00 – FIR Filtering Results Review

& Practical Applications

• [p. 2] Mean square signal estimation principles
• [p. 3] Orthogonality principle
• [p. 6] FIR Wiener filtering concepts
• [p.10] FIR Wiener filter equations
• [p. 12] Wiener Predictor
• [p. 16] Examples
• [p. 24] Link between information signal and predictor behaviors
• [p. 27] Examples
• [p. 37] FIR Weiner filter and error surfaces
• [p. 43] Application: Channel equalization
• [p. 52] Application: Noise cancellation
• [p. 56] Application: Noise cancellation with information leakage
• [p. 59] Application: Spatial filtering
• [p. 66] Appendices
• [p. 70] References
Mean Square Signal Estimation

Distorted received signal

Optimal processor

best estimate of transmitted signal \( s \) (as a function of received signal \( x \))

Transmitted signal

Possible procedure:

Mean square estimation, i.e.,

\[
\xi = E \left\{ \left| s - \hat{d}(x) \right|^2 \right\}
\]

minimize \( \xi \)

leads to

\[
\hat{d}(x) = E[ s | x ]
\]

(proof given in Appendix A)

- **conditional mean**, usually nonlinear in \( x \) [exception when \( x \) and \( s \) are jointly normal Gauss Markov theorem]
- Complicated to solve,
- Restriction to Linear Mean Square Estimator (LMS), estimator of \( s \) is **forced** to be a linear function of measurements \( x \):

\[
\hat{d} = h^H x
\]

- Solution via Wiener Hopf equations using orthogonality principle
Orthogonality Principle

Use LMS Criterion: estimate \( s \) by \( \hat{d} = \hat{h}^H x \)
where weights \( \{h_i\} \) minimize MS error:

\[
\sigma_e^2 = E\left\{ |s - \hat{d}(x)|^2 \right\}
\]

**Theorem:** Let error \( e = s - \hat{d} \)
\( \hat{h} \) minimizes the MSE quantity \( \sigma_e^2 \) if \( \hat{h} \) is chosen such that

\[
E\left\{ex^*_i\right\} = E\left\{x^*_i e\right\} = 0, \ \forall \ i = 1, \ldots, N
\]
i.e., the error \( e \) is orthogonal to the observations \( x_i, i = 1 \ldots, N \)
used to compute the filter output.

**Corollary:** minimum MSE obtained: \( \sigma_{e_{\text{min}}}^2 = E\left\{se^*\right\} \) where \( e \) is the minimum error obtained for the optimum filter vector.

(Proof given in Appendix B)
\[ P = 2 \]
\[ \hat{d}(n) = h_0 x(n) + h_1 x(n-1) \]
**Typical Wiener Filtering Problems**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Form of Observations</th>
<th>Desired Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filtering of signal in noise</td>
<td>( x(n) = s(n)+w(n) )</td>
<td>( d(n) = s(n) )</td>
</tr>
<tr>
<td>Prediction of signal in noise</td>
<td>( x(n) = s(n)+w(n) )</td>
<td>( d(n) = s(n+p) ); ( p &gt; 0 )</td>
</tr>
<tr>
<td>Smoothing of signal in noise</td>
<td>( x(n) = s(n)+w(n) )</td>
<td>( d(n) = s(n-q) ); ( q &gt; 0 )</td>
</tr>
<tr>
<td>Linear prediction</td>
<td>( x(n) = s(n-1) )</td>
<td>( d(n) = s(n) )</td>
</tr>
</tbody>
</table>
FIR Wiener Filtering Concepts

- Filter criterion used: minimization of mean square error between $d(n)$ and $\hat{d}(n)$.

- What are we doing here?

We want to design a filter (in the generic sense can be: filter, smoother, predictor) so that:

$$\hat{d}(n) = \sum_{k=0}^{P-1} h^*(k)x(n-k)$$

How $d(n)$ is defined specifies the operation done:

- filtering: $d(n)=s(n)$
- predicting: $d(n)=s(n+p)$
- smoothing: $d(n)=s(n-p)$
How to find $h_k$?

