Derivation of the Matched Filter

Jaeden Bardati

February 2025

This homework will guide the student through the derivation of the matched filter, a very important technique used in all kinds of experimental physics (including LIGO gravitational waveform fitting).

1 Relevant Mathematical Background

Recall that the **inner product** of $a(t)$ and $b(t)$ is defined as

$$\langle a|b\rangle = \int_{-\infty}^{\infty} a^*(\tau)b(\tau)d\tau \tag{1}$$

Also, recall that the **convolution** of $a(t)$ and $b(t)$ is defined as

$$(a * b)(t) = \int_{-\infty}^{\infty} a(\tau)b(t-\tau)d\tau \tag{2}$$

Similarly, the **cross-correlation** of $a(t)$ and $b(t)$ is defined as

$$R_{ab}(t) = \int_{-\infty}^{\infty} a^*(\tau - t)b(\tau)d\tau \tag{3}$$

where the star in the superscript denotes a complex conjugate.[1] The **auto-correlation** function of $a(t)$ is simply defined as $R_a(t) \equiv R_{aa}(t)$.

(a) Show that for real functions, $R_{ab}(t) = a(t) * b(-t)$. If we wanted to implement cross-correlation in practice, what could we do instead?

(b) Visually compare the inner product vs cross-correlation defintions. What do you notice? Why is the cross-correlation sometimes called the "sliding inner product"?

(c) Show that the auto-correlation function is Hermitian. That is, $R_a^*(t) = R_a(-t)$.

[1] Note that the cross-corelation function is sometimes denoted by a five-pointed star $(a \star b)(t)$. However, this is very easy to confuse with the six-pointed star that denotes convolution, so we will use the $R_{ab}(t)$ notation here to avoid this confusion.

2 Guided Derivation

Matched filters are used extensively in signal analysis, particularly when you have a known signal that you want to find in a very noisy data set. The basic idea behind it is that you compare a template signal to your data $x(t)$ which contains some true signal $s(t)$ that you want to pick out with your template and some noise $n(t)$, such that the data you observe is $x(t) = s(t) + n(t)$. The goal is to find some way to "match" the template signal to the particular signal that we are looking for.

To do this, we will do some convolution operation on the given data $x(t)$ with some filter $h(t)$ to get a new, convolved output $y(t) = (x * h)(t)$ which somehow distinguishes where the signal is in the data. We want to "match" the filter $h(t)$ to any given $s(t)$. The way to do this is to choose the filter $h(t)$ that maximizes the signal-to-noise ratio (SNR) at a given t.

We will start the derivation by noting that we can split the output into a signal and noise part. Namely, $y(t) = (s * h)(t) + (n * h)(t) \equiv y_s(t) + y_n(t)$.

(d) Show that $y_s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)S(\omega)e^{i\omega t}d\omega$, where $H(\omega)$ and $S(\omega)$ are the Fourier transforms of $h(t)$ and $s(t)$, respectively. *Hint: Recall that convolution in the time domain is equivalent to multiplication in the frequency domain.*

Now, we will consider minimizing the signal to noise ratio (SNR) at some time t_0. This is simply the ratio of the power of output that is due to the signal compared to the averaged power of output that is noise. Namely,

$$\text{SNR}(t_0) = \frac{|y_s(t_0)|^2}{E\{|y_n(t)|^2\}} \tag{4}$$

Using the Wiener-Khinchin theorem, we can write the expectation value of $|n(t)|^2$ as

$$E\{|n(t)|^2\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_n(\omega)d\omega \tag{5}$$

where $S_n(\omega)$ is the power spectral density of the noise.

(e) Using the above expression, rewrite the SNR as

$$\text{SNR}(t_0) = \frac{1}{2\pi}\frac{|\int_{-\infty}^{\infty} H(\omega)S(\omega)e^{i\omega t}d\omega|^2}{\int_{-\infty}^{\infty}|H(\omega)|^2 S_n(\omega)d\omega} \tag{6}$$

Hint: The expectation value here acts only on $n(t)$, not $h(t_0)$.

(f) Using the Cauchy-Schwarz inequality $|\langle a|b\rangle|^2 \leq \langle a|a\rangle\langle b|b\rangle$, show that the signal to noise can be given the upper bound

$$\text{SNR}(t_0) \leq \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{|S(\omega)|^2}{S_n(\omega)} d\omega \tag{7}$$

Thus, the $h(t)$ that maximizes the SNR will be the one where the SNR equals this bound.

(g) Show that the upper bound is met if we set $H(\omega) = Ae^{i\omega t_0}\frac{S^*(\omega)}{S_n(\omega)}$ for an arbituary constant A.

Now we will consider white noise (i.e. uncorrelated, zero mean noise) and a real signal $s(t)$. Doing so, we can simplify this to $H(\omega) = e^{i\omega t_0}S(-\omega)$.

(h) Plug this expression for $H(\omega)$ back into $y_s(t)$ and show that it is just a a simple autocorrelation function $y_s(t) = R_s(t)$. Hint: Use a change of variable $t - t_0 \to t$.

(i) Therefore, show that $y(t) = x(t) * s(-t)$ and thus infer what $h(t)$ equals.

To recap, we found that the filter $h(t)$ required to maximize the signal to noise ratio (SNR; the power of the signal $s(t)$ to the power of the noise $n(t)$ in the observed data) in the convolution of the observed data $x(t)$ with that filter, is simply the time-inversion of the signal we expect to see somewhere in the data. Another way of saying this is that, to extract the location of a template in a dataset, we compute the cross-correlation of the template with that dataset. If we divide this by the noise estimate, we get the signal to noise of our template in the dataset. An observed signal that appears like the template in the dataset will appear as a peak in this SNR plot. You can apply this with something like `numpy.correlate` in Python. You will do this in the corresponding Jupyter notebook part of this homework.