8 Digital Filter Structures

Introduction

- Input-output relation of an LTI system can be realized using different computational algorithms
- Basic realization forms of FIR and IIR digital filters are considered
- Mitra's book covers also various more sophisticated realizations of digital filters, e.g. lattice structures, allpass sections, and state space structures, not discussed in this course

Time-Domain Characterizations

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

Linear Constant Coefficient Difference Equation:

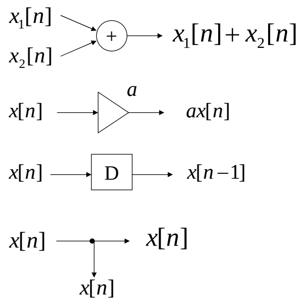
$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

State-Space Equations:

$$\mathbf{s}[n+1] = \mathbf{A}\mathbf{s}[n] + \mathbf{B}x[n]$$
$$y[n] = \mathbf{C}\mathbf{s}[n] + dx[n]$$

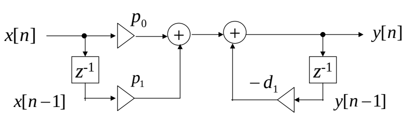
Basic Building Blocks

- Adder:
- Multiplier:
- Unit delay:
- Branch node:



Basic Operations

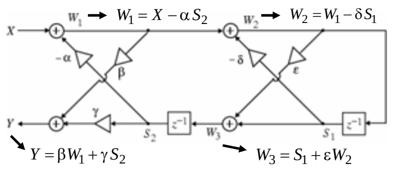
- Addition / Subtraction
- Multiplication (constant coefficient)
- Delay (memory)
- Example: First-order digital filter



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

Analysis of Block Diagrams

- <u>Example:</u> Analyze the cascaded lattice structure shown below where the *z*-dependence of signal variables are not shown for brevity



- The signal values in the delay outputs are:

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

Analysis of Block Diagrams

- Substituting the values of delay elements in the first four equations we get

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

- Solving W_2 from the second equation we get

$$W_2 = W_1 / (1 + \delta z^{-1})$$

and solving W_3 from the third equation we get

$$W_3 = (\varepsilon + z^{-1}) W_2$$

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 the first and fourth equation gives

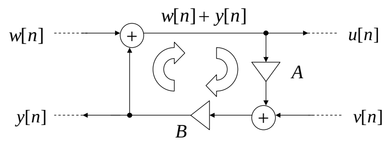
$$W_1 = X - \alpha z^{-1} W_3, \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

we finally arrive at

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

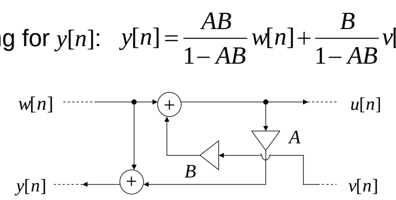
The Delay-Free Loop Problem

- A block diagram containing delay-free loops is physically non-realizable
- Example: $y[n] = B\{A(w[n] + y[n]) + v[n]\}$



The Delay-Free Loop Problem

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



Delay-free loop realization

Equivalent Structures

- Two digital filter structures are defined to be *equivalent* if they have the same transfer function
- Generation of an equivalent structure via the *transpose operation*:
  1) Reverse all paths,
  2) Replace pick-off (branching) nodes by adders, and vice versa,
  3) Interchange the input and output nodes

The original structure and the transposed structure have the same transfer function

Basic FIR Digital Filter Structures

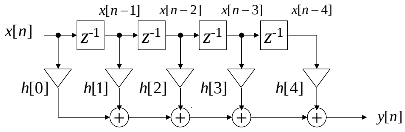
- Transfer function of a causal FIR filter of length M:

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

$H(z)$ is a polynomial in z^{-1} of degree M-1

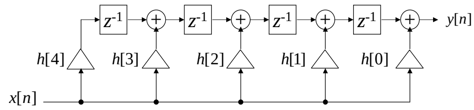
- Input-output relation is given by: $y[n] = \sum_{k=0}^{M-1} h[k] x[n-k]$
- The output $y[n]$ is the weighted sum of the input $x[n]$ and and its M-1previous values
- The weights are the the values of the unit impulse response $h[n]$

Direct Form FIR Filter Structure



- The products $h[k]x[n\text{-}k]$ are accumulated to form the output $y[n]$
- The structure is called a *tapped delay line* or a *transversal filter*

Transposed Direct Form FIR Filter Structure



- Both direct form structures are canonic with respect to delays
- Direct form FIR structures are computationally efficient when using modern signal processors

Polyphase Realization

- *Polyphase decomposition* of the FIR transfer function results in a parallel structure of an FIR filter

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

- Expressing the above equation as a sum of two terms, one containing the even-indexed coefficients and the other containing the odd-indexed coefficients