Minimize the MSE: $E \left\{ \left| d(n) - \hat{d}(n) \right|^2 \right\}$

$$\sum_{k=0}^{P-1} h_k^* x(n-k) = h^H x$$

$$h = [h_0, h_{P-1}]^T, \quad x = [x(n), x(n-P+1)]^T$$

Wiener filter is a linear filter $\Rightarrow$ orthogonality principle applies

$$\Rightarrow E \left\{ x(n-i) e^*(n) \right\} = 0, \; \forall i = 0, ..., P-1$$

$$E \left\{ x(n-i) \left[ d(n) - \sum_{k=0}^{P-1} h_k^* x(n-k) \right]^* \right\} = 0, \; \forall i = 0, ..., P-1$$

$$\Rightarrow r_{xd}(-i) - \sum_{k=0}^{P-1} h_k^* R_x(k-i) = 0, \; \forall i = 0, ..., P-1$$
Matrix form:

\[ r_{dx}^* (i) = \sum_{k=0}^{P-1} h_k R_x (k-i), \quad \forall i = 0, \ldots, P-1 \]

\[
\begin{bmatrix}
1 & 10 & 0.5 & 0.25 & 0.125 & \ldots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
10 & 10 & 10 & 10 & 10 & \ldots \\
\end{bmatrix} \begin{bmatrix}
R_x (0) \\
R_x (1) \\
R_x (2) \\
\vdots \\
R_x (P-1) \\
\end{bmatrix} \begin{bmatrix}
h_0 \\
h_1 \\
\vdots \\
h_{P-1} \\
\end{bmatrix}
\]

Note: different notation than in [Therrien, section 7.3]!
Minimum MSE (MMSE) obtained when $h$ is obtained from solving WH equations.

For best $h$ obtained:

$$\sigma^2_{e_{\text{min}}} = E\left\{ |e_{\text{min}}|^2 \right\} = E\left\{ (d(n) - \hat{d}(n))e^*_\text{min} \right\}$$

$$= E\left\{ d(n)e^*_\text{min}(n) \right\}$$

$$= E\left\{ d(n)d(n) - \sum_{k=0}^{P-1} h^*_k x(n-k) \right\}^*$$

$$= R_d(0) - \sum_{k=0}^{P-1} h_k r_{dx}(k)$$

$$\sigma^2_{e_{\text{min}}} = R_d(0) - h^T r_{dx}$$
Summary: FIR Wiener Filter Equations

- FIR Wiener filter is a FIR filter such that:

\[
\hat{d}(n) = \sum_{k=0}^{P-1} h_k^* x(n-k)
\]

where \( \sigma_e^2 = E\left[\left|d(n) - \hat{d}(n)\right|^2\right] \) is minimum.
• How $d(n)$ is defined specifies the specific type of Wiener filter designed:

  filtering:

  smoothing:

  predicting:

• $\rightarrow$ W-H eqs: 

\[
\begin{align*}
R_x h &= r_{dx}^* 
\Rightarrow h_{opt} = R_x^{-1} r_{dx}^* \\
\sigma_{e_{mn}}^2 &= R_d(0) - h_{opt}^T r_{dx} = R_d(0) - r_{dx}^T h_{opt}
\end{align*}
\]
● One-step ahead Wiener predictor
  • tracking of moving series
  • forecasting of system behavior
  • data compression
  • telephone transmission

● W-H equations

\[ h_{opt} = R_x^{-1} r_{dx}^* \]

where \[ \hat{d}(n) = \sum_{\ell=0}^{P-1} h_{\ell}^* x(n - \ell) \]

\[ d(n) = ? \]
Wiener predictor geometric interpretation: Assume a 1-step ahead predictor of length 2 (no additive noise)

\[ e(n) = x(n+1) - \hat{d}(n) \]

\[ \hat{d}(n) = x(n) - x(n-1) \]

\[ e(n) \] is the error between true value \( x(n+1) \) and predicted value for \( x(n+1) \) based on predictor inputs \( x(n) \) and \( x(n-1) \)

\( \Rightarrow \) represents the new information in \( x(n+1) \) which is not already contained in \( x(n) \) or \( x(n-1) \)

\( \Rightarrow \) \( e(n) \) is called the innovation process corresponding to \( x(n) \)
Geometric interpretation, cont’

Assume \( x(n+1) \) only has **NO** new information (i.e., information in \( x(n+1) \) is that already contained in \( x(n) \) and \( x(n-1) \)). Filter of length 2.

