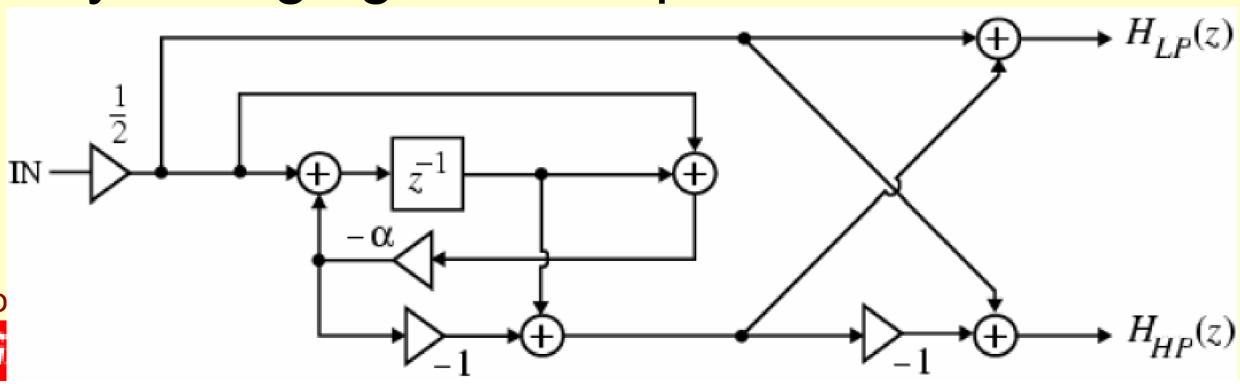
- where $A_1(z) = \dfrac{-\alpha + z^{-1}}{1-\alpha z^{-1}}$ is a 1st-order allpass transfer function

Tunable Lowpass and Highpass Digital Filters

- A realization of $H_{LP}(z)$ and $H_{HP}(z)$ based on the allpass-based decomposition is shown below

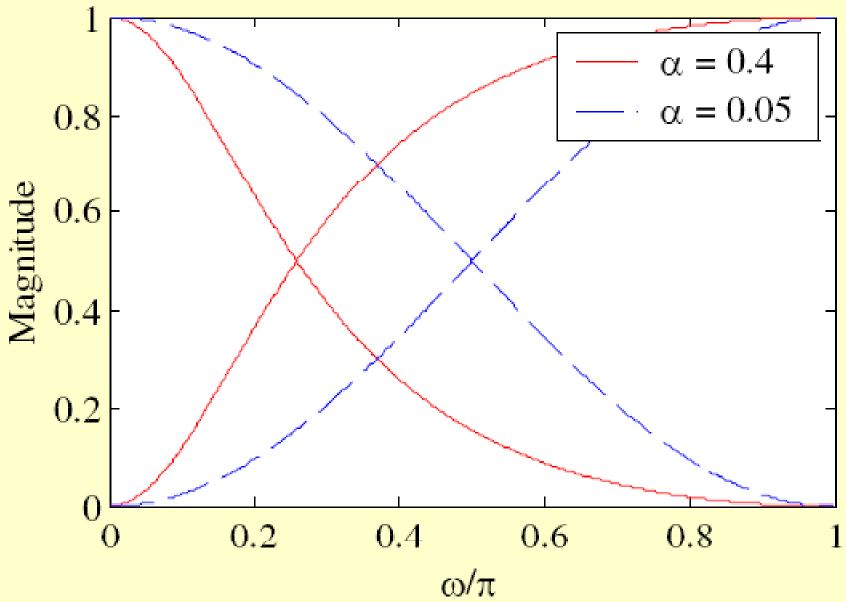


- The 1st-order allpass filter can be realized using any one of the 4 single-multiplier allpass structures
- In the following example, the 3-dB cutoff frequency can be varied by changing the multiplier coefficient α



Tunable Lowpass and Highpass Digital Filters

- Figure below shows the composite magnitude responses of the two filters for two different values of α



Tunable Bandpass and Bandstop Digital Filters

- The 2nd-order bandpass transfer function

$$H_{BP}(z) = \frac{1-\alpha}{2}\left(\frac{1-z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}}\right)$$

and the 2nd-order bandstop transfer function

$$H_{BS}(z) = \frac{1+\alpha}{2}\left(\frac{1-\beta z^{-1}+z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}}\right)$$

also form a doubly-complementary pair

- Thus, they can be expressed in the form

$$H_{BP}(z) = 1/2[1 - A_2(z)]$$

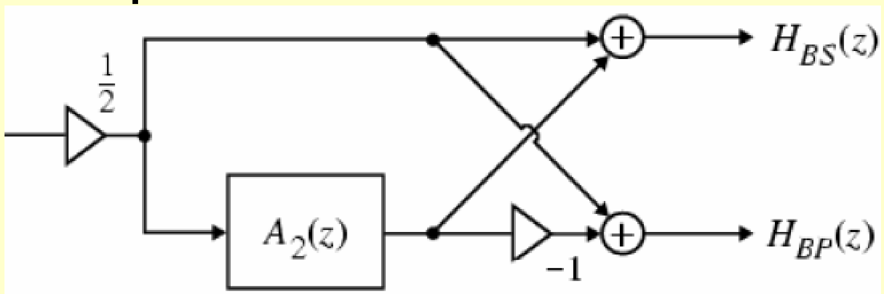
$$H_{BS}(z) = 1/2[1 + A_2(z)]$$

- where

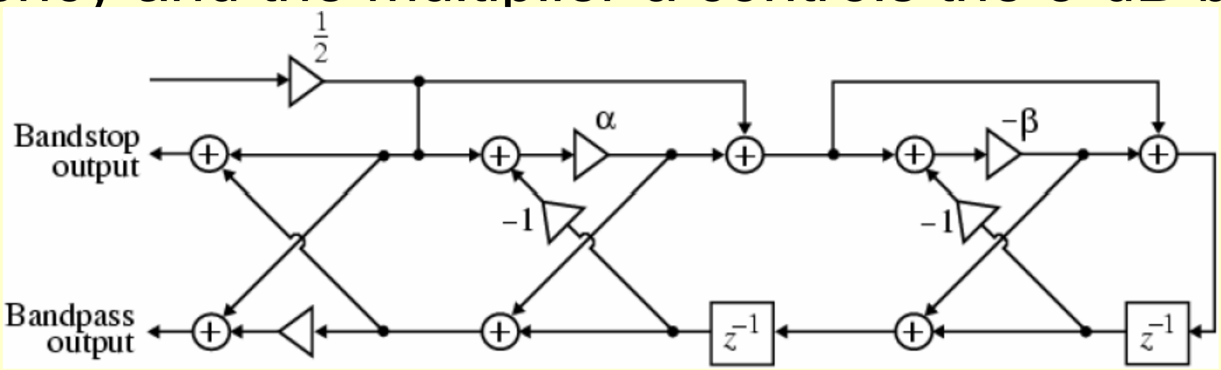
$$A_2(z) = \frac{\alpha-\beta(1+\alpha)z^{-1}+z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}}$$

Tunable Bandpass and Bandstop Digital Filters

- A realization of $H_{BP}(z)$ and $H_{BS}(z)$ based on the allpass-based decomposition is shown below

