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: \( y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \)
- State-Space Equations: \( \begin{align*}
    s[n+1] &= As[n] + Bs[n] \\
    y[n] &= Cs[n] + ds[n]
\end{align*} \)

Basic Operations

- Addition / Subtraction
- Multiplication (constant coefficient)
- Delay (memory)
- Example: First-order digital filter
  \( y[n] = p_2 x[n] + p_1 x[n-1] - d_1 y[n-1] \)

Basic Building Blocks

- Adder: \( x_1[n] + x_2[n] \)
- Multiplier: \( a[n] x[n] \)
- Unit delay: \( x[n] \rightarrow D \rightarrow x[n-1] \)
- Branch node: \( x[n] \rightarrow X[n] \)

Analysis of Block Diagrams

- Example: Analyze the cascaded lattice structure shown below where the z-dependence of signal variables are not shown for brevity
  \( \begin{align*}
    W_2 &= X - \alpha_1 S_2 \\
    W_1 &= W_2 - \beta_1 S_1 \\
    S_1 &= z^{-1} W_2 \\
    S_2 &= z^{-1} W_3
  \end{align*} \)
Analysis of Block Diagrams

- Substituting the values of delay elements in the first four equations we get
  \[ W_1 = X - \alpha z^{-1}W_3 \]
  \[ W_2 = W_1 - \beta z^{-1}W_2 \]
  \[ Y = \gamma W_1 + \gamma z^{-1}W_3 \]

- Solving \( W_2 \) from the second equation we get
  \[ W_2 = W_1 / (1 + \beta z^{-1}) \]

- Solving \( W_3 \) from the third equation we get
  \[ W_3 = (\epsilon + z^{-1})W_2 + \gamma \]

The Delay-Free Loop Problem

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

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

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 its \( M \)-previous values

- The weights are the values of the unit impulse response \( h[n] \)
Direct Form FIR Filter Structure

- The products $h[k]x[n-k]$ 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:
  
  \[ H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} \]

- Expressing the above equation as a sum of two terms, one containing the even-indexed coefficients and the other containing the odd-indexed coefficients:
  
  \[ H(z) = (h(0) + h(2)z^{-1} + h(4)z^{-2} + h(6)z^{-3} + h(8)z^{-4}) + z^{-1}(h(1) + h(3)z^{-1} + h(5)z^{-2} + h(7)z^{-3}) \]

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

Polyphase Decomposition

- In general, an $L$-branch polyphase decomposition of the transfer function $H(z)$ of order $M-1$ is of the form:
  
  \[ H(z) = \sum_{m=0}^{L-1} z^{-m}E_m(z^L) \]

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

Polyphase Realization

- Using the notations:
  
  \[ E_0(z) = h(0) + h(2)z^{-1} + h(4)z^{-2} + h(6)z^{-3} + h(8)z^{-4} \]
  \[ E_1(z) = h(1) + h(3)z^{-1} + h(5)z^{-2} + h(7)z^{-3} \]

- The transfer function $H(z)$ can be written as:
  
  \[ H(z) = z^{-L}E_0(z^L) + z^{-2L}E_1(z^L) \]

- Similarly, by grouping the terms differently, the transfer function can be rewritten as:
  
  \[ H(z) = E_0(z^L) + z^{-L}E_1(z^L) \]

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

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

Linear-Phase FIR Structures

- Length $M$ is even ($M=8$)
  
  $$ H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} $$

Basic IIR Filter Structures

- The transfer function is rational
  
  $ \frac{P(z)}{D(z)} $ where $P(z)$ and $D(z)$ are polynomials in $z^{-1}$

- Direct forms: Coefficients are directly the transfer function coefficients
  
  $ H(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1z^{-1} + p_2z^{-2} + p_3z^{-3}}{1 + d_1z^{-1} + d_2z^{-2} + d_3z^{-3}} $

- Considering the numerator and denominator separately
  
  $ H_1(z) = \frac{W(z)}{X(z)} = \frac{P(z)}{1 + d_1z^{-1} + d_2z^{-2} + d_3z^{-3}} $ and $ H_2(z) = \frac{Y(z)}{D(z)} = \frac{1}{1 + d_1z^{-1} + d_2z^{-2} + d_3z^{-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) = \frac{P(z)}{D(z)} = \frac{1}{D(z)} \cdot P(z) $
Direct Form II

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

\[ H(z) = P(z) \frac{1}{D(z)} = \frac{1}{D(z)} P(z) \]

Canonic Structure

- The number of delays can be reduced by noticing that the same signal value \( w_1[n] \) is stored into both delay lines

Canonic Direct Form II Structure

Canonic structure with respect to delays

Additional Direct Form I Structures

- Factoring the numerator and denominator

\[ H(z) = \frac{P(z)P_2(z)P_3(z)}{D_1(z)D_2(z)D_3(z)} \]

- 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
- For a first-order section $\beta_2=0$
- 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
- General structure:

Parallel Realizations

- Parallel realizations are obtained by making use of the partial fraction expansion of the transfer function
- 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:
- 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 $\omega_0$
- 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 $\omega_0$ 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 90°

**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
  
  \[H_1(z) = \frac{1 - (\cos \omega_0) z^{-1}}{1 + (\cos \omega_0) z^{-1} + z^{-2}}\]
  
  \[H_2(z) = \frac{\sin \omega_0 z^{-2}}{1 + (\cos \omega_0) z^{-1} + z^{-2}}\]

- Solving the signal values from the structure
  
  \[y_1[n] = w_1[n] - \cos(\omega_0) w_1[n-1]\]
  
  \[y_2[n] = \sin(\omega_0) w_1[n-1]\]

  \[w_2[n] = w_1[n-1]\]
  
  \[w_1[n] = 2\cos(\omega_0) w_1[n-1] - w_2[n-1] + z[n]\]
  
  \[w_1[n] = 2\cos(\omega_0) w_1[n-1] - w_1[n-2] + z[n]\]

- Using the $z$-transform and solving $W_1(z)$
  
  \[W_1(z) = 2\cos(\omega_0) z^{-1}W_1(z) - z^{-2}W_1(z) + X(z)\]
  
  \[\implies W_1(z)\left[1 - 2\cos(\omega_0) z^{-1} + z^{-2}\right] = X(z)\]

- 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
  
  \[ Y_1(z) = W_1(z) \left[ \frac{1 - \cos(\omega_0) z^{-1}}{1 - 2 \cos(\omega_0) z^{-1} + z^{-2}} \right] X(z) \]
  
  \[ Y_2(z) = \sin(\omega_0) z^{-1} W_1(z) = \frac{\sin(\omega_0) z^{-1}}{1 - 2 \cos(\omega_0) z^{-1} + z^{-2}} X(z) \]

- Solving the system functions $H_1(z)$ and $H_2(z)$ gives
  
  \[ H_1(z) = \frac{X(z)}{X(z)} \frac{1 - \cos(\omega_0) z^{-1}}{1 - 2 \cos(\omega_0) z^{-1} + z^{-2}} \]
  
  \[ H_2(z) = \frac{Y_1(z)}{X(z)} = \frac{\sin(\omega_0) z^{-1}}{1 - 2 \cos(\omega_0) z^{-1} + z^{-2}} \]

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

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.