1 Lecture Outline

Reading: Chapter 13 in Quateri

Lectures are based on Chapter 6 in Loizou

- Introduction
- Time-Domain Derivation of the Wiener Filter problem
- Frequency-Domain Derivation of the Wiener Filter problem
- Wiener Filters for Noise Reduction
- Iterative Wiener Filter

2 Introduction

While the idea and parameters for the spectral subtraction methods were essentially found heuristically, Wiener filters can be derived as a solution to an optimization problem. Wiener filters are the optimal filters for minimizing the mean-square error between a given and desired signal.

3 Introduction to Wiener Filter Theory

The basic Wiener filtering problem is shown Fig. 1. The goal is to design a system such that the filter output $\hat{d}[n]$ is close to our desired signal $d[n]$. We constrain filter to be linear and FIR (stable and easy to evaluate). For now, don't think about speech enhancement yet! Let's just solve the Wiener filtering problem.

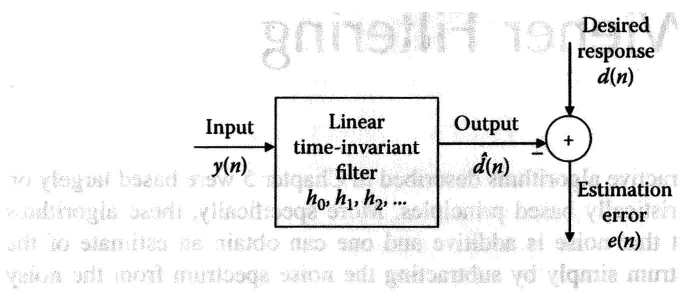

FIGURE 6.1 Block diagram of the statistical filtering problem.

Figure 1: Basic Wiener Filtering Problem.

Assuming a length M FIR system, the filter output is

$$\begin{aligned}\hat{d}[n] &= \sum_{k=0}^{M-1} h_k y[n-k] \\ &= \mathbf{h}^T \mathbf{y}[n] \qquad (1)\end{aligned}$$

where the vector of filter coefficients

$$\mathbf{h} = [h_0, h_1, h_2, ..., h_{M-1}]^T \qquad (2)$$

and the input sample regressor (most recent M samples)

$$\mathbf{y}[n] \quad = \quad [y[n], y[n-1], ..., y[n-M+1]]^T \tag{3}$$

Our goal is to compute $\mathbf{h}$ such that the mean square error (MSE), $E\left\{e^2[n]\right\} = E\left\{(d[n]-\hat{d}[n])^2\right\}$, is minimized.

4 Wiener Filter Derivation in the Time Domain

For the derivations that follow, it is assumed that the input signal, $y[n]$, and the desired signal, $d[n]$ are realizations of jointly wide-sense stationary processes. We define the estimation error,

$$\begin{aligned} e[n] &= d[n] - \hat{d}[n] \\ &= d[n] - \mathbf{h}^T\mathbf{y}[n]. \end{aligned} \tag{4}$$

The optimal filter coefficients are found by minimizing the mean-square of the error $e[n]$, $E\{e^2[n]\}$:

$$\begin{aligned} J &= E\{e^2[n]\} \\ &= E\{(d[n] - \mathbf{h}^T\mathbf{y}[n])^2\} \\ &= E\{d^2[n]\} - 2\mathbf{h}^T E\{\mathbf{y}d[n]\} + \mathbf{h}^T E\{\mathbf{y}\mathbf{y}^T\}\mathbf{h} \\ &= E\{d^2[n]\} - 2\mathbf{h}^T\mathbf{r_{yd}} + \mathbf{h}^T\mathbf{R_{yy}}\mathbf{h} \end{aligned} \tag{5}$$

where $\mathbf{r_{yd}}$ is the cross-correlation vector between the input and desired signal and $\mathbf{R_{yy}}$ is the autocorrelation matrix of the input signal.

The cost function J is a quadratic function in $\mathbf{h}$ and therefore has a single global minimum value. If we take the derivative of J as a function of the vector $\mathbf{h}$ we get

$$\frac{\partial J}{\partial \mathbf{h}} = -2\mathbf{r_{yd}} + 2\mathbf{h^T}\mathbf{R_{yy}}. \tag{6}$$