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

Polyphase Realization

- Using the notations

$$E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4}$$
$$E_1(z) = h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3}$$

$H(z)$ can be written as: $\boxed{H(z) = E_0(z^2) + z^{-1}E_1(z^2)}$

- Similarly, by grouping the terms differently, the transfer function can be rewritten as

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

where

$$E_0(z) = h[0] + h[3]z^{-1} + h[6]z^{-2}$$
$$E_1(z) = h[1] + h[4]z^{-1} + h[7]z^{-2}$$
$$E_2(z) = h[3] + h[5]z^{-1} + h[8]z^{-2}$$

Polyphase Decomposition

- In general, an L-branch polyphase decomposition of the transfer function $H(z)$ of order M-1 is of the form

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

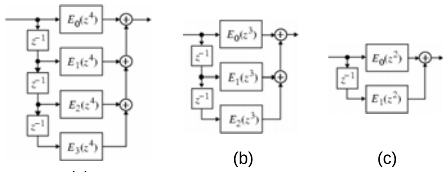
where $E_m(z) = \sum_{n=0}^{\lfloor M/L \rfloor} h[Ln+m]\ z^{-n}, \quad 0 \le m \le L-1$

with $h[n] = 0$, for $n \ge M$

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

Polyphase Realization

- A realization of the transfer function $H(z)$ based on the polyphase decomposition is called a *polyphase realization*



(a) (b) (c)

Polyphase realizations of an FIR transfer function: Four-branch (a), three-branch (b), and two-branch (c) structures

Linear-Phase FIR Structures

- Linear-phase FIR filter of length M is characterized by the symmetric impulse response

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

- An antisymmetric impulse response condition

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

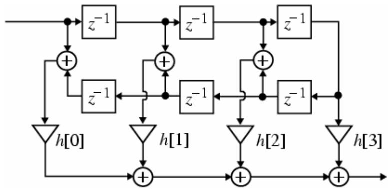
results in a constant group delay and "almost linear-phase" property

Symmetry of the impulse response coefficients can be used to reduce the number of multiplications

Linear-Phase FIR Structures

- Length M is odd (M=7)

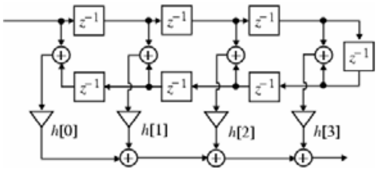
$$H(z) = h[0]\left(1+z^{-6}\right) + h[1]\left(z^{-1}+z^{-5}\right) + h[2]\left(z^{-2}+z^{-4}\right) + h[3]z^{-3}$$



Linear-Phase FIR Structures

- Length M is even (M=8)

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



Basic IIR Filter Structures

- The transfer function is rational
- <u>Direct forms:</u> Coefficients are directly the transfer function coefficients

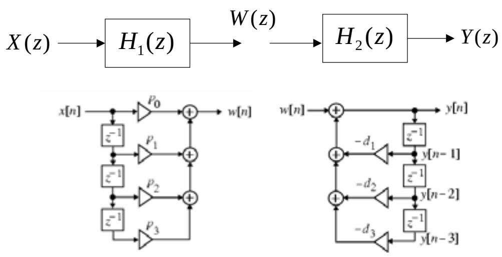
$$H(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + p_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

- Considering the numerator and denominator separately

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

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

Basic IIR Filter Structures

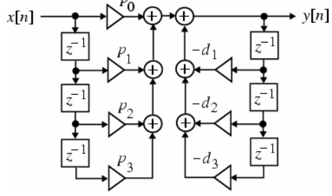


- $H_1(z)$ realizes the zeros and $H_2(z)$ realizes the poles of the transfer function $H(z)$

Direct Form I

- Considering the basic cascade realization results in ***Direct form I*** :

$$H(z) = P(z) \cdot \frac{1}{D(z)}$$

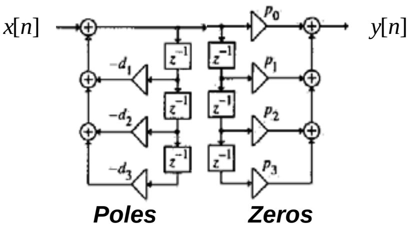


Zeros ***Poles***

Direct Form II

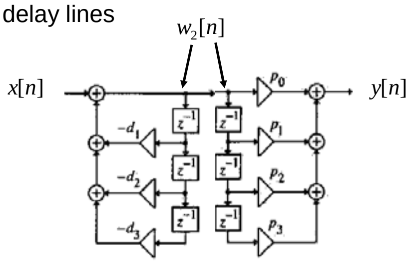
- Changing the order of blocks in cascade results in ***direct form II***

$$H(z) = P(z) \cdot \frac{1}{D(z)} = \frac{1}{D(z)} \cdot P(z)$$

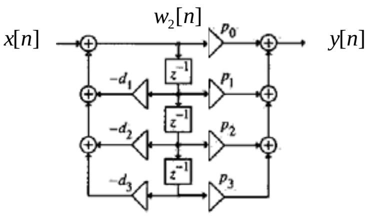


Canonic Structure

- The number of delays can be reduced by noticing that the same signal value $w_2[n]$ is stored into both delay lines

