Digital filters

Topics: Frequency response, impulse response, FIR & IIR.

A digital filter is essentially a weighted sum operating on sampled data, altering its spectral content. The basic form is shown in Figure 1.

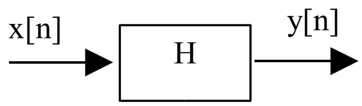


Figure 1: A digital filter, denoted H, operates on input data x[n], producing output y[n].

The digital filter's role is usually to enhance desired frequencies, supress noisy or unwanted frequencies, or simply select a certain band of frequencies from the input.

Let's start by considering a basic digital filter: the two-point average.

$$y[n] = \frac{x[n] + x[n-1]}{2}$$

As can be seen this filter averages the current and previous input sample, which makes $y[n]$ smoother than the input $x[n]$. This filter, though simple, is very useful. It eliminates alternating-sample or Nyquist frequency noise in $x[n]$. To see this, consider that at Nyquist frequency ($F = 0.5$), a sinusoid is

$$\begin{aligned} A\cos(2\pi Fn + \theta) &= A\cos(\pi n + \theta) \\ &= A\cos(\theta)\cos(\pi n) + 0 \\ &= A\cos(\theta)(-1)^n \end{aligned}$$

The values of this sinuoid alternate in sign, i.e, $A\cos\theta, -A\cos\theta, A\cos\theta, -A\cos\theta, \ldots$. Therefore, the output of the two-point average to Nyquist-frequency input is zero, since two successive samples cancel when added.

The two-point average can of course be made more general. The M-point average is

$$\begin{aligned} y[n] &= \frac{x[n] + x[n-1] + x[n-2] + \cdots x[n-(M-1)]}{M} \\ &= \frac{1}{M}\sum_{k=0}^{M-1} x[n-k] \end{aligned}$$

As M increases, this type of averaging smooths the input to a greater degree. Therefore, the M point average is a simple noise suppression filter.

Instead of smoothing the input, we may want to sharpen it. Suppose we want to remove any zero frequency, or "direct current" (DC), component in the input. A two-point difference will accomplish this:

$$y[n] = \frac{x[n] - x[n-1]}{2}$$

FIR Filters

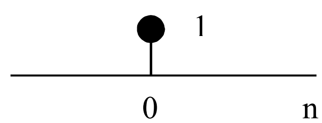
The general case of "moving average" filter is one in which input samples are added, with suitable weights, to produce the output:

$$y[n] = \sum_{k=0}^{M-1} b_k x[n-k]$$

The M-point average is the special case when all the weights $b_k = 1/M$. The general case includes smoothing, sharpening, or many other types of filters.

With any filter, it is important to understand its transient response, i.e., what the output is to a brief burst of input. The simplest kind of transient is a signal that is "on" only for one sample, and "off" otherwise. This is called an impulse function, or sometimes, a "unit-pulse" function. Mathematically, it is defined as

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$



The response of any filter to an impulse is called, appropriately enough, its "impulse response". The impulse response of the general moving average above is

$$h[n] = \sum_{k=0}^{M-1} b_k \delta[n-k]$$

(Filters are usually denoted by a capital H, and their impulse responses by $h[n]$.) It is easy enough to see that

$$h[n] = 0 \quad n < 0 \text{ or } n \geq M$$
$$h[0] = b_0,\ h[1] = b_1,\ \cdots,\ h[M-1] = b_{M-1}$$

Therefore, the impulse response of a moving average filter is finite in duration, and it is simply the weights used in the average. For that reason, moving average filters are also called FIR for "finite (duration) impulse response" filters. It is important to realize that FIR filters have a finite duration response to a transient event. This is not true for all filters, as we see later, when we study IIR or "infinite (duration) impulse response" filters, whose transient response lasts for an infinite amount of time.

