Digital Filters
Structure and Design

The goal of this section is to allow for simple realization of causal IIR and FIR digital filters.
Before we proceed, recall that we saw the magnitude of $H(e^{j\omega})$

$$|H(e^{j\omega})| = \sqrt{H(e^{j\omega})H(e^{-j\omega})} = \sqrt{H(z)H(z^{-1})}$$

at $z = e^{j\omega}$

Find the phase of $H(e^{j\omega})$

$$\angle H(e^{j\omega}) = \frac{1}{2j} \ln \left[ \frac{H(e^{j\omega})}{H(e^{-j\omega})} \right] = \frac{1}{2j} \ln \left[ \frac{H(z)}{H(z^{-1})} \right]$$

at $z = e^{j\omega}$
$$H(z) = \frac{\sum_{m=0}^{M} p_m z^{-m}}{\sum_{n=1}^{N} d_n z^{-n}}$$

Now we can calculate the frequency response for a stable rational transfer function with real-coefficients:

$$|H(e^{j\omega})|^2 = \left| \frac{p_0}{d_0} \right|^2 \prod_{k=1}^{M} \frac{|(e^{j\omega} - \zeta_k)(e^{j\omega} - \zeta_k^*)|}{\prod_{k=1}^{N} |(e^{j\omega} - \lambda_k)(e^{j\omega} - \lambda_k^*)|}$$

and

$$\angle H(e^{j\omega}) = \angle \frac{p_0}{d_0} + \omega(N - M) + \sum_{k=1}^{M} \angle(e^{j\omega} - \zeta_k) + \sum_{k=1}^{M} \angle(e^{j\omega} - \lambda_k)$$

**Applications**

Consider two filters:

$$H_1(z) = \frac{z + b}{z + a} \quad \text{and} \quad H_2(z) = \frac{bz + 1}{z + a}$$

Can you say something about $|H_1(\Omega)|$ versus $|H_2(\Omega)|$?

$$|H_1(\Omega)| = |H_1(e^{j\omega})| = \sqrt{H(e^{j\omega}) H(e^{-j\omega})} = \sqrt{\left\{ \begin{array}{ll} \frac{2 + b}{2 + a} & \frac{\sqrt{2} - b}{\sqrt{2} + a} \\
\frac{b + 1}{2 + a} & \frac{\sqrt{2} + b}{\sqrt{2} + a} \end{array} \right\}$$

How about $\angle H_1(\Omega)$ versus $\angle H_2(\Omega)$?

$$\angle H_1(e^{j\omega}) = \tan^{-1} \left( \frac{\sin \omega}{b + c \omega} \right) = \tan^{-1} \left( \frac{\sin \omega}{\omega(a + c \omega)} \right)$$

$$\angle H_2(e^{j\omega}) = \tan^{-1} \left( \frac{b \sin \omega}{\sqrt{2} + c \omega} \right) - \tan^{-1} \left( \frac{\sin \omega}{\omega(a + c \omega)} \right)$$

Can you generalize this to any filter with a stable rational transfer function with real coefficients:
Block Diagram Representation

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

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

Convolution sum:

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

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:
\[
W_1 = X - \alpha z^{-1} W_3 = X - \alpha z^{-1} (z^{-1} + \delta) W_2 \\
W_2 = W_1 - \delta z^{-1} W_2 \quad \Rightarrow \quad W_1 = (1 + \delta z^{-1}) W_2 \\
W_3 = z^{-1} W_2 + \epsilon W_2 = (z^{-1} + \epsilon) W_2 \\
Y = \beta W_1 + y z^{-1} W_3 = \left[ \beta (1 + \delta z^{-1}) + y z^{-1} (z^{-1} + \epsilon) \right] W_2
\]

\[
\Rightarrow X = (1 + \delta z^{-1}) W_2 + \alpha z^{-1} (z^{-1} + \epsilon) W_2 \\
Y(z) = X(z) \left[ \frac{\beta (1 + \delta z^{-1}) + y z^{-1} (z^{-1} + \epsilon)}{1 + \delta z^{-1} + \alpha z^{-1} (z^{-1} + \epsilon)} \right]
\]

\[
\frac{Y(z)}{X(z)} = \frac{\beta + (\beta \delta + y \epsilon) z^{-1} + Y z^{-2}}{1 + (\delta + \alpha \epsilon) z^{-1} + \alpha z^{-2}}
\]

\[
y[n] + (\delta + \alpha \epsilon) y[n-1] + \alpha y[n-2] = \beta x[n] + (\beta \delta + y \epsilon) x[n-1] + \delta x[n-2]
\]
Equivalent Structures and Transpose Operation

A very simple way to make equivalent structure is:

1. reverse all paths
2. replace pick-off nodes by addres 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:

$$H(z) = \prod_{k=1}^{6} \left( 1 + \beta_{1,k} \frac{z^{-1}}{z} + \beta_{2,k} \frac{z^{-2}}{z^2} \right)$$
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.