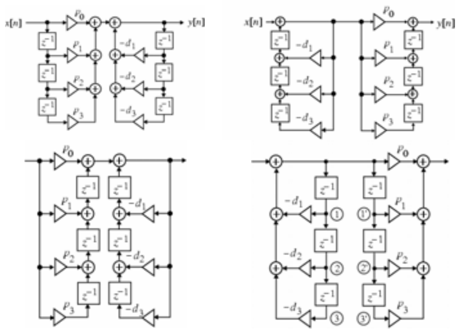


Canonic Direct Form II Structure

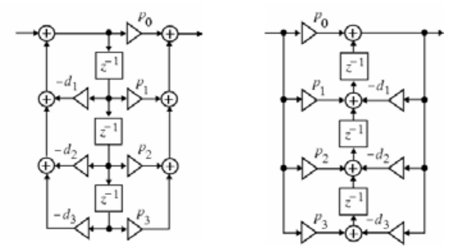


⇨ ***Canonic structure with respect to delays***

Additional Direct Form I Structures



Direct Form II Structures



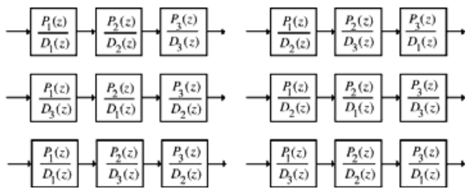
Direct Form II and Direct Form II transposed

Cascade Realizations

- Factoring the numerator and denominator

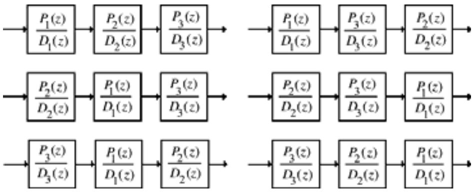
$$H(z) = \frac{P_1(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)}$$

- Various alternatives in pairing the poles and zeros



Cascade Realizations

- Various alternatives in ordering the sections



- Different realizations behave differently under finite wordlength constraints

First and Second Order Blocks in Cascade

- Usually the polynomials are factored into a product of first and second order polynomials

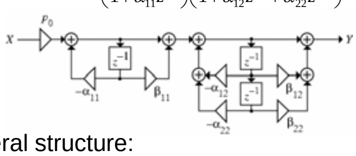
$$H(z)=\prod_k\left(\frac{1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}\right)$$

- For a first-order section $\alpha_{2k}=\beta_{2k}=0$
- Realizing complex conjugate poles and zeros with second order blocks results in real coefficients

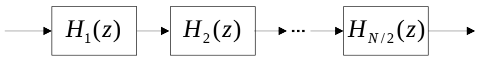
First and Second Order Blocks in Cascade

- Example: Third order transfer function

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



- General structure:

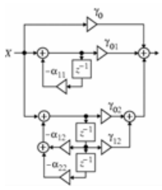


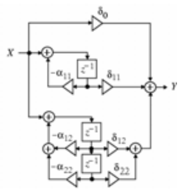
Parallel Realizations

- Parallel realizations are obtained by making use of the partial fraction expansion of the transfer function

$$H(z)=\gamma_0+\sum_k\left(\frac{\gamma_{0k}+\gamma_{1k}z^{-1}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}\right),\quad H(z)=\delta_0+\sum_k\left(\frac{\delta_{1k}z^{-1}+\delta_{2k}z^{-2}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}\right)$$

Parallel form I

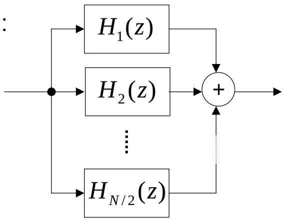




Parallel form II

Parallel Realizations

- General structure:



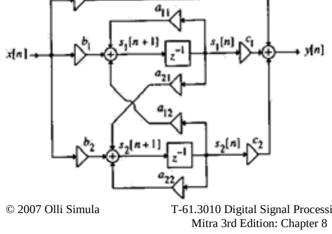
- Easy to realize:
  - No choices in section ordering and
  - No choices in pole and zero pairing

State-Space Structures

- A second-order IIR digital filter can be described by the state-space equations:

$$\begin{bmatrix}s_1[n+1]\\ s_2[n+1]\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}s_1[n]\\ s_2[n]\end{bmatrix}+\begin{bmatrix}b_1\\ b_2\end{bmatrix}x[n]$$

$$y[n]=\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}s_1[n]\\ s_2[n]\end{bmatrix}+dx[n]$$