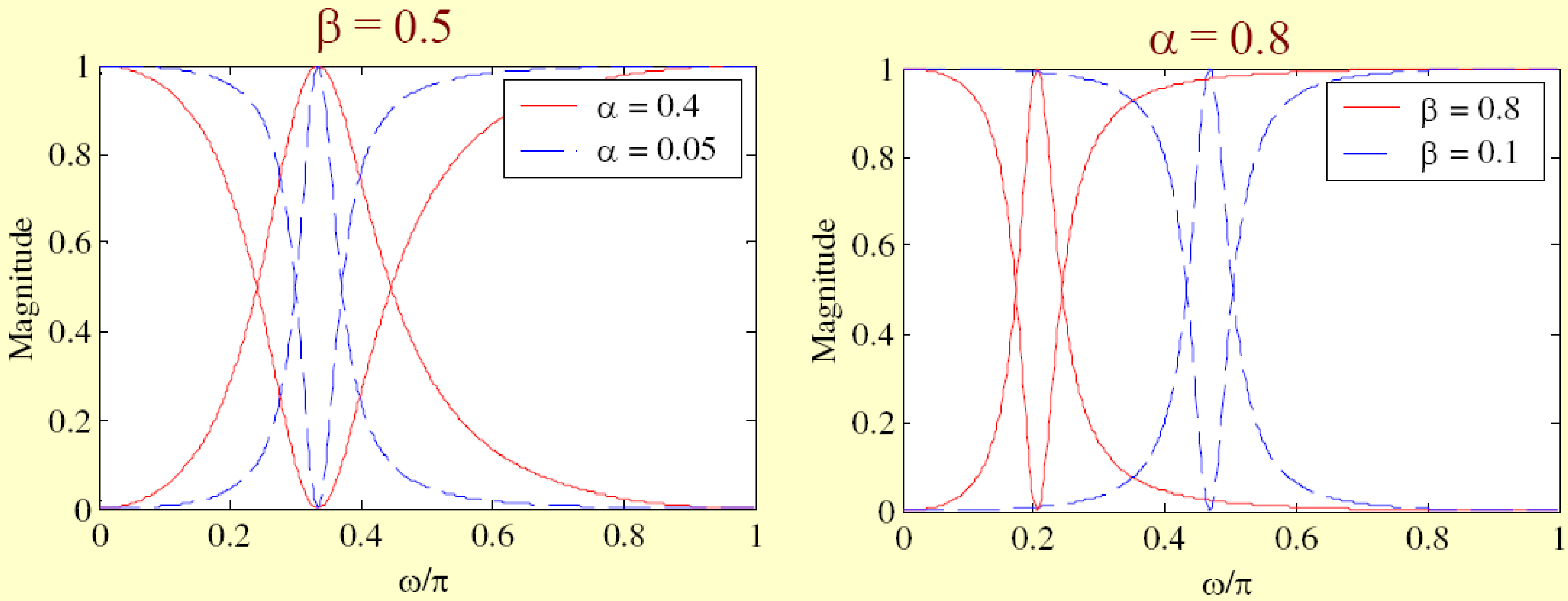


- The 2nd-order allpass filter is realized using a cascaded single-multiplier lattice structure
- In the following structure, the multiplier β controls the center frequency and the multiplier α controls the 3-dB bandwidth



Tunable Bandpass and Bandstop Digital Filters

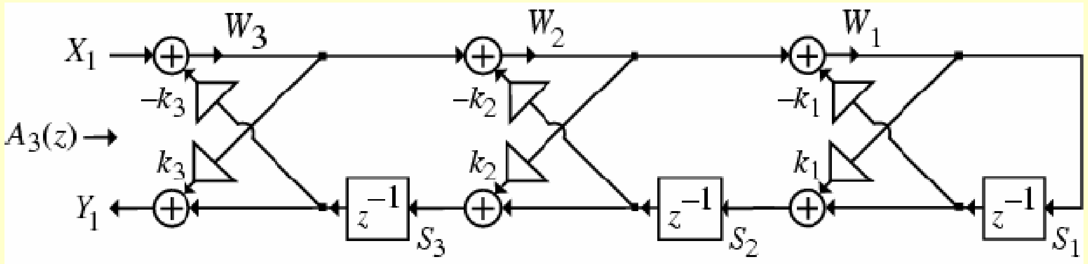
- Figure below illustrates the parametric tuning property of the overall structure



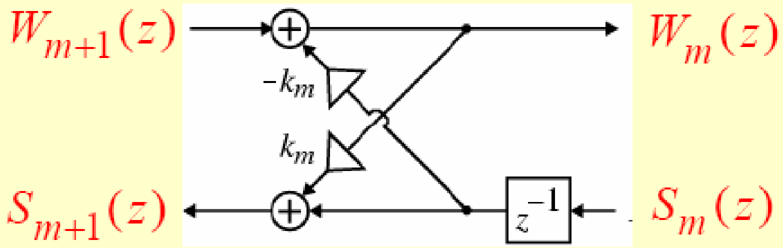
Original PowerPoint slides prepared by S. K. Mitra

IIR Tapped Cascaded Lattice Structures

- Consider the cascaded lattice structure derived earlier for the realization of an allpass transfer function



- A typical lattice two-pair here is as shown below



- Its input-output relations are given by

$$W_m(z) = W_{m+1}(z) - k_m z^{-1} S_m(z)$$

$$S_{m+1}(z) = k_m W_m(z) + z^{-1} S_m(z)$$

IIR Tapped Cascaded Lattice Structures

- From the input-output relations we derive the chain matrix description of the two-pair:

$$\begin{bmatrix} W_{m+1}(z) \\ S_{m+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & k_m z^{-1} \\ k_m & z^{-1} \end{bmatrix} \begin{bmatrix} W_m(z) \\ S_m(z) \end{bmatrix}$$