Plot \( x(n+1), \hat{d}(n), e(n) \)

\[
\hat{d}(n) = \\
e(n) = 
\]
Geometric interpretation, cont’

Assume $x(n+1)$ only has new information (i.e., information in $x(n+1)$ is that **NOT** already contained in $x(n)$ and $x(n-1)$). Filter of length 2.

Plot $x(n+1), \hat{d}(n), e(n)$
**Example 1: Wiener filter (filter case: \(d(n) = s(n)\) & white noise)**

Assume \(x(n)\) is defined by

\[s(n)\] signal

\(s(n), w(n)\) uncorrelated

\(w(n)\) white noise, zero mean \(R_w(n) = 2\delta(n)\)

\(s(n)\)

\(R_s(n) = 2 (0.8)^{|n|}\)
<table>
<thead>
<tr>
<th>Filter length</th>
<th>Filter coefficients</th>
<th>MMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>[0.405, 0.238]</td>
<td>0.81</td>
</tr>
<tr>
<td>3</td>
<td>[0.382, 0.2, 0.118]</td>
<td>0.76</td>
</tr>
<tr>
<td>4</td>
<td>[0.377, 0.191, 0.01, 0.06]</td>
<td>0.7537</td>
</tr>
<tr>
<td>5</td>
<td>[0.375, 0.188, 0.095, 0.049, 0.029]</td>
<td>0.7509</td>
</tr>
<tr>
<td>6</td>
<td>[0.3751, 0.1877, 0.0941, 0.0476, 0.0249, 0.0146]</td>
<td>0.7502</td>
</tr>
<tr>
<td>7</td>
<td>[0.3750, 0.1875, 0.0938, 0.0471, 0.0238, 0.0125, 0.0073]</td>
<td>0.7501</td>
</tr>
<tr>
<td>8</td>
<td>[0.3750, 0.1875, 0.038, 0.049, 0.0235, 0.0119, 0.0062, 0.0037]</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Example 2: Application to Wiener filter  (filter case: $d(n) = s(n)$ & colored noise)

$s(n)$, $w(n)$ uncorrelated, and zero-mean

$w(n)$ noise with $R_w(n) = 2 (0.5)^{|n|}$

$s(n)$ signal with $R_s(n) = 2 (0.8)^{|n|}$
<table>
<thead>
<tr>
<th>Filter length</th>
<th>Filter coefficients</th>
<th>MMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>[0.4156, 0.1299]</td>
<td>0.961</td>
</tr>
<tr>
<td>3</td>
<td>[0.4122, 0.0750, 0.0878]</td>
<td>0.9432</td>
</tr>
<tr>
<td>4</td>
<td>[0.4106, 0.0737, 0.0508, 0.0595]</td>
<td>0.9351</td>
</tr>
<tr>
<td>5</td>
<td>[0.4099, 0.0730, 0.0499, 0.0344, 0.0403]</td>
<td>0.9314</td>
</tr>
<tr>
<td>6</td>
<td>[0.4095, 0.0728, 0.0495, 0.0338, 0.0233, 0.0273]</td>
<td>0.9297</td>
</tr>
<tr>
<td>7</td>
<td>[0.4094, 0.0726, 0.0493, 0.0335, 0.0229, 0.0158, 0.0185]</td>
<td>0.9289</td>
</tr>
<tr>
<td>8</td>
<td>[0.4093, 0.0726, 0.0492, 0.0334, 0.0227, 0.0155, 0.0107, 0.0125]</td>
<td>0.9285</td>
</tr>
</tbody>
</table>
Example 3: 1-step ahead predictor

RP \( x(n) \) defined as \( x(n) = x(n-1) + v(n) \quad |a| < 1 \)

\( v(n) \) is white noise. 1-step predictor of length 2.

\[
\hat{x}(n) = a_1 x(n-1) + a_2 x(n-2)
\]

\[
a = \begin{bmatrix} +0.5 \\ 0 \end{bmatrix}
\]

\( seed = 1024 \)

AR (1) process

Predictor

\( a = 0.5 \)
Link between Predictor behavior & input signal behavior

1) Case 1: \( s(n) = \) process with correlation

\[
R_s(k) = \delta(k) + 0.5\delta(k - 1) + 0.5\delta(k + 1)
\]

Investigate performances of N-step predictor as a function of changes N

2) Case 2: \( s(n) = \) process with correlation

\[
R_s(k) = a^{|k|}, \quad |a| < 1
\]

