Chapter 8

Digital Filter Structures

1. Block Diagram Representation

- 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

---

1. Block Diagram Representation

- The convolution sum description of an LTI discrete-time system can, in principle, be used to implement the system.
- For an IIR finite-dimensional system, this approach is not practical as here the impulse response is of infinite length.
- However, a direct implementation of the IIR finite-dimensional system is practical

---

1. Block Diagram Representation

- In the time domain, the input-output relations of an LTI digital filter is given by the convolution sum or, by the linear constant coefficient difference equation
- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid computational algorithm.
1. Block Diagram Representation

Using the above equation we can compute \( y(n) \) for \( n \geq 0 \) knowing the initial condition \( y(-1) \) and the input \( x(n) \) for \( n \geq -1 \)

\[
\begin{align*}
y(0) &= -d_1y(-1) - p_2x(0) + p_3x(-1) \\
y(1) &= -d_1y(0) + p_2x(1) + p_3x(0) \\
y(2) &= -d_1y(1) + p_2x(2) + p_3x(1)
\end{align*}
\]

We can continue this calculation for any value of \( n \) we desire (by iterative computation).

1.1 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:

- **Advantages of block diagram/signal flow chart representation**
  1. Easy to write down the computational algorithm by inspection.
  2. Easy to analyze the block diagram to determine the explicit relation between the output and input.

1.2 Analysis of Block Diagrams

Steps of Analyzing Block Diagrams
- Carried out by writing down the expressions for the output signals of each adder as a sum of its input signals, and developing a set of equations relating the filter input and output signals in terms of all internal signals.
- Eliminating the unwanted internal variables then results in the expression for the output signal as a function of the input signal and the filter parameters that are the multiplier coefficients.
1.2 Analysis of Block Diagrams

**Example**
- Consider the single-loop feedback structure shown below.

The output $E(z)$ of the adder is $E(z) = X(z) + G_2(z)Y(z)$
But from the figure, $Y(z) = G_1(z)E(z)$

1.2 Analysis of Block Diagrams

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

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

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

1.3 Canonic and 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 in the next slide is noncanonic as it employs two delays to realize a first-order difference equation.

1.3 Canonic and Noncanonic Structures

- \[ y(n) = -d_{y}y(n-1) + p_{x}x(n) + p_{x}x(n-1) \]

2. Equivalent Structures

- Two digital filter structures are defined to be equivalent if they have the same transfer function.
- We describe next a number of methods for the generation of equivalent structures.
- However, a fairly simple way to generate an equivalent structure from a given realization is via the transpose operation.

2. Equivalent Structures

**Transpose Operation**

(1) Reverse all paths
(2) Replace pick-off nodes by adders, and vice versa
(3) Interchange the input and output nodes

- All other methods for developing equivalent structures are based on a specific algorithm for each structure.
2. Equivalent Structures

- There are literally an infinite number of equivalent structures realizing the same transfer function.
- It is thus impossible to develop all equivalent realizations.
- In this course we restrict our attention to a discussion of some commonly used structures.

2. Equivalent Structures

- Under infinite precision arithmetic any given realization of a digital filter behaves identically to any other equivalent structure.
- However, in practice, due to the finite wordlength 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 wordlength effects in each case, and select the one showing the least effects.

3. FIR Digital Filter Structures

- In certain cases, it is possible to develop a structure that by construction has the least quantization effects.
- We defer the review of these structures after a discussion of the analysis of quantization effects (not included in Kuo’s revised book).
- Here, we review some simple realizations that in many applications are quite adequate.

3. FIR Digital Filter Structures

- A causal FIR filter of order $N-1$ is characterized by a transfer function $H(z)$ given by
  $$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$
  which is a polynomial in $z^{-1}$.
- In the time-domain the input-output relation of the above FIR filter is given by
  $$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$
3.1 Direct Form FIR Digital Filter Structures

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

A direct form realization of an FIR filter can be readily developed from the convolution sum description as indicated below for $N = 5$.

$$y(n) = h(0)x(n) + h(1)x(n - 1) + h(2)x(n - 2) + h(3)x(n - 3) + h(4)x(n - 4)$$

which is precisely of the form of the convolution sum description.

The direct form structure shown on the previous slide is also known as a tapped delay line or a transversal filter.

3.2 Cascade Form FIR Digital 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

$$H(z) = h(0) \prod_{k=1}^{N/2} \left(1 + \beta_{2k} z^{-1} + \beta_{2k+1} z^{-2}\right)$$

where $k = N/2$ if $N$ is even, and $k = (N+1)/2$ if $N$ is odd, with $\beta_{2k} = 0$. 