- The chain matrix description of the cascaded lattice structure is therefore

$$\begin{bmatrix} X_1(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} 1 & k_3 z^{-1} \\ k_3 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & k_2 z^{-1} \\ k_2 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & k_1 z^{-1} \\ k_1 & z^{-1} \end{bmatrix} \begin{bmatrix} W_1(z) \\ S_1(z) \end{bmatrix}$$

- From the above equation we arrive at

$$\begin{aligned} X_1(z) = \{1 &+ [k_1(1+k_2) + k_2 k_3] z^{-1} \\ &+ [k_2 + k_1 k_3 (1+k_2)] z^{-2} + k_3 z^{-3}\} W_1(z) \\ &= (1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}) W_1(z) \end{aligned}$$

IIR Tapped Cascaded Lattice Structures

- Using the relation $S_1(z) = W_1(z)$ and the relations

$$k_1 = d_1'', \; k_2 = d_2', \quad k_3 = d_3$$

- The transfer function $W_1(z)/X_1(z)$ is thus an all-pole function with the same denominator as that of the 3rd-order allpass function $A_3(z)$

$$\frac{W_1(z)}{X_1(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

IIR Tapped Cascaded Lattice Structures

Gray-Markel Method

- A two-step method to realize an *M*th-order arbitrary IIR transfer function

$$H(z) = P_M(z)/D_M(z)$$

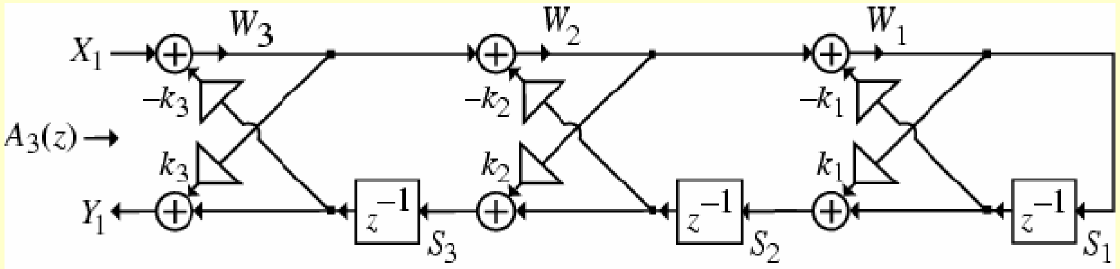
- Step 1: An intermediate allpass transfer function $A_m(z) = z^{-M} D_M(z^{-1})/D_M(z)$ is realized in the form of a cascaded lattice structure
- Step 2: A set of independent variables are summed with appropriate weights to yield the desired numerator $P_M(z)$

IIR Tapped Cascaded Lattice Structures

- To illustrate the method, consider the realization of a 3rd-order transfer function

$$H(z) = \frac{P_3(z)}{D_3(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

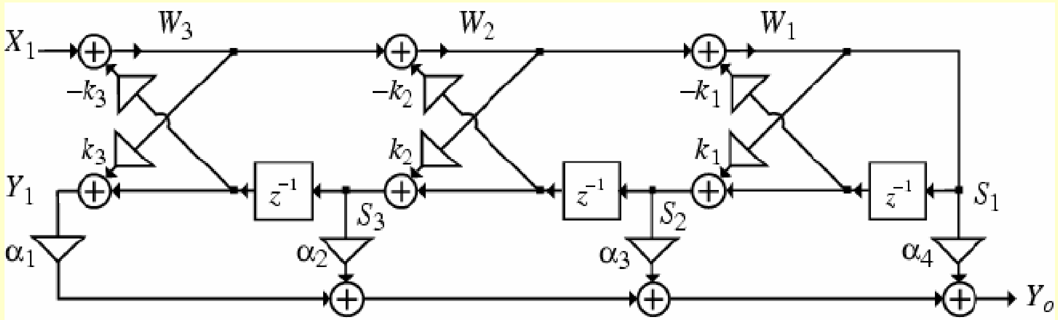
- In the first step, we form a 3rd-order allpass transfer function $A_3(z) = Y_1(z)/X_1(z) = z^{-3}D_3(z^{-1})/D_3(z)$
- Realization of $A_3(z)$ has been illustrated earlier resulting in the structure shown below