Equating (6) to zero gives

$$\mathbf{R_{yy}}\mathbf{h} = \mathbf{r_{yd}} \tag{7}$$

The optimal or Wiener filter is the solution to (7). Because of the wide-sense stationarity assumption, the autocorrelation matrix $\mathbf{R_{yy}}$ is positive definite and invertible yielding as the optimal solution or Wiener filter

$$\mathbf{h_{opt}} = \mathbf{R_{yy}^{-1}}\mathbf{r_{yd}}. \tag{8}$$

By substituting the Wiener filter into (5), we can find the minimum cost

$$\begin{aligned} J_{min} &= E\{d^2[n]\} - 2\left(\mathbf{R_{yy}^{-1}}\mathbf{r_{yd}}\right)^T\mathbf{r_{yd}} + \left(\mathbf{R_{yy}^{-1}}\mathbf{r_{yd}}\right)\mathbf{R_{yy}}\mathbf{R_{yy}^{-1}}\mathbf{r_{yd}} \\ &= E\{d^2[n]\} - 2\mathbf{r_{yd}^T}\mathbf{R_{yy}^{-T}}\mathbf{r_{yd}} + \mathbf{r_{yd}^T}\mathbf{R_{yy}^{-T}}\mathbf{r_{yd}} \\ &= E\{d^2[n]\} - \mathbf{r_{yd}^T}\mathbf{R_{yy}^{-T}}\mathbf{r_{yd}} \\ &= E\{d^2[n]\} - \mathbf{r_{yd}^T}\mathbf{h_{opt}} \end{aligned} \tag{9}$$

5 Wiener Filters Derivation in the Frequency Domain

Suppose that we now allow for two-sided (noncausal) , infinite duration filters such that

$$\hat{d}[n] = \sum_{k=-\infty}^{\infty} h[k]y[n-k] \quad -\infty < n < \infty \tag{10}$$

This filter is called the *Wiener smoother filter*. Of course, $\hat{d}[n]$ is just the convolution of $h[n]$ and $y[n]$, which can be expressed as

$$\hat{d}[n] = h[n] * y[n]. \tag{11}$$

In the frequency-domain, we have

$$\hat{D}(\omega) = H(\omega)Y(\omega). \tag{12}$$

We define the estimation error in the frequency-domain as

$$\begin{aligned} E(\omega) &= D(\omega) - \hat{D}(\omega) \\ &= D(\omega) - H(\omega)Y(\omega). \end{aligned} \tag{13}$$

The cost function (expected squared-error) to minimize is given by

$$\begin{aligned} J &= E\{|D(\omega) - \hat{D}(\omega)|^2\} \\ &= E\{[D(\omega) - H(\omega)Y(\omega)]^* \, [D(\omega) - H(\omega)Y(\omega)]\} \\ &= E\{|D(\omega)|^2\} - H(\omega)\underbrace{E\{Y(\omega)D^*(\omega)\}}_{S_{yd}} - \\ &\quad H^*(\omega)\underbrace{E\{D(\omega)Y^*(\omega)\}}_{S_{dy}} + |H(\omega)|^2 \underbrace{E\{|Y(\omega)|^2\}}_{S_{yy}} \end{aligned} \tag{14}$$

where $S_{yy}(\omega) = E\{|Y(\omega)|^2\}$ is power spectral density of y and $S_{yd}(\omega) = E\{Y(\omega)D^*(\omega)\}$ is cross-power spectral density of y and d.

In order to find the optimal $H(\omega)$ we need to calculate the derivative of J with respect to $H(\omega)$. Note that this is a complex derivative:

$$\begin{aligned} \frac{\partial J}{\partial H(\omega)} &= H^* S_{yy}(\omega) - S_{yd}(\omega) \\ &= [H(\omega)S_{yy}(\omega) - S_{dy}]^* = 0 \end{aligned} \tag{15}$$

If we use the fact that $S_{yd}(\omega) = S^*_{dy}(\omega)$ we can solve for $H(\omega)$

$$H(\omega) = \frac{S_{dy}(\omega)}{S_{yy}(\omega)} \tag{16}$$

which is the frequency-domain equivalent to the time-domain solution in (8).

6 Wiener Filters for Noise Reduction

6.1 Time-Domain

In speech enhancement, the filter input is the noisy speech signal

$$\begin{aligned} y[n] &= x[n] + d[n] \\ &\updownarrow \\ Y(\omega) &= X(\omega) + D(\omega) \end{aligned} \tag{17}$$

and the "desired" signal is the clean speech, $x[n]$ (notation changed from Wiener filter figure). The objective is to use a Wiener filter to estimate $x[n]$ or $X(\omega)$.

