Filtering Signals and Images

One-dimensional signals (eg, audio recordings) and two-dimensional images can be *filtered* in many ways. We might use filters to

- Eliminate "static" from a radio signal.
- Remove a "hum" at some frequency from an audio recording.
- Increase the amount of "base" in music, if we like it that way.
- Enhance the location of "edges" in an image.
- Undo the effect of blurring or other distortions that have been introduced into a signal or image.

We can also characterize how the signal has been distorted in terms of a filter.

Linear Filters

We will look only at *linear* filters. If X and Y are signals or images, and F is a linear filter, then

$$\begin{aligned} F(X+Y) &= F(X) + F(Y) \\ F(cX) &= cF(X) \end{aligned}$$

where c is any real number.

By "adding" together signals (or images), we mean adding together their values at each point in time (or at each location in an image), to give a new signal (or image).

By "multiplying" a signal or image by a constant, we mean multiplying the value at each time or point by that constant.

Some Kinds of Filters

One way of characterizing a linear filter is by what it does to sinusoidal components of the signal at various frequencies — corresponding to different pitches in an audio signal.

A *low-pass* filter lets the low frequency components through, but eliminates or reduces components whose frequency is above a certain cutoff.

A *high-pass* filter lets the high frequency components through, but not the low frequency components.

A *band-pass* filter lets through components that have frequencies in a certain range.

Analog vs. Digital Filters

Filters that have various effects can be built from analog components (eg, transistors). This is the traditional way to make things like radios and stereo systems.

If we can convert the signal or image to numbers in a computer (ie, to *digitize* it), we can instead use a program to produce the filtered signal from the input signal.

We may then need to convert the resulting numbers back to an analog signal (eg, if we want to listen to it).

These conversions are done by devices called *analog-to-digital* and *digital-to-analog* converters.

Sampling and Quantization

When we digitize a continuously varying signal, we inevitably introduce innaccuracies of two types.

First, we will have to *sample* the signal at only certain times. For instance, a meteorologist might record the air temperature only every hour, even though the temperature is changing continually.

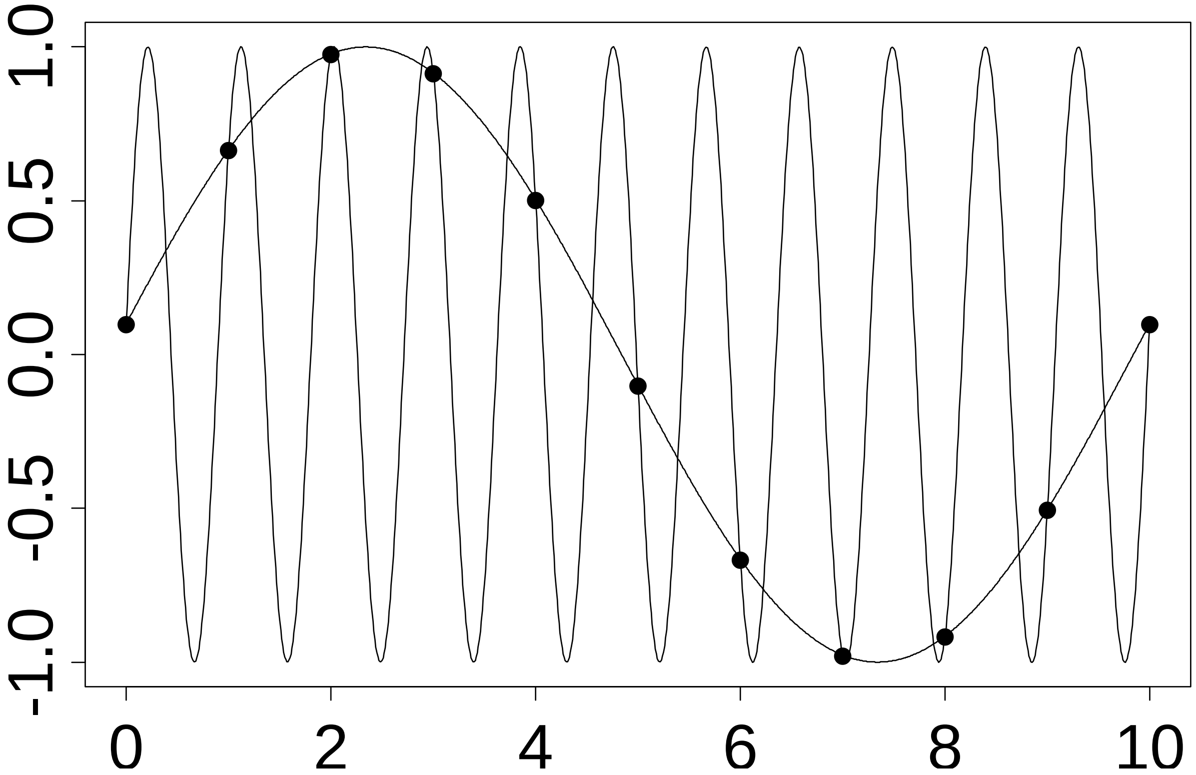
Second, we will have to *quantize* the value of the signal. For instance, the meteorologist might record the temperature (in degrees Celcius) to only one decimal place, even though the real temperature is not quantized.

We might be able to mostly ignore quantization, if the accuracy is fairly high.

Ignoring the effects of sampling can be disastrous, however!

The Problem of Aliasing

Here's what happens if we sample a signal that has a sinusoidal component whose frequency is greater than half the sampling frequency:



The true analog signal is a sine wave with frequency 1.1 cycles per time period.

We sample once per time period.

What we see looks like a sine wave with frequency 0.1 cycles per time period.

The Sampling Theorem

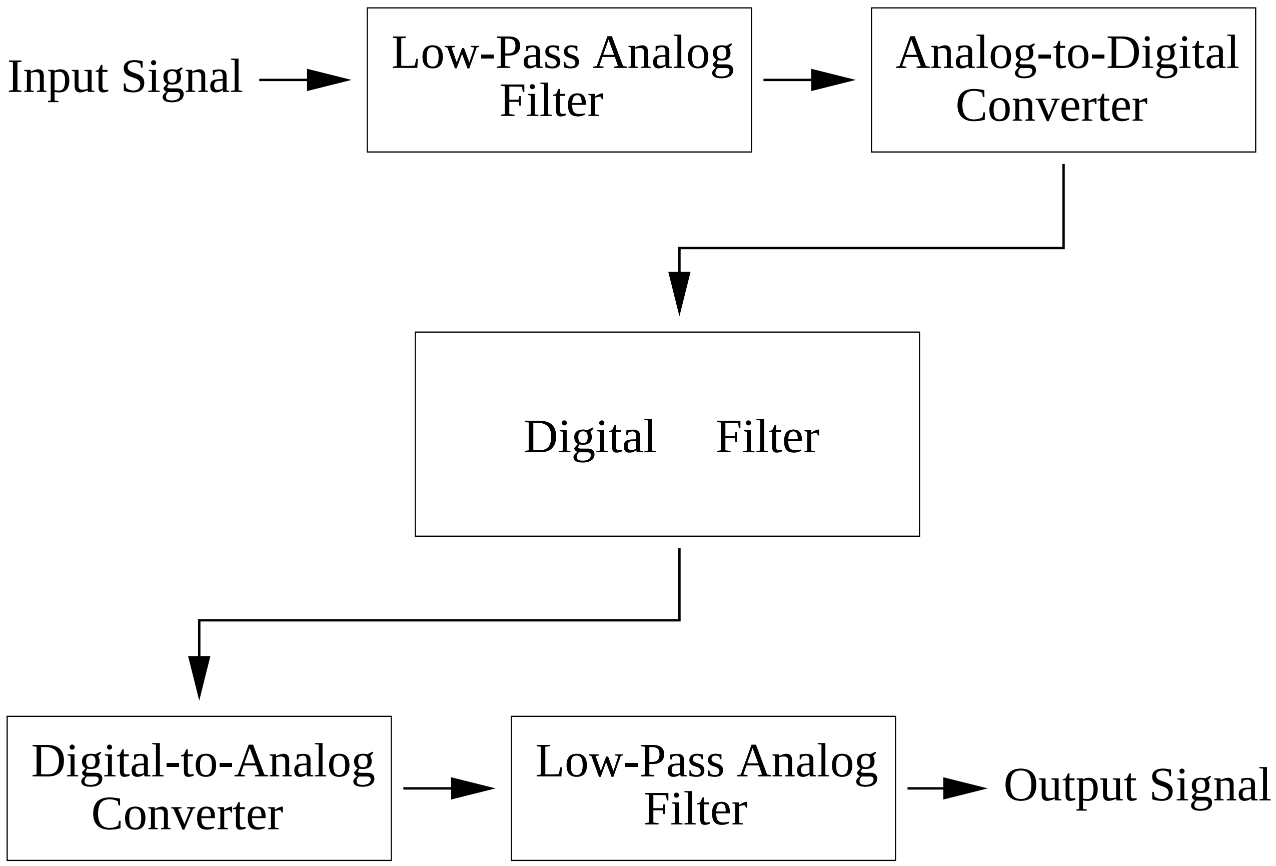
There is a theorem (discussed in Fiume's book) that a signal can be reconstructed from its sampled form as long as all its components have frequencies less that half the sampling frequency.

If we try to digitize a signal at a sampling frequency that is less than twice the maximum frequency present, we may get nonsense!

We can avoid this by putting an analog filter before our analog-to-digital converter.

The same is true in reverse: If we reconstruct an analog signal from a sampled signal, we can't expect the frequencies above half the sampling rate to be reconstructed properly.

Overall Structure of a Digital Filtering System



Impulse Response of a Linear Filter

We can describe a linear filter whose behaviour doesn't change over time by its *impulse response* — what the output signal is when the input is all 0's except for a single 1.

Here's an example of an impulse response:

```
Input:   ... 0,   1,   0,    0,    0,    0 ...
Output:  ... 0, 0.7, 1.3, -0.7, -0.1,    0 ...
```

From linearity, we can deduce the output of this filter for any other input. For example:

```
Input:   ... 0,   1,   2,    0,    0,    0 ...
Output:
         ... 0, 0.7, 1.3, -0.7, -0.1,    0 ...
       + ... 0,   0, 1.4,  2.6, -1.4, -0.2 ...

       = ... 0, 0.7, 2.7,  1.9, -1.5, -0.2 ...
```

Filtering as Convolution

We can also describe the operation of a linear filter as a *convolution*. Suppose the input signal is

$$\ldots,\ P_{-2},\ P_{-1},\ P_0,\ P_1,\ P_2,\ \ldots$$

and the filter has an impulse response of

$$\ldots,\ w_{-2},\ w_{-1},\ w_0,\ w_1,\ w_2,\ \ldots$$

Then the output signal, $\bar{P}_i$, will be given by

$$\bar{P}_i \ = \ \sum_{j=-\infty}^{+\infty} w_{i-j} P_j$$

This is the convolution of P and w, sometimes written as $P * w$. It turns out that $P * w = w * P$.

Of course, we can't compute $\bar{P}$ on a computer if the impulse response extends to $\pm\infty$, but we can if the impulse response is finite (eg, if $w_i = 0$ for $i < L$ and for $i > U$).

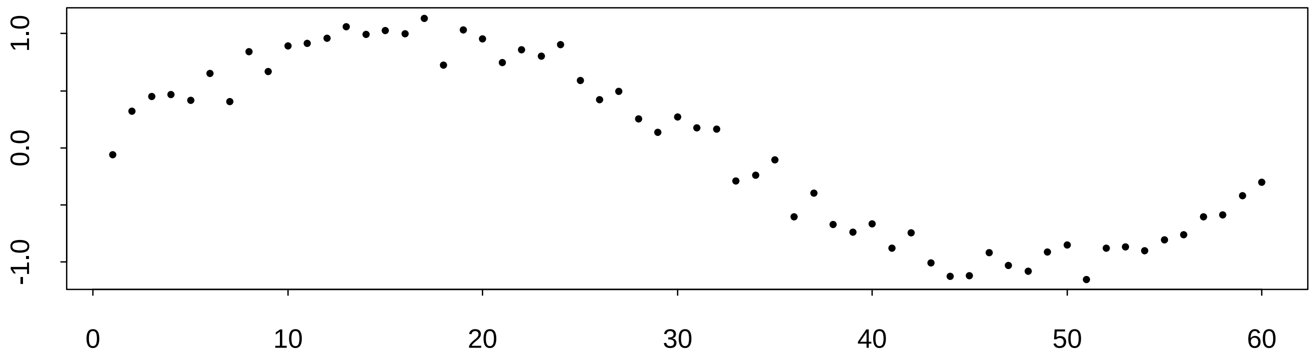
Moving Average Filters

The *n-point moving average* filter has an impulse response of length n, with values of

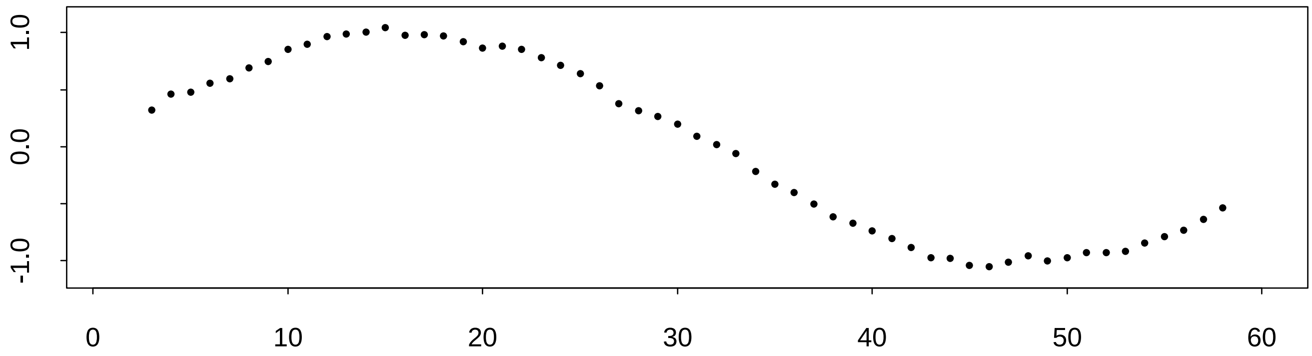
$$1/n,\ 1/n,\ \ldots,\ 1/n,\ 1/n$$

Which is w_0 might vary, but we don't care here.

Here is an example of filtering with a 5-point moving average:



Original Signal



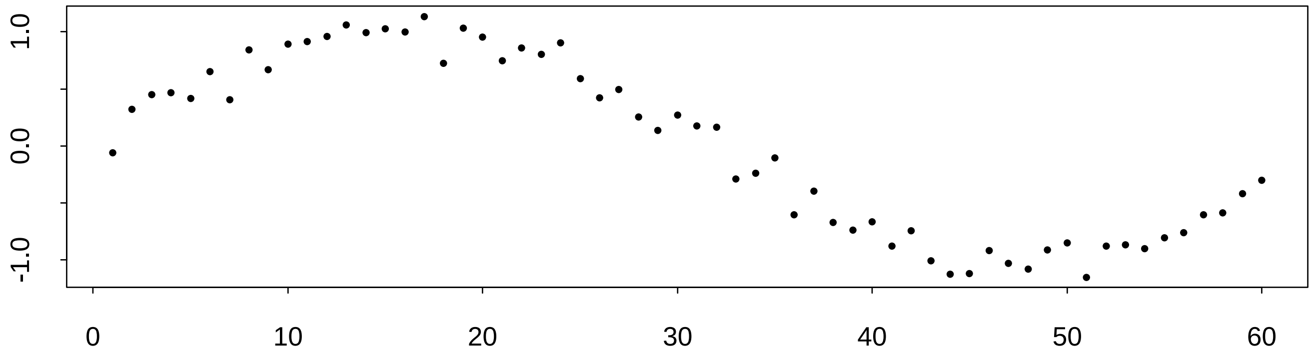
Filtered Signal

This is a low-pass filter, which might be used to suppress noise.