Investigate performances of predictor as a function of changes in \( a \)
1) **Case 1**: \( s(n) = \) wss process with

\[
R_s(k) = \delta(k) + 0.5\delta(k - 1) + 0.5\delta(k + 1)
\]
2) Case 2: \( s(n) = \text{wss process with } R_s(k) = a^{|k|}, \ |a| < 1 \)

\[
A = [-0.9:0.1:0.9];
\text{for } k0=1:\text{length}(A)
\quad a = A(k0);
\text{for } k=1:3
\quad rs(k) = a^{(k-1)};
\text{end}
Rs = \text{toeplitz}(rs(1:2));
rdox = [rs(2); rs(3)];
h(:,k0) = Rs \backslash rdx;
\text{mmse}(k0) = rs(1) - h(:,k0) \ast rdx;
\text{end}
\text{stem}(A, \text{mmse})
\text{xlabel('value of a')}
\text{ylabel('MMSE(a')}
\text{title('MMSE(a) for 1-step predictor of length 2, \ldots \text{for } R_s(k)=a^{|k|}')}
\]
Example 4:

\[ s(n) = \text{process with} \]
\[ w(n) = \text{white noise, zero mean} \]
\[ s(n), w(n) \text{ uncorrelated} \]

Design the 1-step ahead predictor of length 2.
Compute MMSE.
<table>
<thead>
<tr>
<th>Length</th>
<th>Coefficients</th>
<th>MMSE</th>
<th>Filter MMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>[0.3238, 0.1905]</td>
<td>1.2381</td>
<td>0.81</td>
</tr>
<tr>
<td>3</td>
<td>[0.3059, 0.16, 0.0941]</td>
<td>1.2094</td>
<td>0.76</td>
</tr>
<tr>
<td>4</td>
<td>[0.3015, 0.1525, 0.0798, 0.0469]</td>
<td>1.2023</td>
<td>0.7537</td>
</tr>
<tr>
<td>5</td>
<td>[0.3004, 0.1506, 0.0762, 0.0762, 0.04, 0.0234]</td>
<td>1.2006</td>
<td>0.7509</td>
</tr>
<tr>
<td>6</td>
<td>[0.3001, 0.1502, 0.0753, 0.0381, 0.0199, 0.0199]</td>
<td>1.2001</td>
<td>0.7502</td>
</tr>
<tr>
<td>7</td>
<td>[0.3, 0.15, 0.0751, 0.0376, 0.0190, 0.001, 0.0059]</td>
<td>1.2</td>
<td>0.7501</td>
</tr>
<tr>
<td>8</td>
<td>[0.3, 0.15, 0.075, 0.0375, 0.0188, 0.0095, 0.0050, 0.003]</td>
<td>1.2</td>
<td>0.75</td>
</tr>
<tr>
<td>Length</td>
<td>1-step ahead MMSE</td>
<td>2-step ahead MMSE</td>
<td>Filter MMSE</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>2</td>
<td>1.2381</td>
<td>1.5124</td>
<td>0.81</td>
</tr>
<tr>
<td>3</td>
<td>1.2094</td>
<td>1.494</td>
<td>0.76</td>
</tr>
<tr>
<td>4</td>
<td>1.2023</td>
<td>1.4895</td>
<td>0.7537</td>
</tr>
<tr>
<td>5</td>
<td>1.2006</td>
<td>1.4884</td>
<td>0.7509</td>
</tr>
<tr>
<td>6</td>
<td>1.2001</td>
<td>1.4881</td>
<td>0.7502</td>
</tr>
<tr>
<td>7</td>
<td>1.2</td>
<td>1.4880</td>
<td>0.7501</td>
</tr>
<tr>
<td>8</td>
<td>1.2</td>
<td>1.4880</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Example 5:

\[ R_s(n) = 2(0.8)^{|n|} \]

\[ R_w(n) = 2(0.5)^{|n|} \]

- \( s(n) \) = process with
  - \( w(n) \) = wss noise, zero mean
  - \( s(n), w(n) \) uncorrelated