Note that a FIR filter is linear: the response to any input is the sum of responses to the input components. For example if $x[n] = x_1[n] + x_2[n]$, then the output is the sum of outputs to $x_1[n]$ and $x_2[n]$ separately:

$$
\begin{aligned}
y[n] &= \sum_{k=0}^{M-1} b_k x[n-k] = \underbrace{\sum_{k=0}^{M-1} b_k x_1[n-k]}_{y_1[n]} + \underbrace{\sum_{k=0}^{M-1} b_k x_2[n-k]}_{y_2[n]} \\
&= y_1[n] + y_2[n]
\end{aligned}
$$

This linearity property is very important in studying filters: it means we can break down the problem of predicting the response to a complex input into responses to each component of the input.

Frequency response

We saw earlier how the two-point average has zero output to any Nyquist frequency input. This is an example of *frequency response.* We can calculate the frequency response for any FIR filter by observing what happens to an input sinusoid at frequency F. Suppose that $x[n] = Ae^{j2\pi Fn+\theta} = A\cos(2\pi Fn + \theta) + jA\sin(2\pi Fn + \theta)$. Then the output must be

$$
\begin{aligned}
y[n] &= \sum_{k=0}^{M-1} b_k A e^{j2\pi F(n-k)+\theta} \\
&= Ae^{j2\pi Fn+\theta} \sum_{k=0}^{M-1} b_k e^{-j2\pi Fk} \\
&= \underbrace{Ae^{j2\pi Fn+\theta}}_{x[n]} H(F), \quad \text{where} \quad H(F) = \sum_{k=0}^{M-1} b_k e^{-j2\pi Fk}
\end{aligned}
$$

This result is important in two ways. First, it says that if a sinusoid is the input to an FIR filter, then the sinusoid emerges again at the output, unchanged in frequency. However, the sinusoid's amplitude and phase are changed by multiplication with a complex number $H(F)$ that depends on F. Second, $H(F)$, which is determined by the filter coefficients, is the *frequency response* function of the filter: it specifies the response to a sinusoid of any input frequency.

The frequency response can be applied together with the linearity property to determine the response to any sum of input sinusoids. Since we know that every signal is the sum of sinusoids (from our study of spectral analysis), this means that the frequency response completely determines the response to any input signal.

Time delay and algorithmic delay

Two sampled signals $x[n]$ and $y[n]$ are called time-delayed versions of one another if $y[n] = x[n-d]$. The value of "d" is the amount of delay. A similar relationship may exist for signals before sampling, in which case the delay need not be an integer number of samples. Suppose $w(t)$ and $r(t)$ are two signals for which $r(t) = w(t-\tau)$. Then τ is the delay in seconds. After sampling, the relationship between the two signals is $r[n] = w\left(nt_s - \tau\right)$. If $\tau = dt_s + \varepsilon$, then we can express the delay as "d" samples plus a fractional delay of ε.

Since any type of digital processing takes time, we can expect some delay in observing any feature of an input signal at the output. The *algorithmic delay* of a DSP system is the number of samples from when the signal is applied to when the processed signal appears at the output. A simple measure that is often used is the sample index of the peak of the impulse response. For example, if the inpulse response of a FIR filter is

$$\{h[n]\} = \{1, -1, 2, 5, 2, -1, 1\}$$

i.e., $h[0] = 1,\ \ h[1] = -1,\ \ \text{etc}$, then the peak occurs at $n = 3$; therefore, the algorithmic delay is 3 samples.

Magnitude and phase response

Any frequency response can be represented in polar form

$$H(F) = |H(F)|\, e^{j\theta(F)}$$

Here $|H(F)|$ is the *magnitude* and $\theta(F)$ is the *phase* response. As we saw in the previous section, the response to any input sinusoid $x[n] = Ae^{j2\pi Fn}$ is

$$y[n] = Ae^{j2\pi Fn} H(F) = A\,|H(F)|\exp\left(j2\pi F(n + \frac{\theta(F)}{2\pi F})\right)$$

We see that the input sinusoid has been scaled in amplitude by $|H(F)|$ and shifted in time samples by the ratio $\theta(F)/2\pi F$. Therefore, the phase response represents a time delay applied to the input, which is potentially different for each frequency. The *phase delay* of a filter is defined as

$$\tau_p(F) = -\frac{\theta(F)}{2\pi F}$$

with the negative sign since the delay is $n - \tau_p(F)$. The phase delay can also be computed locally by applying a derivative:

$$\tau_g(F) = -\frac{1}{2\pi}\frac{d\theta(F)}{dF}$$

This is called the *group delay* of the filter. Both phase delay and group delay have units of samples.

