1 Lecture Outline

Reading: Chapter 3 Discrete-Time Systems (review)

- FIR and IIR filters
- Time domain methods to compute (simple) impulse responses

Note that we will only introduce a few new concepts in Chapter 3 since this chapter is mainly a review from a standard undergraduate course in signals and systems.

2 Digital Filters Introduction (3.4)

DT LTI systems can be classified into FIR or IIR systems depending on whether they have an impulse response that if finite duration (FIR) or infinite duration (IIR).

The discrete-time convolution equation is

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k). \tag{1}$$

2.1 FIR Filters (3.4)

An FIR filter has impulse response $h(n)$ that extends only over a finite time interval say $0 \leq n \leq M$ and is identically zero beyond that

$$\{h(0),\ h(1),\ h(2),\ \ldots, h(M),\ 0,\ 0, \ldots\} \tag{2}$$

The convolution equation is thus

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k). \tag{3}$$

In the FIR case, the length of the impulse response is $M+1$ samples

$$\mathbf{h} = [h(0),\ h(1),\ h(2),\ \ldots, h(M)]^T. \tag{4}$$

The elements of this vector are called filter coefficients and thus $\mathbf{h}$ is often called the coefficient vector.

Figure 1: Orfanidis p. 105 Fig. 3.4.1 Impulse response classes

Example: Given an FIR filter with coefficients

$$\mathbf{h} = [h(0),\ h(1),\ h(2)]^T = [1,\ 2,\ 3]^T \tag{5}$$

and an input signal

$$\mathbf{x} = [x(0),\ x(1),\ x(2),\ x(3)]^T = [4,\ 5,\ 6,\ 7]^T \tag{6}$$

compute the output $y(n) = h(n) \star x(n)$. Assume the system is at rest, i.e. $y(-1) = y(-2) = 0$.

We can view the convolution for FIR filters as an inner product

$$y(n) = \mathbf{h}^T \mathbf{x}(n) \tag{7}$$

where the *regressor* is $\mathbf{x}(n) = [x(n),\ x(n-1),\ \ldots,\ x(n-M)]^T$. Thus we have for the output samples,

$$\begin{aligned}
y(0) &= \mathbf{h}^T \mathbf{x}(0) = [1,\ 2,\ 3] \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} = 4 \\
y(1) &= \mathbf{h}^T \mathbf{x}(1) = [1,\ 2,\ 3] \begin{bmatrix} 5 \\ 4 \\ 0 \end{bmatrix} = 13 \\
y(2) &= \mathbf{h}^T \mathbf{x}(2) = [1,\ 2,\ 3] \begin{bmatrix} 6 \\ 5 \\ 4 \end{bmatrix} = 28 \\
y(3) &= \mathbf{h}^T \mathbf{x}(3) = [1,\ 2,\ 3] \begin{bmatrix} 7 \\ 6 \\ 5 \end{bmatrix} = 34
\end{aligned} \tag{8}$$

Note that the basic operation in the inner product is a Multiply and ACcumulate. Virtually all Digital Signal Processors (DSPs) have a MAC instruction which can execute (effectively) in a single clock cycle.

3 IIR Filters (3.4)

An IIR filter has an impulse response $h(n)$ of infinite duration, defined over the infinite interval $0 \leq n \leq \infty$. The convolution equation then becomes

$$y(n) \quad = \quad \sum_{k=0}^{\infty} h(k)x(n-k). \tag{9}$$

This I/O equation is not computationally feasible because we are required to sum an infinite number of terms. We therefore restrict the class of IIR filters to those for which the infinite number of filter coefficients are coupled through a constant-coefficient linear difference equation

$$y(n) \quad = \quad -\sum_{k=1}^{M} a_k y(n-k) + \sum_{k=0}^{L} b_k x(n-k). \tag{10}$$

For this I/O equation the impulse response of our special class of IIR filters satisfies constant-coefficient difference equations of the general type

$$h(n) \quad = \quad -\sum_{k=1}^{M} a_k h(n-k) + \sum_{k=0}^{L} b_k \delta(n-k). \tag{11}$$

Figure 2: Orfanidis p. 269 Fig. 7.1.3 IIR filter block diagram

Figure 3: I/O relations and the impulse response

4 Time domain methods to compute (simple) impulse responses

Example: Determine the impulse response of the digital filter

$$y(n) \quad = \quad x(n-3). \tag{12}$$

A sample-by-sample I/O table gives

n	$x(n)$	$y(n)$
0	1	0
1	0	0
2	0	0
3	0	1
4	0	0
5	0	0

thus (with no surprise)

$$h(n) \quad = \quad \delta(n-3). \tag{13}$$

Example: Determine the impulse response of the 4-pt "moving average" digital filter

$$y(n) \quad = \quad 0.25\left[x(n)+x(n-1)+x(n-2)+x(n-3)\right]. \tag{14}$$

By definition we have

$$h(n) \quad = \quad 0.25\left[\delta(n)+\delta(n-1)+\delta(n-2)+\delta(n-3)\right]. \tag{15}$$

Example: Determine the impulse response of the digital filter

$$y(n) \quad = \quad ay(n-1)+x(n). \tag{16}$$

By definition we have

$$h(n) = ah(n-1) + \delta(n), \tag{17}$$

however, this coupled or recursive equation is not too useful. We'd rather have a closed form solution. A sample-by-sample I/O table gives

n	$x(n)$	$y(n)$
0	1	1
1	0	a
2	0	a^2
3	0	a^3
4	0	a^4
5	0	a^5

which is generalized to

$$h(n) = \begin{cases} a^n, & n \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$