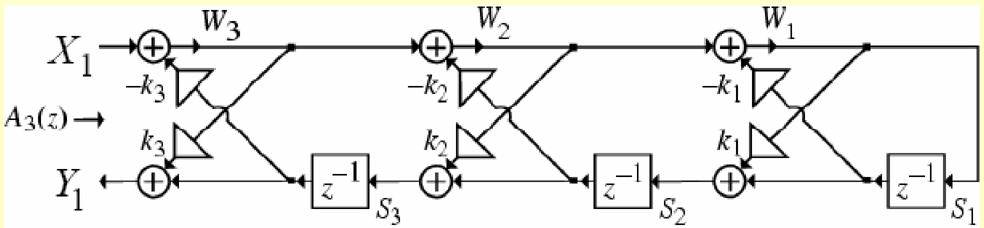
Original PowerPoint slides prepared by S. K. Mitra

IIR Tapped Cascaded Lattice Structures

- Objective: Sum the independent signal variables Y_1, S_1, S_2, and S_3 with weights $\{\alpha_i\}$ as shown below to realize the desired numerator $P_3(z)$



- To this end, we first analyze the cascaded lattice structure realizing and determine the transfer functions $S_1(z)/ X_1(z)$, $S_2(z)/ X_1(z)$, and $S_3(z)/ X_1(z)$



IIR Tapped Cascaded Lattice Structures

- We have already shown

$$\frac{S_1(z)}{X_1(z)} = \frac{1}{D_3(z)}$$

- From the figure it follows that

$$S_2(z) = (k_1 + z^{-1})S_1(z) = (d_1'' + z^{-1})S_1(z)$$

and hence

$$\frac{S_2(z)}{X_1(z)} = \frac{d_1'' + z^{-1}}{D_3(z)}$$

- The following relation are shown previously

$$S_2(z) = (d_1'' + z^{-1})S_1(z)$$
$$S_3(z) = d_2' W_2(z) + z^{-1}S_2(z)$$
$$S_1(z) = W_2(z) - d_1'' z^{-1}S_1(z)$$

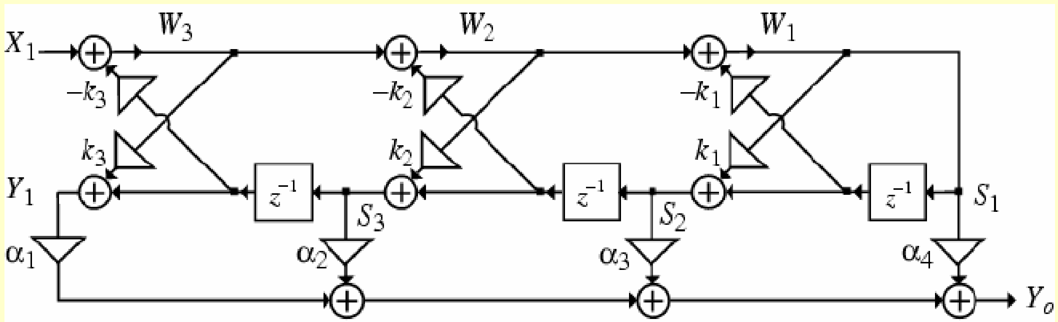
- From the last equation we get

$$W_2(z) = (1 + d_1'' z^{-1})S_1(z)$$

IIR Tapped Cascaded Lattice Structures

- Note: The numerator of is precisely the numerator of the allpass transfer function $A_i(z) = \dfrac{S_i(z)}{W_i(z)}$
- We now form

$$\frac{Y_o(z)}{X_1(z)} = \alpha_1 \frac{Y_1(z)}{X_1(z)} + \alpha_2 \frac{S_3(z)}{X_1(z)} + \alpha_3 \frac{S_2(z)}{X_1(z)} + \alpha_4 \frac{S_1(z)}{X_1(z)}$$



- Substituting the expressions for the various transfer functions in the above equation we arrive at

$$\frac{Y_o(z)}{X_1(z)} = \frac{\alpha_1(d_3 + d_2 z^{-1} + d_1 z^{-2} + z^{-3}) + \alpha_2(d_2' + d_1' z^{-1} + z^{-2}) + \alpha_3(d_1'' + z^{-1}) + \alpha_4}{D_3(z)}$$

IIR Tapped Cascaded Lattice Structures

- Comparing the numerator of $Y_0(z)/\, X_1(z)$ with the desired numerator $P_3(z)$ and equating like powers of z^{-1} we obtain

$$\alpha_1 d_3 + \alpha_2 d_2' + \alpha_3 d_1'' + \alpha_4 = p_0$$
$$\alpha_1 d_2 + \alpha_2 d_1' + \alpha_3 = p_1$$
$$\alpha_1 d_1 + \alpha_2 = p_2$$
$$\alpha_1 = p_3$$

