Lecture 5) Wiener Filters

Block diagram of channel filtering problem.

\[
\begin{align*}
\text{Input} & \quad \text{FIR Filter} & \quad \text{Output} & \quad \text{Desired response} \\
\mu(n), \nu(n) \ldots & \quad h_0, h_1, \ldots & \quad y(n) & \quad d(n) \\
& & \downarrow & \quad \text{Estimation error} \\
& & e(n) \quad \text{or} \quad \sigma \quad \text{or} \quad \epsilon
\end{align*}
\]

Notes:
1. Input sequence is complex - general case; Real sequences can be treated as a special case.
2. The filter is linear and discrete time filter; Superposition applies and filter operates on sampled version of the input signal.
3. Desired response of the filter is known - adaptation is closed loop performance feedback adaptation.
4. The performance is formulated in terms of estimation error.

In general LTI filters can be either FIR or IIR. Due to stability concerns, in most usage of practical applications we use FIR.

[Diagram with FIR filter structure]
From Figure, we have

\[ y(n) = \sum_{k=0}^{n-1} w^k u(n-k), \quad n = 0, 1. \]

- Denote complex convolution

If we define \( \mathbf{u}(n) = [u(n) \ u(n-1) \ \ldots \ u(n-H+1)]^T \in \mathbb{C}^{H \times 1} \)

\[ \mathbf{w} = [w_0 \ w_1 \ \ldots \ w_{H-1}]^T \in \mathbb{C}^{H \times 1} \]

Then,

\[ y(n) = \mathbf{w}^* \mathbf{u}(n), \quad H \text{-denote Hermitian transpose.} \]

The estimation error is given by

\[ e(n) = d(n) - y(n) \]

\[ = d(n) - \sum_{k=0}^{n-1} \mathbf{w}^* \mathbf{u}(n-k) \]

\[ = d(n) - \mathbf{w}^* \mathbf{u}(n) \]

As an optimization criterion, we choose to minimize the mean-square value of \( e(n) \).

With this in mind, the optimization cost function is given by

\[ J(\mathbf{w}) = E\{e(n)^* e(n)\} = E\{|e(n)|^2\} \]

\[ = E\{(d(n) - \mathbf{w}^* \mathbf{u}(n))^* (d(n) - \mathbf{w}^* \mathbf{u}(n))\} \]

\[ = E\{d(n)^* d(n) - \mathbf{w}^* \mathbf{d}(n) \mathbf{u}(n)^* \mathbf{u}(n) - d(n)^* \mathbf{u}(n)^* \mathbf{u}(n) + \mathbf{w}^* \mathbf{d}(n) \mathbf{u}(n)^* \mathbf{u}(n) \} \]

\[ = E\{d(n)^* d(n) - \mathbf{w}^* \mathbf{d}(n) \mathbf{u}(n)^* \mathbf{u}(n) - d(n)^* \mathbf{u}(n)^* \mathbf{u}(n) + \mathbf{w}^* \mathbf{d}(n) \mathbf{u}(n)^* \mathbf{u}(n) \} \]

\[ + \mathbf{w}^* E\{\mathbf{d}(n) \mathbf{u}(n)^* \mathbf{u}(n)\} \]
Consider a few by few of the above equation:

\[ \sum_{n=0}^{\infty} w_n d^*(n) = F \int d(n) d^*(n) = F \int d(n)^2 = \frac{1}{2} \text{ of the desired sequence} \]

Define a cross correlation vector:

\[ p = \sum_{n=0}^{\infty} w_n \int d(n) d^*(n) = [E \int u(n) d^*(n)]^T E \int u(n) d^*(n) \]

Then we have:

\[ \sum_{n=0}^{\infty} w_n \int d^*(n) = \sum w_n \cdot p \]

\[ F \int d(n) u^*(n) = E \int u(n) d^*(n) \Rightarrow w = p^H w \]

\[ \sum_{n=0}^{\infty} w_n \int u(n) d^*(n) = w^H R w \]

where \( R = F \int u(n) u^*(n) \) - is the correlation matrix.

Therefore, the cost function can be rewritten as:

\[ J(w) = F d(n) - w^H p - p^H w + w^H R w \]

Notes:
1. \( J(w) \) is a function of \( w \); For different \( w \) we get different values of \( J(w) \).