- Design the 1-step ahead predictor of length 2
- Design 1-step back smoother of length 2
<table>
<thead>
<tr>
<th>Length</th>
<th>Coefficients</th>
<th>MMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>[0.3325, 0.1039]</td>
<td>1.3351</td>
</tr>
<tr>
<td>3</td>
<td>[0.3297, 0.06, 0.0703]</td>
<td>1.3237</td>
</tr>
<tr>
<td>4</td>
<td>[0.3285, 0.0589, 0.0406, 0.0476]</td>
<td>1.3185</td>
</tr>
<tr>
<td>5</td>
<td>[0.3279, 0.0584, 0.04, 0.0275, 0.0322]</td>
<td>1.3161</td>
</tr>
<tr>
<td>6</td>
<td>[0.3276, 0.0582, 0.0396, 0.0270, 0.0186, 0.0218]</td>
<td>1.315</td>
</tr>
<tr>
<td>7</td>
<td>[0.3275, 0.0581, 0.0394, 0.0268, 0.0183, 0.0126, 0.0148]</td>
<td>1.3145</td>
</tr>
<tr>
<td>8</td>
<td>[0.3275, 0.0581, 0.0394, 0.0267, 0.018, 0.0124, 0.0085, 0.0100]</td>
<td>1.3142</td>
</tr>
<tr>
<td>Length</td>
<td>MMSE (Col. Noise)</td>
<td>( 1 )-step</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.3351</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1.3237</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1.3185</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1.3161</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1.315</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1.3145</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1.3142</td>
</tr>
</tbody>
</table>
Comments
Example 6:

$s(n)$ and $w(n)$ defined as before with $w(n)$ zero mean, and $s(n)$ and $w(n)$ uncorrelated

$$R_s(n) = 2(0.8)^{|n|}$$

$$R_w(n) = 2(0.5)^{|n|}$$

Design the 3-step ahead predictor of length 2, and associated MMSE
Wiener Filters and Error Surfaces

Recall $h_{opt}$ computed from

$$\sigma_e^2 = E \left\{ \left| d(n) - h^H x \right|^2 \right\} = E \left\{ (d(n) - h^H x)(d(n) - h^H x)^* \right\}$$

$$= R_d(0) + h^H E \left\{ xx^H \right\} h - 2 \text{Real} \left( h^T r_{dx} \right)$$

for real signals $d(n), x(n)$

$$\sigma_e^2 = R_d(0) + h^T R_x h - 2 h^T r_{dx}$$

using the fact that

$$\left( h^H x \right)^* = x^H h$$

$$d(n) h^H x = h^H d(n) x$$
\[ \sigma_e^2 = R_d(0) + h^T R_x h - 2h^T r_{dx} \]

\( \square \) For filter length \( P = 1 \) \quad \( h = h_0 \quad x = x(n) \)

\[ \sigma_e^2 = R_d(0) + h(0)^2 R_x(0) - 2h(0) r_{dx}(0) \]
\[
\sigma_e^2 = R_d(0) + h^T R_x h - 2 h^T r_{dx}
\]

For filter length \( P = 2 \)

\[
h = \begin{bmatrix} h(0), h(1) \end{bmatrix}^T ; \quad x = \begin{bmatrix} x(n), x(n-1) \end{bmatrix}^T
\]

\[
\sigma_e^2 = R_d(0) + h^T R_x h - 2 h^T r_{dx}
\]

\[
= R_d(0) + \begin{bmatrix} h(0), h(1) \end{bmatrix} \begin{bmatrix} R_x(0) & R_x(1) \\ R_x(1) & R_x(0) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \end{bmatrix}
\]

\[
- 2 \begin{bmatrix} h(0), h(1) \end{bmatrix} \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \end{bmatrix}
\]

\[
\Rightarrow \quad \sigma_e^2 = A_0 h(0)^2 + A_1 h(1)^2 + A_2 h(1) + A_3 h(0)
\]

\[
+ A_4 h(0) h(1) + R_d(0)
\]
shape of $\sigma_e^2$ depends on $R_x$ information: $\lambda(R_x)$ & eigenvectors

$\sigma_e^2 = R_d(0) + h^T R_x h - 2 h^T r_{dx}$

moves $\sigma_e^2$ up and down

$R_d$: specifies shape of $\sigma_e^2(h)$

$r_{dx}$: specifies where the bowl is in the 3-d plane but doesn’t change the shape of the bowl