- Solving the above equations we arrive at

$$\alpha_1 = p_3$$
$$\alpha_2 = p_2 - \alpha_1 d_1$$
$$\alpha_3 = p_1 - \alpha_1 d_2 - \alpha_2 d_1'$$
$$\alpha_4 = p_0 - \alpha_1 d_3 - \alpha_2 d_2' - \alpha_3 d_1''$$

IIR Tapped Cascaded Lattice Structures

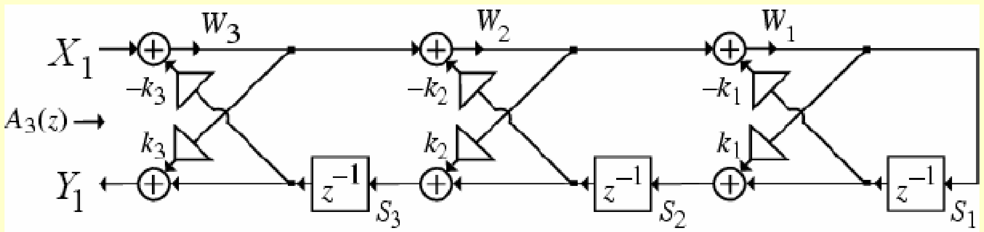
- Example - Consider

$$H(z)=\frac{P_3(z)}{D_3(z)}=\frac{0.44z^{-1}+0.362z^{-2}+0.02z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

- The corresponding intermediate allpass transfer function is given by

$$A_3(z)=\frac{z^{-3}D_3(z^{-1})}{D_3(z)}=\frac{-0.2+0.18z^{-1}+0.0.4z^{-2}+z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

- The allpass transfer function was realized earlier in the cascaded lattice form as shown below



- where $k_3 = d_3 = -0.2, \quad k_2 = d_2' = 0.2708333 \quad k_1 = d_1'' = 0.3573771$

IIR Tapped Cascaded Lattice Structures

- Other pertinent coefficients are:

$$d_1 = 0.4,\ d_2 = 0.18,\ d_3 = -0.2,\ d_1' = 0.4541667$$

$$p_0 = 0,\ p_1 = 0.44,\ p_2 = 0.36,\ p_3 = 0.02,$$

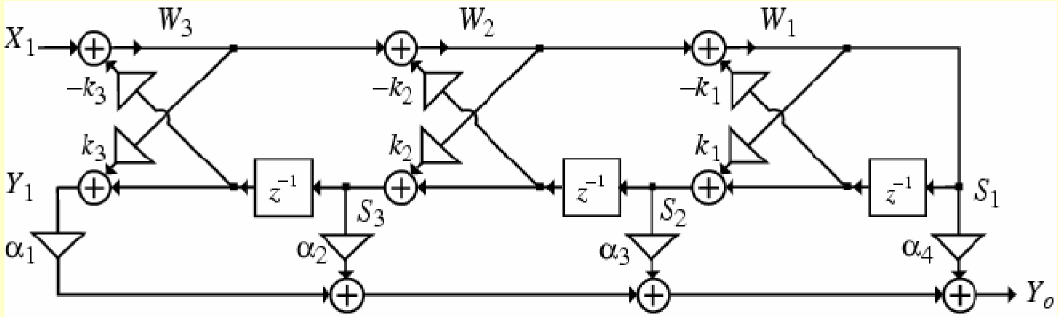
- Substituting these coefficients in

$$\alpha_1 = p_3 \qquad \alpha_3 = p_1 - \alpha_1 d_2 - \alpha_2 d_1'$$

$$\alpha_2 = p_2 - \alpha_1 d_1 \qquad \alpha_4 = p_0 - \alpha_1 d_3 - \alpha_2 d_2' - \alpha_3 d_1''$$

$$\alpha_1 = 0.02,\ \alpha_2 = 0.352 \qquad \alpha_3 = 0.2765333,\ \alpha_4 = -0.19016$$

