

PRACTICAL ISSUES IN WIENER FILTER IMPLEMENTATION*

Douglas L. Jones

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

The weiner-filter, $W_{\text{opt}} = R^{-1}P$, is ideal for many applications. But several issues must be addressed to use it in practice.

Exercise 1 *(Solution on p. 3.)*

In practice one usually won't know exactly the statistics of x_k and d_k (i.e. R and P) needed to compute the Wiener filter.

How do we surmount this problem?

Exercise 2 *(Solution on p. 3.)*

In many applications, the statistics of x_k, d_k vary slowly with time.

How does one develop an **adaptive** system which tracks these changes over time to keep the system near optimal at all times?

Exercise 3 *(Solution on p. 3.)*

How can $r_{xx}^k(l)$ be computed efficiently?

Exercise 4

how does one choose N ?

1 Tradeoffs

Larger $N \rightarrow$ more accurate estimates of the correlation values \rightarrow better W_{opt} . However, larger N leads to slower adaptation.

NOTE: The success of adaptive systems depends on x, d being roughly stationary over at least N samples, $N > M$. That is, all adaptive filtering algorithms require that the underlying system varies slowly with respect to the sampling rate and the filter length (although they can tolerate occasional step discontinuities in the underlying system).

*Version 1.1: Feb 14, 2004 3:54 pm -0600

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

2 Computational Considerations

As presented here, an adaptive filter requires computing a matrix inverse at each sample. Actually, since the matrix R is Toeplitz, the linear system of equations can be solved with $O(M^2)$ computations using Levinson's algorithm, where M is the filter length. However, in many applications this may be too expensive, especially since computing the filter output itself requires $O(M)$ computations. There are two main approaches to resolving the computation problem

1. Take advantage of the fact that R^{k+1} is only slightly changed from R^k to reduce the computation to $O(M)$; these algorithms are called Fast Recursive Least Squares algorithms; all methods proposed so far have stability problems and are dangerous to use.
2. Find a different approach to solving the optimization problem that doesn't require explicit inversion of the correlation matrix.

NOTE: Adaptive algorithms involving the correlation matrix are called **Recursive least Squares** (RLS) algorithms. Historically, they were developed after the LMS algorithm, which is the simplest and most widely used approach $O(M)$. $O(M^2)$ RLS algorithms are used in applications requiring very fast adaptation.

Solutions to Exercises in this Module

Solution to Exercise (p. 1)

Estimate the statistics

$$r_{xx}(l) \simeq \frac{1}{N} \sum_{k=0}^{N-1} x_k x_{k+l}$$

$$r_{xd}(l) \simeq \frac{1}{N} \sum_{k=0}^{N-1} d_k x_{k-l}$$

then solve $W_{\text{opt}} = R^{-1} = P$

Solution to Exercise (p. 1)

Use short-time windowed estimates of the correlation functions.

NOTE:

$$\left(r_{xx}(l) \right)^k = \frac{1}{N} \sum_{m=0}^{N-1} x_{k-m} x_{k-m-l}$$

$$\left(r_{dx}(l) \right)^k = \frac{1}{N} \sum_{m=0}^{N-1} x_{k-m-l} d_{k-m}$$

and $W_{\text{opt}}^k \simeq \left(R_k \right)^{-1} P_k$

Solution to Exercise (p. 1)

Recursively!

$$r_{xx}^k(l) = r_{xx}^{k-1}(l) + x_k x_{k-l} - x_{k-N} x_{k-N-l}$$

This is critically stable, so people usually do

$$(1 - \alpha) \left(r_{xx}^k(l) \right) = \alpha r_{xx}^{k-1}(l) + x_k x_{k-l}$$