$R_d(0)$: moves bowl up and down in 3-d plane but doesn’t change shape or location of bowl
Correlation matrix Eigenvalue Spread Impact on Error Surface Shape
see plots

Eigenvector Direction for $2 \times 2$ Toeplitz Correlation Matrix

$$R_x = \begin{bmatrix} R_x(0) & R_x(1) \\ R_x(1) & R_x(0) \end{bmatrix} \rightarrow \text{normalize correlation} \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}$$

eigenvalues of $R_x$
$$(1 - \lambda)^2 - a^2 = 0 \quad \Rightarrow \quad \lambda = \begin{cases} 1 - a \\ 1 + a \end{cases}$$
eigenvectors
$$\begin{pmatrix} 1 - \lambda & a \\ a & 1 - \lambda \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} = 0$$
$$\Rightarrow (1 - \lambda)u_{11} + au_{12} = 0$$
$$\lambda_1 = 1 - a \quad \Rightarrow \quad (\chi' - \chi + a)u_{11} + au_{12} = 0$$
$$\Rightarrow u_{11} = -u_{12} \quad \Rightarrow u_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$
$$\lambda_2 = 1 + a \quad \Rightarrow \quad u_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
Error surface shape and eigenvalue ratios

$a=0.1$

$a=0.5$

$a=0.99$
Goal: Recover $s(n)$ by estimating channel distortion (applications in communications, control, etc.)

Assumptions:
1) $v(n)$ is stationary, zero-mean, uncorrelated with $s(n)$.
2) $v(n) = 0$ & $D=0$

Application to Channel Equalization

Goal: Implement the equalization filter $H(z)$ as a stable causal FIR filter

Information Available:
\[ x(n) = z(n) + v(n) \]
\[ d(n) = s(n-D) \]

Channel output
Additive noise due to sensors
Assume: \( W(z) = 0.2798 + z^{-1} + 0.2798z^{-2} \)

Questions:

1) Assume \( v(n) = 0 \) & \( D=0 \). Identify the type of filter (FIR/IIR) needed to cancel channel distortions. Identify resulting \( H(z) \).

2) Identify whether the equalization filter is causal and stable.

3) Assume \( v(n) = 0 \) & \( D \neq 0 \). Identify resulting \( H_2(z) \) in terms of \( H(z) \).
equalizer impulse response - no noise
Assume $D \neq 0$

\[
\hat{d}(n) = d(n) = s(n-D)
\]
## Application to Noise Cancellation

- **Goal:** Recover $s(n)$ by compensating for the noise distortion while having only access to the related noise distortion signal $v(n)$ (applications in communications, control, etc.)

\[
\begin{align*}
\text{Signal source} & \quad s(n) \\
\text{White noise source} & \quad w(n) \\
\text{Transformation} & \quad \text{Of white noise into correlated noise} \\
\text{Wiener filter} & \quad A(z) \\
\text{Information Available:} & \quad s(n) + w(n) \text{ & } v(n) \\
\text{Assumption:} & \quad w(n) \text{ is stationary, zero-mean, uncorrelated with } s(n).
\end{align*}
\]
Assume: \( v(n) = av(n - 1) + w(n), \quad a = 0.6, \)
\( w(n) \) white noise with variance \( \sigma_w^2 \)
\( s(n) = \sin(\omega_0 n + \phi), \quad \phi \sim U[0, 2\pi] \)

Compute the FIR Wiener filter of length 2 and evaluate filter performances
Results Interpretation
Application to Noise Cancellation (with information leakage)

- Signal source $s(n)$
- White noise source $w(n)$
- Wiener filter

$d(n) = s(n) + w(n)$

$e(n) = d(n) - \hat{d}(n)$

**Information Available:**

$s(n) + w(n)$ & $v(n) + Ks(n)$

**Assumption:** $w(n)$ is stationary, zero-mean, uncorrelated with $s(n)$. 

Transformation of white noise into correlated noise $A(z)$
Results Interpretation
Application to Spatial Filtering

Information Available:
- snapshot in time of received signal retrieved at two antennas & reference signal

Assumption:
- $v_1(n), v_2(n)$ zero mean wss white noise RPs independent of each other and of $s(n)$.