2. We seek to minimize \( J(w) \) by selecting proper set of weights.

3. From calculus, we know that the minimum is obtained when

\[ \frac{\partial J(w)}{\partial w} = 0 \]
Calculus refresher: Derivative of scalar function of vector argument

Let \( \mathbf{f}(\mathbf{x}): \mathbf{x} \in \mathbb{C}^{N \times 1} \rightarrow \mathbb{C} \)

E.g. \( \mathbf{f} \) maps vector \( \mathbf{x} \) into a scalar. Dimension of \( \mathbf{x} \) is \( N \times 1 \)

Let \( \mathbf{x}_k = a_k + jb_k \), \( k = 0, 1, \ldots, N-1 \)

Then

\[
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix}
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial a_0} \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial b_0} \\
\vdots \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial a_{N-1}} \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial b_{N-1}}
\end{bmatrix}
\]

and

\[
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^*} = \begin{bmatrix}
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_0^*} \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1^*} \\
\vdots \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_{N-1}^*}
\end{bmatrix}
\]

Example 1: Consider a function

\[
\mathbf{f}(\mathbf{x}) = \mathbf{x}^H \mathbf{U} = (\mathbf{U}^T \cdot \mathbf{x}) = \sum_{k=0}^{N-1} x_k^* a_k = \sum_{k=0}^{N-1} (a_k - j b_k) x_k
\]

\[
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^*} = \begin{bmatrix}
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial a_0} \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial b_0} \\
\vdots \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial a_{N-1}} \\
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial b_{N-1}}
\end{bmatrix}
\]

Consider

\[
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial a_k} = \sum_{k=0}^{N-1} (a_k - j b_k) x_k = 2 x_k
\]

\[
\frac{\partial \mathbf{f}(\mathbf{x})}{\partial b_k} = \sum_{k=0}^{N-1} (a_k - j b_k) x_k = -j x_k
\]
Therefore
\[
\sum_{n=0}^{N-1} \sum_{k=0}^{T-1} \left( u_n + j (-j u_n) \right) = \begin{bmatrix} u_0 + u_n \\ u_1 + u_{n+1} \\ \vdots \\ u_{N-1} + u_{n_{N-1}} \end{bmatrix} = 2X
\]

Example 2
\[
\mathbf{p}_2(x) = \mathbf{u}^H \mathbf{x} = \sum_{k=0}^{T-1} u_k^* x_k = \sum_{k=0}^{N-1} u_k^* (a_k + j b_k)
\]

Converse \( \frac{\partial \mathbf{p}_2(x)}{\partial x^H} \)
\[
\frac{\partial \mathbf{p}_2(x)}{\partial u_n} = \sum_{k=0}^{T-1} u_k^* (a_k + j b_k) = 0
\]
\[
\frac{\partial \mathbf{p}_2(x)}{\partial u_{n+1}} = \sum_{k=0}^{T-1} u_k^* (a_k + j b_k) = j u_{n+1}^*
\]

Therefore
\[
\frac{\partial \mathbf{p}_2(x)}{\partial x^H} = \begin{bmatrix} u_0 + j (-j u_0) \\ u_1 + j (-j u_1) \\ \vdots \\ u_{N-1} + j (-j u_{N-1}) \end{bmatrix} = \begin{bmatrix} u_0 - u_0 \\ u_1 - u_1 \\ \vdots \\ u_{N-1} - u_{N-1} \end{bmatrix} = 0
\]

Summary
\[
\frac{\partial \mathbf{p}_1(x)}{\partial x} = x^H \mathbf{u} \quad \frac{\partial \mathbf{p}_1(x)}{\partial x^H} = 2X
\]
\[
\frac{\partial \mathbf{p}_2(x)}{\partial x^H} = 0 \quad \frac{\partial \mathbf{p}_2(x)}{\partial x^H} = 2X
\]
\[
\frac{\partial \mathbf{p}_3(x)}{\partial x^H} = x^H \mathbf{R} \quad \frac{\partial \mathbf{p}_3(x)}{\partial x^H} = 2 \mathbf{R} x
\]
Applying (xx) to the error function given in (x), one obtains

\[ 2 \frac{\partial J}{\partial \omega^x} = \frac{\partial}{\partial \omega^x} \left[ r_d(t) - p^H \omega \right] = 0 \]

or

\[ p \omega = p \]

If the covariance matrix of the input process is nonsingular, the weight vector that minimizes

\[ J(\omega) = E \left[ (u(t))^2 \right] \]

is given by

\[ \omega_{opt} = R^{-1} p \]

where

- \( R \) = correlation matrix of the input vector
- \( P \) = cross correlation vector between input vectors and desired filter response.

Substituting into (x):

\[ J(\omega_{opt}) = r_d(0) - (R^{-1} p)^H \cdot p - p^H (R^{-1} p) + (R^{-1} p)^H \cdot p \cdot (R^{-1} p) \]

\[ = r_d(0) - p^H (R^{-1} p)^H \cdot p - p^H R^{-1} p + p^H (R^{-1} p)^H \cdot R^{-1} p \]

It is easy to prove that \((R^{-1})^H = R^{-1}\). Therefore

\[ J(\omega_{opt}) = r_d(0) - p^H R^{-1} p - p^H R^{-1} p + p^H R^{-1} p \]

\[ = r_d(0) - p^H R^{-1} p = 6d^2 - p^H R^{-1} p \]

\[ J(\omega_{opt}) = 6d^2 - p^H R^{-1} p \Rightarrow \text{minimum value of polynomial surface.} \]