1 Lecture Outline

Reading: Chapter 7 Digital Filter Realizations

- Direct Form I
- Canonical Form a.k.a. Direct Form II
- Cascade (of Second-Order Sections) Form

2 Direct Form I

Consider the general form of a transfer function

$$H(z) \quad = \quad \frac{b_0 + b_1 z^{-1} + \cdots + b_L z^{-L}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}} \tag{1}$$

and its corresponding difference equation

$$y(n) \quad = \quad -a_1 y(n-1) - \cdots + a_M y(n-M) + b_0 x(n) + b_1 x(n-1) + \cdots + b_L x(n-L). \tag{2}$$

If we define the internal states (contents of the delay registers) of the filter to be

$$\begin{aligned} v_i(n) &= x(n-i), \quad i = 0, \ldots, L \\ w_i(n) &= y(n-i), \quad i = 1, \ldots, M \end{aligned} \tag{3}$$

then the difference equation can be rewritten as

$$w_0(n) \quad = \quad -a_1 w_1(n) - \cdots + a_M w_M(n) + b_0 v_0(n) + b_1 v_1(n) + \cdots + b_L v_L(n). \tag{4}$$

Figure 1: Orfanidis p. 267 Figure 7.1.2 Direct-form realization of an Mth order IIR filter.

The internal states are updated via

$$\begin{aligned} v_i(n+1) &= v_{i-1}(n), \quad i = 0, \ldots, L \\ w_i(n+1) &= w_{i-1}(n), \quad i = 1, \ldots, M \end{aligned} \tag{5}$$

which leads to the sample-by-sample processing algorithm for the direct form (also known as direct form I) realization

The main features of the direct-form realization are

- single accumulator for w_0

Figure 2: Orfanidis p. 268 sample processing algorithm for a direct-form realization of an Mth order IIR filter.

- $2(M + L + 1)$ of words of memory
  - M words for internal states $w_1, \ldots, w_M$
  - $L + 1$ words for internal states $v_0, \ldots, v_L$
  - M words for feedback coefficients $a_1, \ldots, a_M$
  - $L + 1$ words for feedforward coefficients $b_0, \ldots, b_{L+1}$

3 Canonical

Definition (Canonical): the simplest form of something.

We'll take simplest form of something to mean the realization with the *least* amount of memory. Again, consider the general form of a transfer function

$$\begin{aligned} H(z) &= \frac{b_0 + b_1 z^{-1} + \cdots + b_L z^{-L}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}} \\ &= \frac{N(z)}{D(z)} \\ &= N(z) \frac{1}{D(z)}. \end{aligned} \tag{6}$$

We can think of this factorization of the transfer function as a grouping of recursive (feedback or IIR) and nonrecursive (FIR) terms of the I/O difference equation

$$y(n) = [b_0 x(n) + b_1 x(n-1) + \cdots + b_L x(n-L)] + [-a_1 y(n-1) - \cdots + a_M y(n-M)]. \tag{7}$$

Figure 3: Orfanidis p. 271 Figure 7.2.1 Regrouping of direct-form terms.

We can commute the factors of $H(z)$ so that

$$H(z) = \frac{1}{D(z)} N(z). \tag{8}$$

In this case the realization becomes

Figure 4: Orfanidis p. 272 Figure 7.2.2 Interchanging $N(z)$ and $1/D(z)$.

Figure 5: Orfanidis p. 272 Figure 7.2.3 Canonical realization form of 2nd order IIR filter.

Figure 6: Orfanidis p. 274 sample processing algorithm for a canonical realization of an Mth order IIR filter.

The difference equation for this realization is written as

$$\begin{aligned} w_0(n) &= x(n) - a_1 w_1(n) - a_2 w_2(n) - \cdots + a_M w_M(n) \\ y(n) &= b_0 w_0(n) + b_1 w_1(n) + \cdots + b_L w_L(n). \end{aligned} \tag{9}$$

Since there is a common set of internal states, the number of internal states for the filter must be

$$K = \max(L, M). \tag{10}$$

Therefore the K internal states are updated via

$$w_i(n+1) = w_{i-1}(n), \quad i = 0, 1, \ldots, K \tag{11}$$

which leads to the sample-by-sample processing algorithm for the canonical form (also known as direct form II) realization:

The main features of the canonical realization are

- dual accumulator for w_0 and y
- $K + M + (L + 1)$ of words of memory

- K words for internal states $\max(L, M)$
- M words for feedback coefficients $a_1, \ldots, a_M$
- $L + 1$ words for feedforward coefficients $b_0, \ldots, b_{L+1}$

4 Cascade Form

We may express $H(z)$ with the factorization

$$\begin{aligned} H(z) &= \frac{b_0 + b_1 z^{-1} + \cdots + b_L z^{-L}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}} \\ &= \prod_{i=0}^{K-1} H_i(z) \\ &= \prod_{i=0}^{K-1} \frac{b_{i0} + b_{i1} z^{-1} b_{i2} z^{-2}}{a_{i0} + a_{i1} z^{-1} a_{i2} z^{-2}} \end{aligned} \tag{12}$$

assuming $M = L$)–if the number of real factors is odd, we'll have a single first-order term in the product. We can then implement $H(z)$ as a cascade of second-order sections (SOS) [each in canonical form (DFII)].

Figure 7: Orfanidis p. 278 Figure 7.3.1 Canonical realization form of Mth order IIR filter.

Theorem: If poles (and zeros) of $H(z)$ are tightly clustered, small changes in the coefficients (as with coefficient quantization on a fixed-point signal processor) can produce large shifts in the poles (zeros).

Proof: (Problem 7.22)

Since a large pole shift could cause our DT system to go unstable, IIR systems (of order greater than 2) should always be implemented as a cascade of second-order sections if the target processor is fixed point. This problem is not as critical in a floating point processor.