We have

$$\begin{aligned} \mathbf{R_{yy}} &= E\{\mathbf{y}\mathbf{y}^T\} = E\{(\mathbf{x}+\mathbf{d})(\mathbf{x}+\mathbf{d})^T\} \\ &= E\{\mathbf{x}\mathbf{x}^T\} + E\{\mathbf{d}\mathbf{d}^T\} + E\{\mathbf{x}\mathbf{d}^T\} + E\{\mathbf{d}\mathbf{x}^T\} \\ &= \mathbf{R_{xx}} + \mathbf{R_{dd}} \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{r}_{\mathbf{y}x} &= E\left\{\mathbf{y}[n]x[n]\right\} \\ &= E\left\{\mathbf{x}[n] + \mathbf{d}[n]\right\} x[n] \\ &= \mathbf{r}_{\mathbf{x}x} \end{aligned} \tag{19}$$

We note that the cross terms are zero since speech and noise assumed uncorrelated. The Wiener filter for speech enhancement is therefore

$$\mathbf{h_{opt}} = \left(\mathbf{R_{xx}} + \mathbf{R_{dd}}\right)^{-1} \mathbf{r}_{\mathbf{x}x} = \mathbf{R_{yy}}^{-1}\mathbf{r}_{\mathbf{x}x} \tag{20}$$

Unfortunately, we do not have access to $\mathbf{r}_{\mathbf{x}x}$ so this filter is not realizable.

6.2 Frequency-Domain

From the frequency-domain solution to the Wiener filter, we have the following components

$$\begin{aligned} S_{yy}(\omega) &= E\{[X(\omega)+D(\omega)]\,[X(\omega)+D(\omega)]^*\} \\ &= E\{X(\omega)X^*(\omega)\} + E\{D(\omega)D^*(\omega)\} \\ &\quad + E\{X(\omega)D^*(\omega)\} + E\{D(\omega)X^*(\omega)\} \\ &= S_{xx}(\omega) + S_{dd}(\omega) \end{aligned} \tag{21}$$

$$\begin{aligned} S_{xy}(\omega) &= E\{X(\omega)\,[X(\omega)+D(\omega)]^*\} \\ &= E\{X(\omega)X^*(\omega)\} + E\{X(\omega)D^*(\omega)\} \\ &= S_{xx}(\omega) \end{aligned} \tag{22}$$

We note that the cross terms are zero since speech and noise assumed uncorrelated. The Wiener filter for speech enhancement is therefore

$$H_{opt}(\omega) = \frac{S_{xx}(\omega)}{S_{xx}(\omega) + S_{nn}(\omega)} = \frac{S_{xx}(\omega)}{S_{yy}(\omega)}. \tag{23}$$

Unfortunately, we do not have access to $S_{xx}(\omega)$.

7 Iterative Wiener Filtering

In order to utilize a practical Wiener filter for speech enhancement, we consider an iterative approach to estimating

$$H_{opt}(\omega) = \frac{S_{xx}(\omega)}{S_{xx}(\omega) + S_{dd}(\omega)}.$$

In the $(i+1)$ iteration, we estimate the clean speech spectrum

$$\hat{X}_{i+1}(\omega) = H_i(\omega)Y(\omega) \tag{24}$$

where $H_i(\omega)$ denotes Wiener filter estimate in i-th iteration. We use $\hat{X}_{i+1}(\omega)$ for $S_{xx}(\omega)$ to obtain (improved) $H_{i+1}(\omega)$. By iterating a few times (use new output to get new Wiener filter), we have an iterative Wiener filter. Finally, we note that we estimate $S_{dd}(\omega) \approx |X(\omega)|^2$ during non-speech segments.

The algorithm used by the Iterative (Generalized) Weiner filter is given below.

Assume we have an estimate for $S_{dd}(\omega)$.

Step 1: Initialize $H_0(\omega)$

Step 2: Compute the filter output for the i-th iteration

$$\hat{X}_i(\omega) = H_{i-1}(\omega)Y(\omega) \tag{25}$$

Step 3: Compute a smoothed estimate

$$S_{xx}(\omega) \quad = \quad \tau S_{xx}(\omega) + (1-\tau)|\hat{X}_i(\omega)|^2 \tag{26}$$

Step 4: During non-voice activity, update noise power spectrum

$$S_{dd}(\omega) \quad = \quad \tau S_{dd}(\omega) + (1-\tau)|Y(\omega)|^2 \tag{27}$$

Step 5: Use $S_{xx}(\omega)$ from (26) and $S_{dd}(\omega)$ to update filter

$$H_i(\omega) = \frac{S_{xx}(\omega)}{S_{xx}(\omega) + S_{dd}(\omega)}$$

Step 4: Go to Step 2 and repeat a few times