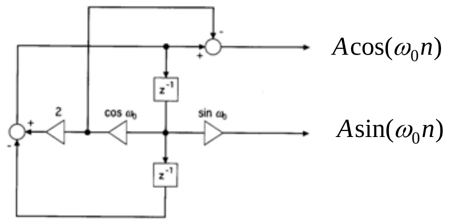
Large number of arithmetic operations needed (when compared to direct form second order blocks)

Digital Oscillators

- There are applications where a digital oscillator or frequency synthesizer is required to generate a discrete-time sinusoid of programmable frequency ω_0
- A second-order recursive digital filter with poles on the unit circle is “marginally stable”
- With non-zero initial conditions, it ideally produces a sinusoidal output
- The frequency ω_0 of the sinusoid is determined by the angle of the unit-circle poles

Recursive Quadrature Oscillators

- A quadrature oscillator generates two sinusoidal outputs of the same frequency and amplitude but the phase differs by 90º



Digital Sine-Cosine Generator

- Consider two causal impulse responses

$$h_1[n] = A\cos(\omega_0 n)\mu[n]$$

$$h_2[n] = A\sin(\omega_0 n)\mu[n]$$

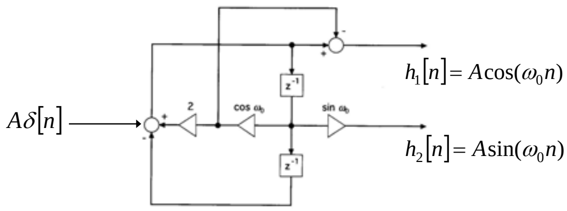
- The corresponding system functions (without gain A) are

$$H_1(z) = \frac{1-(\cos\omega_0)z^{-1}}{1-2(\cos\omega_0)z^{-1}+z^{-2}}$$

$$H_2(z) = \frac{(\sin\omega_0)z^{-1}}{1-2(\cos\omega_0)z^{-1}+z^{-2}}$$

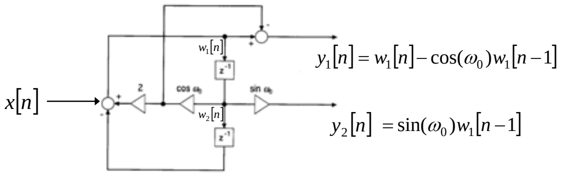
Digital Sine-Cosine Generator

- A two-output recursive structure below has the system functions $H_1(z)$ and $H_2(z)$



Digital Sine-Cosine Generator

- Solving the signal values from the structure



$$\begin{cases} w_2[n] = w_1[n-1] \\ w_1[n] = 2\cos(\omega_0)w_1[n-1] - w_2[n-1] + x[n] \end{cases}$$

$$\Rightarrow \quad w_1[n] = 2\cos(\omega_0)w_1[n-1] - w_1[n-2] + x[n]$$

Digital Sine-Cosine Generator

- Using the z-transform and solving $W_1(z)$

$$W_1(z) = 2\cos(\omega_0)z^{-1}W_1(z) - z^{-2}W_1(z) + X(z)$$

$$\Rightarrow \quad W_1(z)\left[1-2\cos(\omega_0)z^{-1}+z^{-2}\right] = X(z)$$

- Taking the z-transform of the outputs gives

$$Y_1(z) = W_1(z) - \cos(\omega_0)z^{-1}W_1(z)$$

$$Y_2(z) = \sin(\omega_0)z^{-1}W_1(z)$$

- Substituting $W_1(z)$ into the above equations

Digital Sine-Cosine Generator

- Equations for $Y_1(z)$ and $Y_2(z)$ are now

$$Y_1(z) = W_1(z)\left[1-\cos(\omega_0)z^{-1}\right] = \frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}X(z)$$

$$Y_2(z) = \sin(\omega_0)z^{-1}W_1(z) = \frac{\sin(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}X(z)$$

- Solving the system functions $H_1(z)$ and $H_2(z)$ gives

$$H_1(z) = \frac{Y_1(z)}{X(z)} = \frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$$

$$H_2(z) = \frac{Y_2(z)}{X(z)} = \frac{\sin(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$$

Digital Sine-Cosine Generator

- Substituting now $x[n]=A\delta[n]$ we notice that $X(z)=A$
- The expressions for the outputs $Y_1(z)$ and $Y_2(z)$ are now

$$Y_1(z) = AH_1(z) = A\frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$$

$$Y_2(z) = AH_2(z) = A\frac{\sin(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$$

- The oscillator outputs are obtained, e.g., from the inverse z-transform tables