The most important type of phase or group delay is constant: we would want the delay to be the same for all frequencies, so their time relationships to one another are preserved. Constant phase or group delay means that the phase function must be linear:

$$\theta(F) = 2\pi FK$$

where K is independent of frequency.

Examples

We can determine the frequency responses of some simple but important FIR filters, using the equation:

$$H(F) = \sum_{k=0}^{M-1} b_k e^{j2\pi Fn}$$

1) N-point average $y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$.

$$H(F) = \frac{1}{M} e^{-j\pi F(M-1)} \frac{\sin(\pi FM)}{\sin(\pi F)}$$

Note that the response is the Dirichlet function, divided by M. The response has a null at Nyquist ($F = 0.5$) only if M is even. The M-point average is a simple type of *low-pass* filter. The magnitude of this filter is plotted in Figure 2 (a).

2) First and second differences:

$$y[n] = x[n] - x[n-1] \quad \text{and} \quad y[n] = x[n] - 2x[n-1] + x[n-2]$$

The response of the first difference is

$$H(F) = 2j \cdot e^{-j\pi F} \cdot \sin \pi F$$

For the second difference, the response is

$$H(F) = -4 \cdot e^{-j2\pi F} \cdot \sin^2(\pi F)$$

Both the first and second differences are *high-pass* filters. The magnitude response of the second difference is plotted in Figure 2(b).

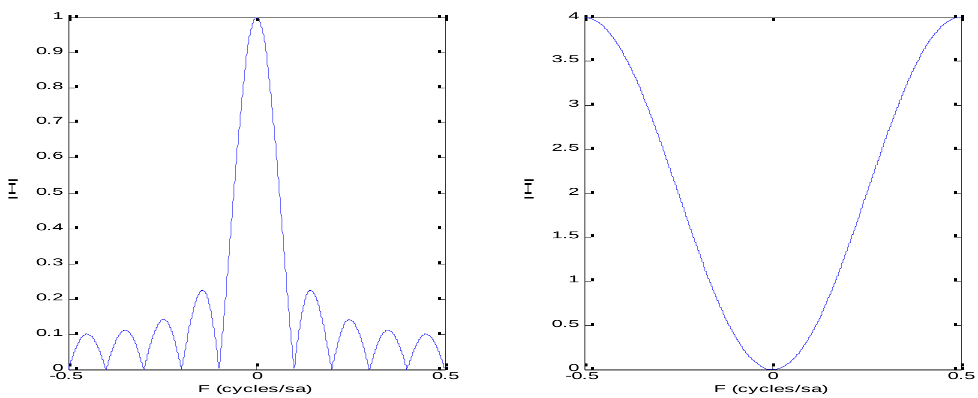
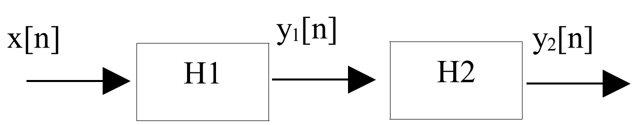


Figure 2: (a) Magnitude of 10-point average; (b) Magnitude of second difference filter

Exercises

1) Suppose that H1 and H2 are two FIR filters connected in series:



If $y_1[n] = \sum_{k=0}^{N-1} b_k x[n-k]$ and $y_2[n] = \sum_{r=0}^{M-1} c_r y_1[n-r]$ then show that

$$y_2[n] = \sum_{s=0}^{P-1} d_s x[n-s]$$

Determine d_s in terms of c_r and b_k.

2) Determine the group delay of the second difference

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