

# Chapters 9 and 10

## Digital Filter Structures and Design

The goal of this section is to allow for simple realization of causal IIR and FIR digital filters.

A structural representation using interconnected basic building blocks is the first step in the hardware or software implementation of an LTI digital filter.

### Block Diagram Representation

Consider a filter with impulse response  $h[n]$ . What is the general input-output relation?

Convolution sum:

Basic building blocks of a digital filter structure consists of:

adder            multiplier            unit delay            pick-off node (feed)

An alternative way to describe a filter, as opposed to convolution sum, is the difference equation.

Example (an accumulator):

$$y[n] = y[n - 1] + x[n]$$

## Canonic and Non-canonic Structures

A realization of a filter which is minimal in its number of delay blocks is called *canonical* given the constraint that no loop is delay-free. The minimal number of delay blocks needed is equal to the order of the filter (order of its difference equation, transfer function, etc).

**Recall:** Order of a system with the following difference equation

$$\sum_{k=0}^N d_k y[n-k] = \sum_{k=0}^M p_k x[n-k]$$

is defined to be  $\max(M, N)$ .

Example:

## Analysis of Block Diagrams

A block diagram can be analyzed in the following steps:

1. write down the output of adders
2. eliminate the internal signals (write them as functions of input signals and multiplier coefficients)

Example:

Note that many block diagrams might result in the same transfer function  
Example:

## Equivalent Structures and Transpose Operation

A very simple way to make equivalent structure is:

1. reverse all paths
2. replace pick-off nodes by address and vice versa
3. interchange the input and the out
4. redraw the figure (optional)

Example:

## Cascade and Parallel Forms

Consider two filters each with transfer functions  $H_1(z)$  and  $H_2(z)$ . We can combine the two filters in the following ways:

$$H_c(z) = H_1(z)H_2(z)$$

$$H_p(z) = H_1(z) + H_2(z)$$

This is true for representation of digital filters also:  
Example:

## Basic FIR Digital Filter Structure

In this section, we look at how to represent an FIR filter (we call these direct representations).

**Recall:** A causal FIR filter is a system with a finite impulse response. Use this and convolution sum:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=0}^M h[k]x[n-k]$$

Example: Consider an FIR filter of order 2.

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2]$$

The transpose operation can be used to find an alternative representation:

## Basic IIR Digital Filter Structure

**Recall:** A causal IIR filter is generically represented by a difference equation of the following form:

$$\sum_{k=0}^N d_k y[n-k] = \sum_{k=0}^M p_k x[n-k]$$

Without loss of generality, we can assume  $d_0 = 1$ .

Now let's use an intermediate variable  $w[n]$  as follows:

$$w[n] = \sum_{k=0}^M h[k] x[n-k]$$

and

$$w[n] = y[n] + \sum_{k=1}^N d_k y[n-k]$$

Each of these equations can now be viewed as FIR filters. We use direct form representations of these filters:

The transpose operation can be used to find an alternative representation: