Filters Explained: At least a small part I didn't understand

Gary Bishop

22 June, 2000

1 Introduction

This is a short note, mostly to myself about the time domain nature of filters. It has bothered me for years that while I felt I had a good intuition for the time-domain description of low-pass filters, I had no intuition at all about band-pass and high-pass filters.

The insights that came to me recently are probably obvious to anyone who has really studied this stuff but they sure weren't obvious to me until recently.

2 Low-pass Filters

Low-pass filters are really simple. For example, just averaging together adjacent samples is a low-pass filter, albeit a poor one. Let's consider this simplest of filters.

For all of the following examples I'm going to assume a sample rate of 10000 samples per second and I'm going to make my low-pass filter kernel 1000 samples wide. This means the cut-off frequency (or highest frequency in the pass-band) should be something less than 10Hz.

A little bit fancier way of thinking about averaging adjacent samples together is convolution with a box filter. A box filter simply has a constant weighting for all the samples within its width (so-called support). The Fourier transform of the box filter is a sinc function as shown in Figure 1. We can see that the half amplitude frequency is about 6Hz and the half power frequency is about 4.4Hz. The first zero occurs at 10Hz. The humps beyond 10Hz are what makes the box filter a "poor" one.

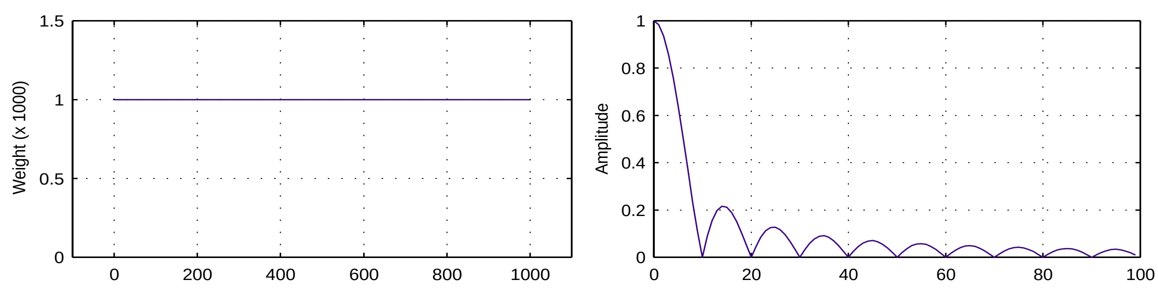

Figure 1: Box filter and its frequency response.

A better low-pass filter is the raised cosine, or Hanning window as the signal processing people call it. A Hanning window that is 1000 samples wide is shown in Figure 2 along with its Fourier transform. We can see from the FT that half amplitude occurs at about 10Hz and half power occurs at about 7Hz. The response beyond 20Hz is much less for this filter than for the box. We traded a somewhat wider pass-band for lower stop-band response.

Things to note about low-pass filters:

- Lower cut-off frequency requires wider support and longer delay.
- Lower stop-band response produces a higher cutoff frequency for the same width support.

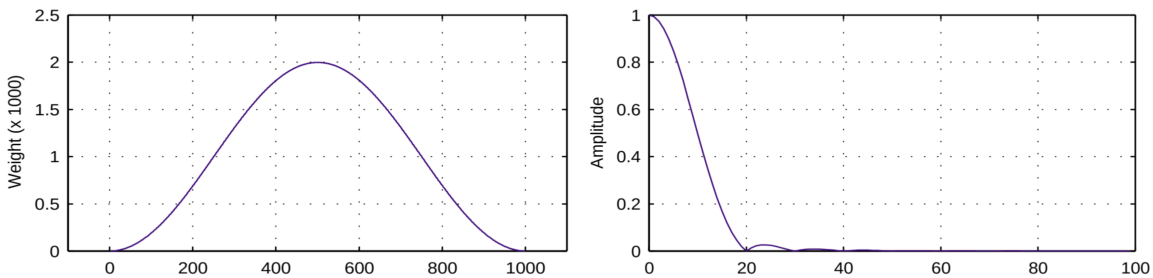

Figure 2: Hanning window and its frequency response

- These symmetric FIR (finite-impulse response) filters produce so called “linear phase”. That is, they introduce a fixed delay for all frequencies.

3 Band-pass filters

The previous, very simple stuff, I have known for years. But what about band-pass filters? Is it possible to have some intuition about the width (number of samples) and weights for a bandpass filter? I could never figure it out.

Then it hit me. Filter designers commonly design band-pass and high-pass filters by first designing a low-pass filter and then transforming it into the filter desired. I can **reason** in the same way.

Suppose I want to convert my low-pass box filter to a band-pass filter with a 200Hz center frequency. In the frequency domain, I want to shift the response up to 200Hz. I could do this in the frequency domain by convolving the filter response with the FT of a 200Hz sine wave which is just a delta at 200Hz. Of course, we know that convolution in the frequency domain is just multiplication in the time domain. So, if I multiply my low-pass filter kernel by a sine wave of the desired center frequency, the resulting filter kernel will be a band-pass filter.

Figure 3 shows the filter kernel and frequency response that results from multiplying the low-pass box kernel by a 200Hz sine wave. Since the low-pass filter was just a box, the band-pass filter is simply 20 cycles (0.1 second) of the 200Hz sine wave. **Wow!** Notice that the band-pass filter's bandwidth is twice that of the low-pass filter on which it was based. The half-amplitude bandwidth is about 12Hz and the half-power bandwidth is a little less than 9Hz. This is caused by the reflection of the low-pass response around the center frequency.

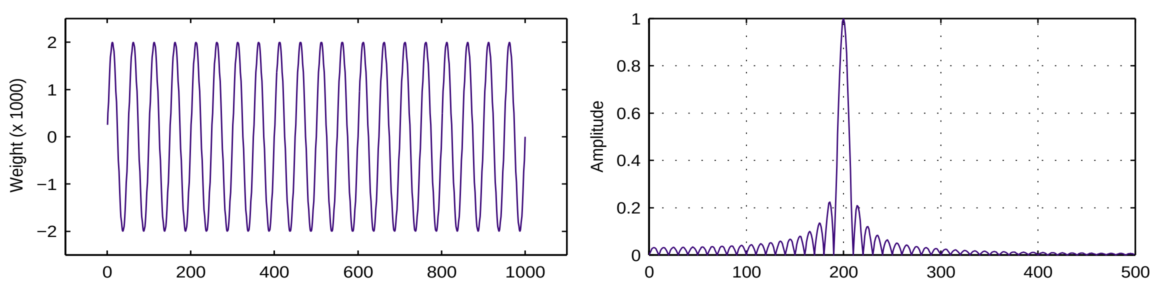

Figure 3: Box band-pass filter kernel and frequency response

Similarly, the Hanning window can be changed into a band-pass filter by multiplying by the center frequency. Figure 4 shows the result. As with the low-pass filters, the Hanning filter has much less response outside the pass-band.

Things to note about band-pass filters:

- Narrowing the pass-band requires widening the support and thus the delay.

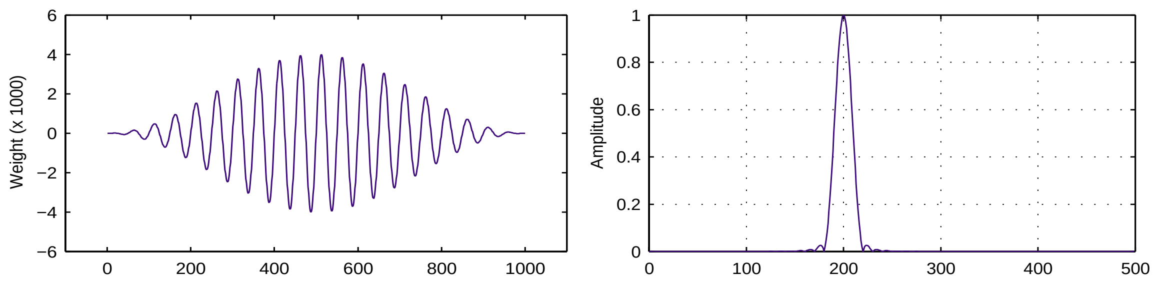

Figure 4: Hanning band-pass filter kernel and frequency response

- The pass-band is twice as wide as a low-pass filter of the same support.
- Lower stop-band response requires a wider pass-band for the same support.
- How does the phase of the sine wave we used to convert the low-pass to a high-pass affect the operation of the resulting band-pass filter?

4 Multiple Pass-bands: Filtering Pulse Waveforms

If we wanted to build a filter that passed multiple frequencies we could just add several of the filter kernels together. Thus we could produce a filter that passed 200 and 400Hz but nothing else just by adding two kernels together. But, if we use the same low-pass prototype for both filters, we could add the sine waves together and then multiply by the prototype.

Of course, we can construct any repetitive waveform from a summation of sine waves with differing amplitudes and phases. So, we can transform a low-pass filter prototype into a filter that passes a particular waveform best.

Here is a concrete example. I want to process the output of a LEPD sensor observing an LED that is pulsed at 100Hz with a duty cycle of 1%. In other words the LED is on for 1 sample and off for 99 samples at 10000 samples per second. The fundamental frequency of this waveform is 100Hz and it has all the harmonics with equal magnitudes. First, I subtract off the mean of the waveform to eliminate response at DC. It's spectrum now should include only the fundamental and harmonics. Then I multiply this by the window function. I haven't worked out yet, how to handle the scaling. If I do this for the Hanning window I get the results shown in Figure 5.

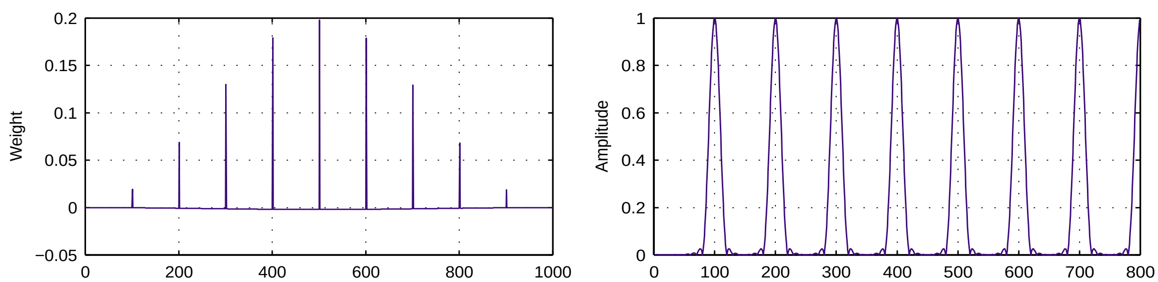

Figure 5: Hanning pulse-pass filter kernel and frequency response

If the waveform consists of frequencies of varying magnitudes, then it needs to be reshaped somehow to pass all the frequencies without changing their amplitudes. Just multiplying the waveform by the window as I described above won't cut it if all the sines are of equal magnitude. What you'll get is sort of a windowed correlator with maximum response when it matches the input signal but it won't necessarily reproduce the input signal. I haven't figured out how to automatically and robustly normalize the waveforms transform in this way.

In our existing tracker systems we have used “dark-light-dark” to eliminate the offset and some measure of background noise. This corresponds to a filter kernel having weights -0.5, 1, -0.5. The frequency response of this kernel is shown in Figure 6.

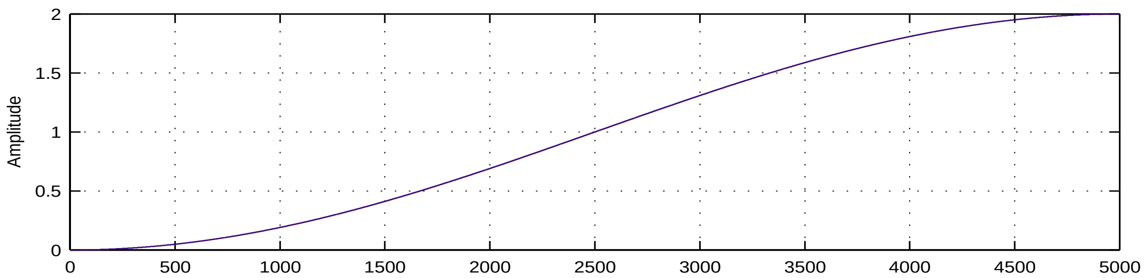

Figure 6: Dark-light-dark frequency response

I tested this filter with a synthesized input signal consisting of the pulse train plus an offset of 0.4 and a normally distributed noise signal with standard deviation equal 0.1. The pulse train is off for the first 2 seconds of the simulation and on for the last 2 seconds. In Figure 7 I plotted only every 100th sample because this is the sample we'd use to determine LED visibility. You can see from the plot that the pulse filter reduces the noise but at the cost of introducing delay.

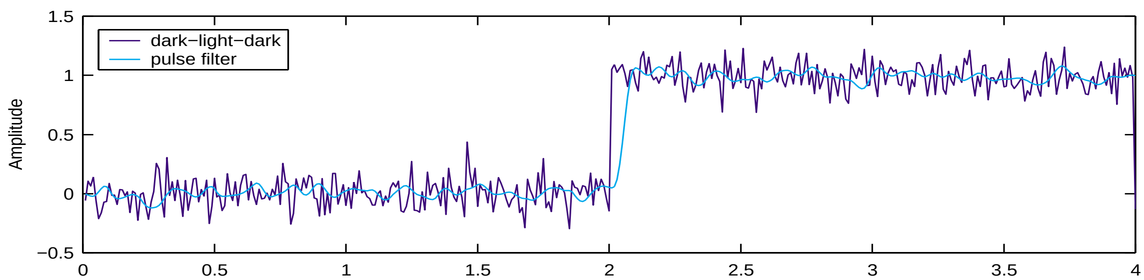

Figure 7: Filters for LEPD output

Things to note about these pulse filters:

- They are just like the band-pass filters with respect to bandwidth and support.
- It is interesting the think about the efficiency of this. In the presence of noise, this pulse filter passes much more noise to the output than a filter for a single frequency would. How does the signal-to-noise ratio compare? Suppose you drove the LED with a sine having the same average power as the pulse, would the SNR be better using a single frequency filter? If we assume white noise, the amount of noise passing through the filter is proportional to the bandwidth of the filter. Our 100Hz pulse consists of a 100Hz fundamental and all the integer harmonics up to 5000Hz (half our sampling rate). The energy should be uniformly distributed across the harmonics. The noise in the various bands will be uncorrelated and thus their variances should add, while the signal should add coherently. This implies, I think, that we get an improvement in the signal-to-noise ratio of square-root of the number of passbands.

5 High-pass filters

What about high-pass filters? Filter designers use an inversion to convert a low-pass prototype to a high-pass filter. I don't see how to do that but I can still think about making a high-pass filter from a low-pass filter.

The key is to subtract the low-pass kernel from an all-pass filter kernel. The all-pass is just a delta centered in the window. So, the resulting high-pass filter will be the same width as the low-pass filter from which it is constructed. The low-pass kernel must be normalized so the sum of its weights is 1 before the subtraction. The point is to have the area of the negative part equal to the area of the positive part (the delta). Figure 8 shows a high-pass filter kernel constructed from the Hanning-window low-pass prototype and the resulting frequency response. The center spike in the kernel on the left goes up to 1000, I zoomed in so we could see the shape of the negative part.

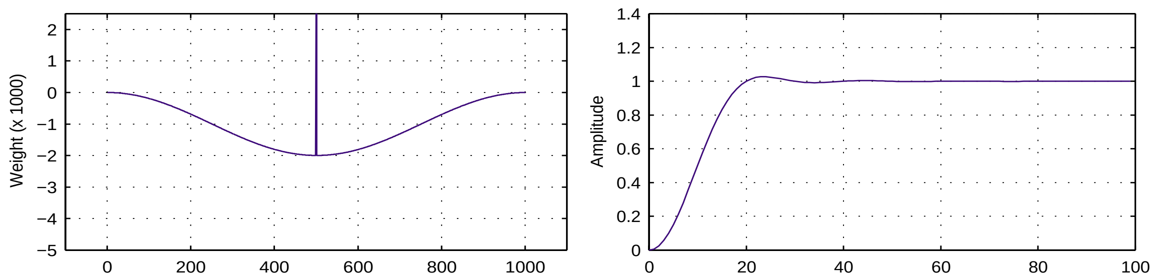

Figure 8: High-pass filter kernel and frequency response.

The half amplitude frequency for the resulting filter is about 10Hz and the half power frequency is about 13Hz. This seems to make sense from reflecting the low-pass filter about the half amplitude frequency.

Things to note about the high-pass filters:

- Narrowing the stop-band requires widening the support.

6 Band-stop filters?

I assume it should be possible to construct a band-stop filter from a band-pass filter using the same subtraction strategy used to construct the high-pass filter. I haven't tried that yet.