- The final realization is as shown below



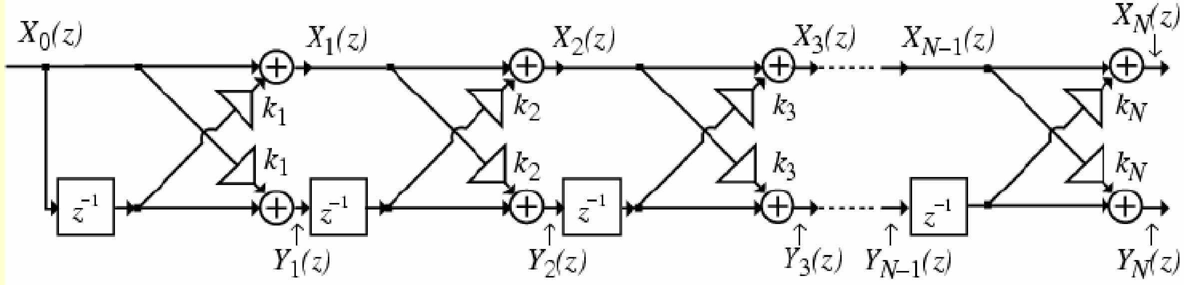
$$k_1 = 0.3573771,\ k_2 = 0.2708333,\ k_3 = -0.2$$

FIR Cascaded Lattice Structures

- An arbitrary *N*th-order FIR transfer function of the form

$$H_N(z) = 1 + \sum_{n=1}^{N} p_n z^{-n}$$

can be realized as a cascaded lattice structure as shown below



- From figure, it follows that

$$X_m(z) = X_{m-1}(z) + k_m z^{-1} Y_{m-1}(z)$$
$$Y_m(z) = k_m X_{m-1}(z) + z^{-1} Y_{m-1}(z)$$

FIR Cascaded Lattice Structures

- In matrix form the above equations can be written as

$$\begin{bmatrix} X_m(z) \\ Y_m(z) \end{bmatrix} = \begin{bmatrix} 1 & k_m z^{-1} \\ k_m & z^{-1} \end{bmatrix} \begin{bmatrix} X_{m-1}(z) \\ Y_{m-1}(z) \end{bmatrix}$$

where $m = 1,2,...,N$

- Denote

$$H_m(z) = \frac{X_m(z)}{X_0(z)}, \qquad G_m(z) = \frac{Y_m(z)}{X_0(z)}$$

- Then it follows from the input-output relations of the *m*-th two-pair that

$$H_m(z) = H_{m-1}(z) + k_m z^{-1} G_{m-1}(z)$$
$$G_m(z) = k_m H_{m-1}(z) + z^{-1} G_{m-1}(z)$$

FIR Cascaded Lattice Structures

- From the previous equation we observe

$$H_1(z) = 1 + k_1 z^{-1}, \; G_1(z) = k_1 + z^{-1}$$

where we have used the facts

$$H_0(z) = X_0(z)/\, X_0(z) = 1$$

$$G_0(z) = Y_0(z)/\, X_0(z) = X_0(z)/\, X_0(z) = 1$$

- It follows from the above that

$$G_1(z) = z^{-1}(z k_1 + 1) = z^{-1} H_1(z^{-1})$$

$\Rightarrow G_1(z)$ is the mirror-image of $H_1(z)$

FIR Cascaded Lattice Structures

- From the input-output relations of the *m*-th two-pair we obtain for $m = 2$:

$$H_2(z) = H_1(z) + k_2 z^{-1} G_1(z)$$
$$G_2(z) = k_2 H_1(z) + z^{-1} G_1(z)$$

- Since $H_1(z)$ and $G_1(z)$ are 1st-order polynomials, $H_2(z)$ and $G_2(z)$ are 2nd-order polynomials
- Substituting $G_1(z) = z^{-1} H_1(z^{-1})$ in the two above equations we get

$$H_2(z) = H_1(z) + k_2 z^{-2} H_1(z^{-1})$$
$$G_2(z) = k_2 H_1(z) + z^{-2} H_1(z^{-1})$$

- We can write $G_2(z) = z^{-2}[k_2 z^2 H_1(z) + H_1(z^{-1})] = z^{-2} H_2(z^{-1})$

$\Rightarrow G_2(z)$ is the mirror-image of $H_2(z)$

FIR Cascaded Lattice Structures

- In the general case, from the input-output relations of the *m*-th two-pair we obtain

$$H_m(z) = H_{m-1}(z) + k_m z^{-1} G_{m-1}(z)$$
$$G_m(z) = k_m H_{m-1}(z) + z^{-1} G_{m-1}(z)$$

