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

Digital Filter Structures

Time-Domain Characterizations

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

Linear Constant Coefficient Difference Equation:
\[ y[n] = \sum_{k=0}^{M} a_k y[n-k] + \sum_{k=0}^{N} b_k x[n-k] \]

State-Space Equations:
\[ s[n+1] = A s[n] + B x[n] \]
\[ y[n] = C s[n] + D x[n] \]

Basic Building Blocks

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

Basic Operations

- Addition / Subtraction
- Multiplication (constant coefficient
- Delay (memory)

Example: First-order digital filter
\[ y[n] = p_0 x[n] + p_1 x[n-1] - d_1 y[n-1] \]

The Delay-Free Loop Problem

- A block diagram containing delay-free loops is physically non-realizable
- Example:

\[ y[n] = B \left[ A \left( x[n] + w[n] \right) + u[n] \right] \]
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 its $M-1$ 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:
  \[ H(z) = h(0) + h(2)z^{-2} + h(4)z^{-4} + h(6)z^{-6} + h(8)z^{-8} \]
- Expressing the above equation as a sum of two terms, one containing the even-indexed coefficients, and the other containing the odd-indexed coefficients:
  \[ H(z) = \left( h(0) + h(2)z^{-2} + h(4)z^{-4} + h(6)z^{-6} + h(8)z^{-8} \right) \]
  \[ + z^{-2}\left( h(1) + h(3)z^{-2} + h(5)z^{-4} + h(7)z^{-6} \right) \]
Polyphase Realization

- Using the notations
  \[ H(z) = H_0(z) + H_1(z)z^{-1} + H_2(z)z^{-2} + H_3(z)z^{-3} + H_4(z)z^{-4} \]

- In a similar manner, by grouping the terms differently, the transfer function can be rewritten as
  \[ H(z) = E_1(z^{-1}) + E_2(z^{-2}) + E_3(z^{-3}) + E_4(z^{-4}) \]

- The subfilters \( E_k(z^{-k}) \) are also FIR filters.

Polyphase Decomposition

- In general, an \( L \)-branch polyphase decomposition of the transfer function \( H(z) \) of order \( M-1 \) is of the form
  \[ H(z) = \sum_{m=0}^{L-1} z^{-m} E_m(z^{-1}) \]
  where \( E_m(z) = \sum_{n=0}^{\frac{M-1}{L}} h[Ln+m] z^{-n}, \quad 0 \leq m \leq L-1 \)

- The subfilters \( E_m(z^{-1}) \) are also FIR filters.

Linear-Phase FIR Structures

- Length \( M \) is odd (\( M=7 \))
  \[ H(z) = h[0](z+z^{-7}) + h[1](z^{-1}+z^{-6}) + h[2](z^{-2}+z^{-5}) + h[3]z^{-3} \]

- Length \( M \) is even (\( M=8 \))
  \[ H(z) = h[0](z+z^{-8}) + h[1](z^{-1}+z^{-7}) + h[2](z^{-2}+z^{-6}) + h[3](z^{-3}+z^{-5}) \]
Basic IIR Filter Structures

- The transfer function is rational
- Direct forms: 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
  \[ H_1(z) = \frac{W(z)}{X(z)} = \frac{P(z)}{D(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}} \]
  \[ H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}} \]

• Considering the basic cascade realization results in Direct Form I:

• Changing the order of blocks in cascade results in Direct Form II:

- 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

\[ x[n] \rightarrow w_2[n] \rightarrow y[n] \]
**Cascade Realizations**

- Factoring the numerator and denominator
  \[ H(z) = \frac{P(z)D(z)}{P(z)D(z)} \]
  
- Various alternatives in pairing the poles and zeros

- 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

\[ H(z) = \prod \left( \frac{1 + \beta z^{-1} + \beta z^{-2}}{1 + \alpha z^{-1} + \alpha z^{-2}} \right) \]

- For a first-order section \( \alpha_2 = \beta_2 = 0 \)

- 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

\[ H(z) = \prod \left( \frac{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}} \right) \]

- General structure:
Parallel Realizations

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

\[
H(z) = \sum \left( \frac{\gamma_i}{1 + z^{-1}} \right)
\]

- General structure:

\[
H(z) = \delta_i \sum \left( \frac{\delta_i z^{-1}}{1 + a_i z^{-2} + b_i z^{-1}} \right)
\]

- 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:

\[
\begin{bmatrix}
x_i[n+1] \\
x_f[n+1]
\end{bmatrix} =
\begin{bmatrix}
a_i & a_n \\
a_{i-1} & a_{n-1}
\end{bmatrix}
\begin{bmatrix}
x_i[n] \\
x_f[n]
\end{bmatrix} + \begin{bmatrix} b_i \end{bmatrix} x[n]
\]

\[
y[n] = c_i \begin{bmatrix} x_i[n] \end{bmatrix} + d[n]
\]

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