

MATCHED FILTER^{*}

Michael Lawrence

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 1.0[†]

^{*}Version 1.3: Dec 21, 2004 2:48 am -0600

[†]<http://creativecommons.org/licenses/by/1.0>

1 Matched Filter

The relevant portion of the block diagram

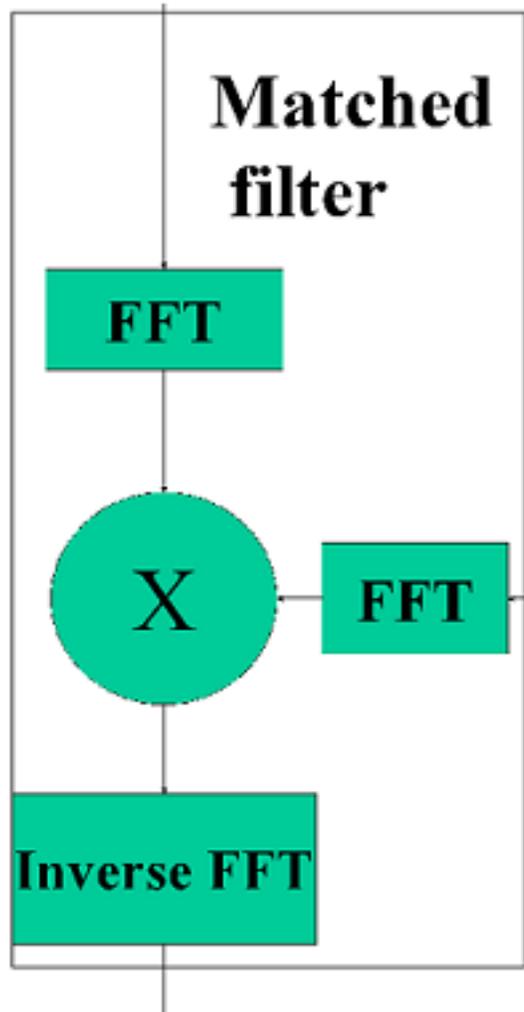


Figure 1: The Matched Filter.

The matched filter segment of our project is perhaps the easiest to code but the most difficult conceptually. An understanding of Fourier analysis, convolution and manipulations in frequency domain is required. Although we made available several supplementary links at the beginning of the course, a brief recapitulation of the basic theory is warranted.

1.1 Fourier Analysis

Jean Baptiste Joseph Fourier¹ discovered, approximately 200 years ago, that any function could be represented by weighted averages of any periodic base function. The most common application of Fourier analysis is through the use of sinusoidal waves, but using a sinusoid as a basis is not necessary - any periodic function will do. This weighing was perceived as a series of coefficients, each coefficient corresponding to a specific frequency of the periodic base function. From consideration of these coefficients, it was realized there existed a **frequency domain**. Thus, there was a medium in which discussion of a non-periodic function's frequency made sense. This discovery has had many impressive results, not the least of which is the concept of a matched filter.

Carl Friedrich Gauss² later expounded on Fourier's realization and found what is the now-known fastest algorithm for computing Fourier coefficients. Unfortunately for 20th century scientists, no one remembered Gauss developed the algorithm until **after** it had been rederived.

1.2 Convolution

Convolution is at the heart of matched filters. One can easily visualize placing a transparency of an image over another image to gauge the similarity of the two images. Convolution does essentially just that with two functions; it places one function over another function and outputs a single value suggesting a level of similarity, then it moves the first function over an infinitesimally small distance and finds another value. The end result is a graph which peaks at the point where the two images are most similar.

Another method of attack may be seen through the Cauchy-Schwarz Inequality. This³ page does an enviable job of approaching matched filtering from that angle.

The concept that binds these two seemingly disparate topics is almost startling in its beauty (for those of us who have ever had to perform convolution integrals). Convolution in the time domain (the domain in which we would want to compare two signals) just happens to be multiplication in the frequency domain. And, as it turns out, performing two Fast Fourier Transforms (FFTs), multiplying the results, and then performing two Inverse FFTs (IFFTs) is computationally faster than performing one convolution between the two signals and provides the same result.

2 Summary

A matched filter compares two signals and outputs a function describing the places at which the two signals are most like one another. This is accomplished through FFTing two signals, multiplying their coefficients and IFFTing the result.

¹http://en.wikipedia.org/wiki/Jean_Baptiste_Joseph_Fourier

²http://en.wikipedia.org/wiki/Carl_Friedrich_Gauss

³"Cauchy-Schwarz Inequality" <<http://cnx.org/content/m10757/latest/>>