A cascade realization for $N = 6$ is shown below.
3.3 Linear-Phase FIR Digital Filter Structures

- Linear-phase FIR filter of length $N$ is characterized by the symmetric impulse response $h(n) = h(N - 1 - n)$
- An antisymmetric impulse response condition $h(n) = -h(N - 1 - n)$ results in a constant group delay and “linear-phase” property
- Symmetry of the impulse response coefficients can be used to reduce the number of multiplications

Length $N$ is odd ($N=7$)

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

Length $N$ is even ($N=8$)

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

General Form

$V$ is even Type 1 and 3

$V$ is odd Type 2 and 4

$N/2$ multipliers
Direct Form needs $N$ multipliers
$(N+1)/2$ multipliers
4. IIR Digital Filter Structures

4.1 Direct Form IIR Digital Filter Structures

- Direct Form
- Cascade Form
- Parallel Form

The causal IIR digital filters we are concerned with in this course are characterized by a real rational transfer function of $z^{-1}$ or, equivalently by a constant coefficient difference equation.

From the difference equation representation, it can be seen that the realization of the causal IIR digital filters requires some form of feedback.

We consider for simplicity a 3rd-order IIR filter with a transfer function (assuming $d_0 = 1$)

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

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

The time-domain representation of $H_2(z)$ is given by

$$y(n) = w(n) - d_1 y(n-1) - d_2 y(n-2) - d_3 y(n-3)$$

Realization of $H_2(z)$ follows from the above equation and is shown below.

Considering the basic cascade realization results in Direct form 1:

$$H(z) = P(z) \cdot \frac{1}{D(z)}$$
4.1 Direct Form IIR Digital Filter Structures

- Changing the order of blocks in cascade results in Direct form II:

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

- Observe in the direct form structure shown below, the signal variable at nodes 1 and 1' are the same, and hence the two top delays can be shared
- Likewise, the signal variables at nodes 2 and 2' are the same, permitting the sharing of the middle two delays
- Following the same argument, the bottom two delays can be shared

Sharing of all delays reduces the total number of delays to 3 resulting in a canonic realization shown below along with its transpose structure.

4.2 Cascade Realizations

- 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 (often sos)
- Consider, for example, \( H(z) = P(z)/D(z) \) expressed as

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

- There are altogether a total of 36 \( (P_1^2 \cdot P_2^2) \) different cascade realizations of \( H(z) \) based on different pole-zero pairings and ordering
- Due to finite wordlength effects, each such cascade realization behaves differently from others
4.2 Cascade Realizations

- One possible realization is shown below.

- General structure:

\[ H_1(z) \rightarrow H_2(z) \rightarrow H_{hc}(z) \]

4.2 Cascade Realizations

- Usually, the polynomials are factored into a product of 1st-order and 2nd-order (sos) polynomials:

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

- for a first-order factor \( \alpha_{21} = \beta_{21} = 0 \)

4.3 Parallel Realizations

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

**Parallel Form I:**

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

**Parallel Form II:**

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

4.3 Parallel Realizations

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

- Example:

\[
H(z) = \frac{P(z)}{D(z)} = p_0 \left( \frac{1 + \beta_{11} z^{-1}}{1 + \alpha_{11} z^{-1} + \alpha_{12} z^{-2}} \right) \left( \frac{1 + \beta_{12} z^{-1} + \beta_{13} z^{-2}}{1 + \alpha_{13} z^{-1} + \alpha_{14} z^{-2}} \right)
\]
4.3 Parallel Realizations

- General structure:

  \[ H_1(z) \quad H_2(z) \quad \vdots \quad H_N(z) \]

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

**Example**

- A partial-fraction expansion of
  \[ H(z) = \frac{0.44 + 0.352z^{-2} + 0.002z^{-3}}{1 + 0.4z^{-1} + 0.18z^{-2} - 0.2z^{-3}} \]
  in \( z^{-1} \) yields
  \[ H(z) = -0.1 + \frac{0.6}{1 - 0.4z^{-1}} + \frac{-0.5 - 0.2z^{-1}}{1 + 0.8z^{-1} + 0.5z^{-2}} \]

- Likewise, a partial-fraction expansion of \( H(z) \) in \( z \) yields
  \[ H(z) = \frac{0.24z^{-1} + 0.2z^{-3} + 0.25z^{-2}}{1 - 0.4z^{-1} + 1.0.8z^{-3} + 0.5z^{-2}} \]

**Their realizations are shown below**