Goal: Denoise received signal
Application to Spatial Filtering, cont’
**Example:** Gain Pattern at filter output

N-element array, 
- desired signal 
  at $\theta_0 = 30^\circ$,  
- interferences 
  at $\theta_1 = -20^\circ$  
  $\theta_2 = 40^\circ$ 
- Noise power: 0.1

**Example:**

Array steering vector

\[
A_w(\theta) = \left| w^H \hat{a}(\theta) \right|^2
\]
Application to Spatial Filtering, cont’

Did we gain anything by using multiple receivers?

- **Assumption**: \( v(n) \) zero mean RP independent of \( s(n) \).

- **Goal**: Compute filter coefficients, filter output, and MMSE.
Appendices
Appendix A: Derivation of proof for Mean Square estimate derivation (p. 3)

Proof:

\[ \xi = E \left\{ |s - \hat{d}(x)|^2 \right\} \]

\[ = E \left\{ (s - \hat{d}(x))(s - \hat{d}(x))^T \right\} \]

Define \( L(s, \hat{s}(x)) \) : loss function

\[ \xi = \int \int L(s, \hat{d}(x)) f(s|x) f(x) d^M s, \quad \text{Bayes Rule} \]

\[ = \int \left[ \int L(s, \hat{d}(x)) f(s|x) ds \right] f(x) d^M x \]

\( \xi \) is minimized if \( K \) is minimized for each value of \( x \)

Problem: find \( \hat{d}(x) \) so that \( K \) is minimum

\[ \frac{\partial K}{\partial \hat{d}} = \frac{\partial}{\partial \hat{d}} \left[ \int \left( s - \hat{d}(x) \right)^2 f(s|x) ds \right] \]

\[ = -2 \int \left( s - \hat{d}(x) \right) f(s|x) ds \]

\[ \frac{\partial K}{\partial \hat{d}} = 0 \quad \Rightarrow \quad \int \left( s - \hat{d}(x) \right) f(s|x) ds = 0 \]

\[ \Rightarrow \int s f(s|x) ds = \int \hat{d}(x) f(s|x) ds \]

\[ \Rightarrow \int s f(s|x) ds = \hat{d}(x) \int f(s|x) ds \]

\[ \Rightarrow \int s f(s|x) ds = \hat{d}(x) \times 1 \]

\[ \Rightarrow E[s|x] = \hat{d}(x) \]
For a minimum we need: \( \frac{\partial^2 K}{\partial \hat{d}^2} > 0 \)

Note:

\[
\frac{\partial^2 K}{\partial \hat{d}^2} = (-2) \partial \left( \int (s - \hat{d}(x)) f(s | x) ds \right) / \partial \hat{d}
\]

\[
= (-2) \int -f(s | x) ds
\]

\[
= (2) \times 1 > 0
\]
Appendix B: Proof of corollary of orthogonality principle

Proof:

\[ e = s - h^H x = s - \left( h + \hat{h} - \hat{h} \right)^H x \]

where \( \hat{h} \) is the weight vector defined so that the orthogonality principle holds. Resulting error is called \( \hat{e} = s - \hat{h} x \)

\[
\sigma_e^2 = E \{ |e|^2 \} = E \left\{ \left( s - \hat{h}^H x + (\hat{h} - h)^H x \right) \left( s - \hat{h}^H x + (\hat{h} - h)^H x \right) \right\}^H \\
= E \left\{ \left( \hat{e} + (\hat{h} - h)^H x \right) \left( \hat{e} + (\hat{h} - h)^H x \right) \right\}^H \\
= E \left\{ \left| \hat{e} \right|^2 + (\hat{h} - h)^H x \hat{e}^H + \left( \hat{h} - h \right)^H x \left( \hat{h} - h \right) \right\} + E \left\{ \hat{e} \right\} \left( \hat{h} - h \right) \\
= E \left\{ \left| \hat{e} \right|^2 \right\} + (\hat{h} - h)^H E \left\{ x \hat{e}^H \right\} \left( \hat{h} - h \right) + E \left\{ \left( \hat{h} - h \right)^H x \right\} \left( \hat{h} - h \right) \\
= E \left\{ \left| \hat{e} \right|^2 \right\} = E \left\{ (s - \hat{h}^H x) \hat{e}^H \right\} = E \left\{ s \hat{e}^H \right\} - \hat{h} E \left\{ x \hat{e}^H \right\} \\
= E \left\{ s \hat{e}^H \right\} = E \left\{ s \hat{e}^* \right\} \]
References