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

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

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:

you can save on delay components by combining min \((N, M)\) branches (up)
Types I-IV FIR Filters

Recall that we have seen that the moving average filter is a linear phase filter. Now, we'd like to investigate other filters with similar phase property:

\[ H(\Omega) = A(\Omega)e^{(\alpha \Omega + \beta)} \]

where \( A(\Omega) \) is a real function.

Show that this filter acts like a delay line.

\[ C = \frac{d}{d\Omega} H(\Omega) \]

Turns out that IIR filters can never ensure a linear phase. In other words, only FIR filters achieve this goal.
Type I and II Filters

Let an FIR filter with length $N$ be such that $h[n] = h[N - n]$. Consider two cases: when $N$ is even (Type-I) versus is odd (Type-II):

$$
\begin{align*}
H(\omega) &= H^*(-\omega) \\
H(\omega) &= A(\omega) e^{j(\omega N + \beta)} \\
A(\omega) e^{j(\omega N + \beta)} &= A(-\omega) e^{-j(\omega N + \beta)}
\end{align*}
$$

Then $h[n] = h[N-n] \Rightarrow H(\omega) = e^{-j\omega n}$.

$$
\begin{align*}
A(\omega) e^{j(\omega N + \beta)} &= e^{j\omega n} e^{-j\omega n} \quad \text{real} \\
&= e^{j(\omega N + \beta)}
\end{align*}
$$

$$
\begin{align*}
A(\omega) &= e^{j(\omega N + 2\pi) n} \\
A(-\omega) &= e^{j(\omega N + 2\pi) n}
\end{align*}
$$

Possible when $\omega = \frac{N}{2}$, $A(\omega) = A(-\omega)$.

Plug in $\mathbb{C}$ $\Rightarrow e^{j\beta} = e^{-j\beta}$.

$$
\begin{align*}
\beta &= 0 \\
\beta &= \pi
\end{align*}
$$
Type III and IV Filters

Let an FIR filter of length $N$ be such that $h[n] = -h[N-n]$. Consider two cases: when $N$ is even (Type-III) versus is odd (Type-IV):

$$h[n] = -h[N-n] \quad \rightarrow \quad H(n) = -e^{-j\frac{N\pi}{2}} H(-n)$$

$$e^{j(nc+\beta)} A(n) = -e^{-j\frac{N\pi}{2}} e^{j(-cn+\beta)} A(-n)$$

$$\rightarrow \quad A(-n) = -e^{j\left(\frac{N+2c}{2}\right)n} A(-n)$$

Possible if

$$A(n) = -A(-n)$$

and

$$c = -\frac{N}{2}$$

On the other hand Real coefficient filter

$$H(n) = H(-n)$$

$$e^{j(nc+\beta)} A(n) = e^{j(-cn+\beta)} A(-n)$$

Plugging here we get

$$A(n) e^{j\left(-\frac{N}{2}n+\beta\right)} = A(-n) e^{\beta}$$

$$\rightarrow \quad e^{j\beta} = -e^{-j\beta} \quad \Rightarrow \quad \beta = \pm \frac{\pi}{2}$$
Type I

\[
\begin{bmatrix}
h[0] & h[1] & \cdots & h[N-1] \\
\end{bmatrix}
\]

\[\frac{N}{2} \text{ odd},\]

\[A(n) = h\left[\frac{N}{2}\right] + 2 \sum_{k=1}^{N/2} h\left[\frac{N}{2} - k\right] c_n \cdot k \cdot \pi
\]

Generalizes our 2-order filter example on page 10.

Type II. Filters

\[A(n) = 2 \sum_{k=1}^{N+1 \over 2} h\left[\frac{N+1}{2} - k\right] c_n \left(\pi\left(\frac{N}{2} - k\right)\right)
\]

Type III

\[N \text{ even; } h[n] = -h[N-n] \rightarrow h\left[\frac{N}{2}\right] \sin(k \pi)
\]

\[A(n) = 2 \sum_{k=1}^{N \over 2} h\left[\frac{N}{2} - k\right] \sin(k \pi n)
\]

Type IV

\[N \text{ odd; } h[n] = -h[N-n] \]

\[A(n) = 2 \sum_{k=1}^{(N+1)/2} h\left[\frac{N+1}{2} - k\right] \sin(n(\pi(k - 1/2))
\]
\[ a_1(n) = 6 - 6 \cos \omega + 4 \cos \omega - 2 \cos 3\omega \]

\( \text{Proof} \quad A_1(\omega) = \begin{cases} -3\pi & \text{when } A(\omega) > 0 \\ -3\pi + \pi & \text{when } A(\omega) < 0 \end{cases} \)

\( \text{Ex. 2:} \)

\[ h_2[n] = -h_1[n] \implies A_2(\omega) \supseteq A_1(\omega) \]

\( \nabla H_2(\omega) = \nabla H_1(\omega) + \pi \)

\( \text{Ex. 3} \)

\[ h_3[n] = h_3[1-n] \quad N = 1 \]

\[ H_2(z) = \frac{1}{2} (1 + \frac{1}{z}) \quad \text{has a zero } z = -1 = e^{j\pi} \]

\[ H(\omega) = 0 \quad \text{when } \omega = \pi \]

\[ H(0) = 1 \quad \text{when } \omega = 0 \]