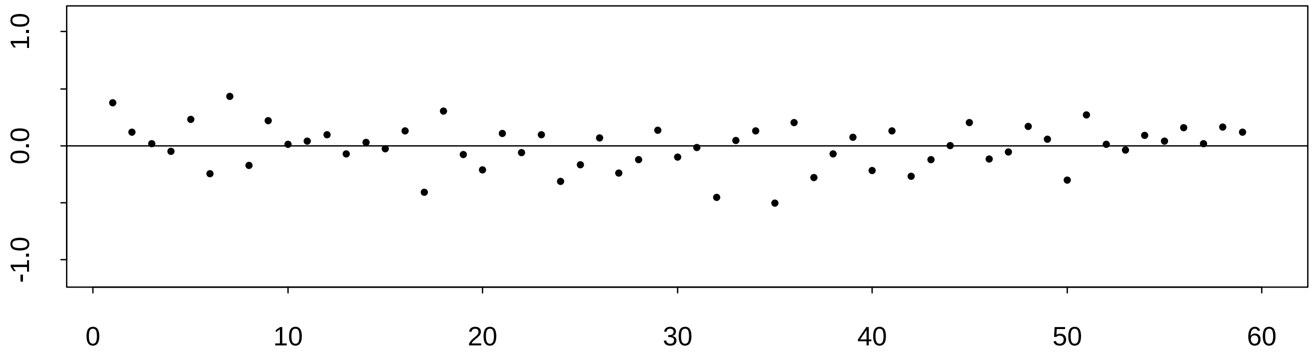
Differencing Filters

We can define a filter that finds the difference between each point and the next. It would have impulse response of $1, -1$.

Here is the result of applying it to the same signal as before:



Original Signal



Filtered Signal

This is a high-pass filter.

What a Filter Does to a Sine Wave

We saw earlier that we can represent functions (eg, signals) as the sum of many sine waves.

We can therefore figure out what a linear filter does to a signal by figuring out what it does to sine waves.

If we apply the filter with impulse response w_0, w_1 to the sine wave $A \sin(2\pi fx + \phi)$, with x being an integer, we get the signal

$$w_0 A \sin(2\pi fx + \phi) \;+\; w_1 A \sin(2\pi f(x-1) + \phi)$$

$$= \quad A_* \sin(2\pi fx + \phi_*)$$

where A_* and ϕ_* depend on w_0, w_1, A, f, and ϕ (but not on x).

For this and any other linear filter, the effect on a sine wave is to convert it to another sine wave, with the same frequency, but maybe with different amplitude or phase.

Frequency Response of a Filter

We can describe what a filter does by saying how it affects the amplitude and phase of sine waves of each frequency. This is the "frequency response" of the filter.

A low-pass filter will reduce the amplitude of sine waves with high frequencies.

A high-pass filter will reduce the amplitude of sine waves with low frequencies.

Filters can also change the phase (perhaps differently for different frequencies). This often doesn't matter, however. People can't hear such phase changes when natural audio signals are filtered.

Filtering With Fourier Transforms

The frequency response of a filter is the basis of another way we could apply it using a computer:

- Convert he signal to a sum of lots of sine waves. (This is called finding the "Fourier transform" of the signal.)
- Change the amplitude and phase of each of these sine waves according to the filter's frequency response.
- Add together the modified sine waves to get the filtered signal. (This is the "inverse Fourier transform".)

This sounds like a lot of work, but because of a clever algorithm called the Fast Fourier Transform it can actually be faster than applying the filter by doing the convolution in the obvious way.

Two-Dimensional Filters

We can also apply filters to images, using a two-dimensional analog of convolution.

For example, we could try to suppress noise by applying a moving average filter with the following pattern:

$$\begin{array}{ccc} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{array}$$

There are two-dimensional analogs of the frequency response and of the Fast Fourier Transform.

See page 153 of Fiume's book for examples of applying filters to an image.