- It can be easily shown by induction that
- Substituting $G_1(z) = z^{-1} H_1(z^{-1})$ in the two above equations we get

$$G_m(z) = z^{-m} H_m(z^{-1}),\ m = 1, 2, \ldots N-1, N$$

$\Rightarrow G_m(z)$ is the mirror-image of $H_m(z)$

FIR Cascaded Lattice Structures

- To develop the synthesis algorithm, we express $H_{m-1}(z)$ and $G_{m-1}(z)$ in terms of $H_m(z)$ and $G_m(z)$ for $m = N,\ N-1,...,2,1$ arriving at

$$H_{N-1}(z) = \frac{1}{(1-k_N^2)}\{H_N(z) - k_N G_N(z)\}$$

$$G_{N-1}(z) = \frac{1}{(1-k_N^2)z^{-1}}\{-k_N H_N(z) + G_N(z)\}$$

- Substituting the expressions for

$$H_N(z) = 1 + \sum_{n=1}^{N} p_n z^{-n}$$

and

$$G_N(z) = z^{-N} H_N(z^{-1}) = \sum_{n=0}^{N-1} p_{N-n} z^{-n} + z^{-N}$$

in the first equation we get

$$H_{N-1}(z) = \frac{1}{1-k_N^2}\{(1-k_N p_N) + \sum_{n=1}^{N-1}(p_n - k_n p_{N-n})z^{-n} + (p_N - k_N)z^{-N}\}$$

Original PowerPoint slides prepared by S. K. Mitra

FIR Cascaded Lattice Structures

- If we choose $k_N = p_N$, then $H_{N-1}(z)$ reduces to an FIR transfer function of order $N-1$ and can be written in the form

$$H_{N-1}(z) = 1 + \sum_{n=1}^{N-1} p'_n z^{-n}$$

- where $p'_n = \frac{p_n - k_N p_{N-n}}{1-k_N^2}, 1 \le n \le N-1$
- Continuing the above recursion algorithm, all multiplier coefficients of the cascaded lattice structure can be computed
- Example - Consider

$$H_4(z) = 1 + 1.2z^{-1} + 1.12z^{-2} + 0.12z^{-3} - 0.08z^{-4}$$

- From the above, we observe $k_4 = p_4 = -0.08$

FIR Cascaded Lattice Structures

- Using $$p'_n = \frac{p_n - k_4 p_{4-n}}{1-k_4^2}, 1 \le n \le 3$$

  we determine the coefficients of $H_3(z)$

  p_3'= 0.2173913, p_2'= 1.2173913, p_1'= 1.2173913
- As a result

  $H_3(z) = 1 + 1.2173913z^{-1} + 1.2173913z^{-2} + 0.2173913z^{-3}$
- Thus, $k_3 = p_3$'= 0.2173913
- Using $$p''_n = \frac{p'_n - k_3 p'_{2-n}}{1-k_3^2}, 1 \le n \le 2$$

  we determine the coefficients of $H_2(z)$

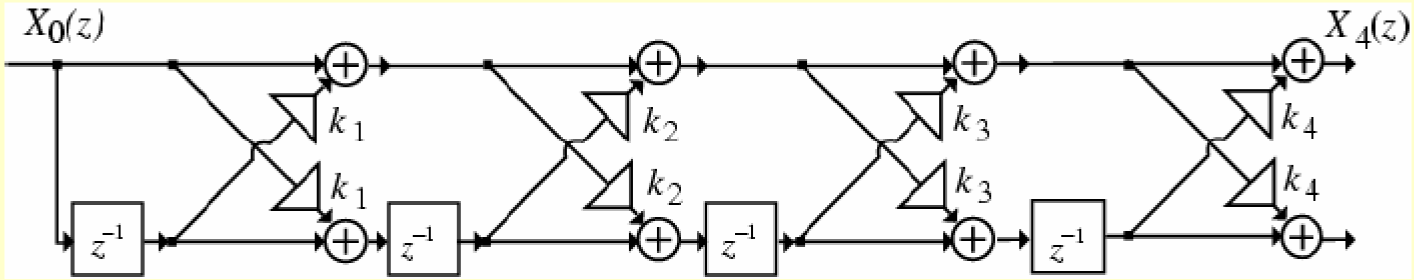
  p_2"= 1.0, p_1"= 1.0

FIR Cascaded Lattice Structures

- As a result, $H_2(z) = 1 + z^{-1} + z^{-2}$
- From the above, we get $k_2 = p_2'' = 1$
- The final recursion yields the last multiplier coefficient

$$k_2 = p_2''/(1+ k_2) = 0.5$$

- The complete realization is shown below



$$k_1 = 0.5,\ k_2 = 1,\ k_3 = 0.2173913,\ k_4 = -0.08$$