Signal Prediction

- Inverse filtering
- Deconvolution
- Matched filters
- Microelectrode Detection Example

Goal: estimate $x(n + L)$

Application that we have already studied and applied

Every sample, we obtain more data to update our estimator with

How do we do so efficiently?

Applications: prediction, modeling, compression

Interference Cancellation

$y(n) = s(n) + v(n)$

- If $x(n)$ are noise reference signals, then $e(n)$ is an estimate of $y(n)$ with some of the noise removed
- Can be generalized to multiple input signals
- Used in many applications (think noise cancelling head phones)
Blind Deconvolution

- In this case the input is unknown as well as the system $G(z)$
- Goal is to estimate the (possibly delayed and scaled) input $w(n)$ and possibly the system $h(n) * g(n) * w(n) \approx b_0 w(n - n_0)$
- Problem is ill-defined if we have no information about $w(n)$
- Need to know something about $r_{xy}(\ell) = r_{xw}(\ell) = g(\ell) * r_w(\ell)$
- Often assume $w(n)$ is a WN$(0, \sigma_w^2)$ process

Optimum Inverse Modeling

- Ideally we would like to obtain $y(n)$ from $x(n)$
- May be difficult if
  - $G(z)$ contains a delay
  - The additive noise is significant
  - $G(z)$ is nonminimum phase
  - The inverse system is IIR
- In practice, $H(z)$ is an FIR filter
- In some applications a delayed estimate is acceptable, $\hat{y}(n) = y(n - D)$
Matched Filters: Estimator Properties

\[ y(n) = c^H x(n) = c^H s(n) + c^H v(n) \]
\[ P_y(n) = E[|y(n)|^2] = E[c^H x(n)x^H(n)c] = c^H R_x(n)c \]

- The output power depends only on the signal autocorrelation matrix
- Because \( s(n) \perp v(n) \), \( R_x(n) \) can be expressed as
  \[ R_x(n) = E[x(n)x^H(n)] \]
  \[ = E[(s(n) + v(n))(s(n) + v(n))^H] \]
  \[ = E[s(n)s^H(n)] + E[v(n)v^H(n)] \]
  \[ = R_s(n) + R_v(n) \]
- The structure of \( R_x(n) \) depends on the statistical model of \( s(n) \)

Matched Filters: Problem Definition

\[ x_M(1) = s(n) + v(n) \]

where \( s(n) \) is the signal of interest and \( v(n) \) is a noise signal

- Suppose we have "brief" events of interest that occur in a noisy background
- Goal: detect the events
- Applications: radar, sonar, microelectrode recordings, communications
- Suppose we decide to form a linear combination, \( y(n) = c^H x(n) \) and then apply a threshold for detection
- Generally it is assumed that
  \[ R_{si}(\ell) = \sigma^2_{si} \delta(\ell) \ \forall i \]
  \[ R_{vj}(\ell) = \sigma^2_{vj} \delta(\ell) \ \forall j \]
  \[ R_{si,vj}(\ell) = 0 \]

Matched Filters: Detection Metrics

\begin{array}{|c|c|c|}
\hline
& \text{Events} & \text{Non-events} \\
\hline
\text{Detected} & TP & FP & ND \\
\text{Not detected} & FN & TN & ND \\
\hline
\text{Total} & NE & NE & ND \\
\hline
\end{array}

- Sensitivity: \( \frac{TP}{NE} = \frac{TP}{TP+FN} \)
  - Fraction of events that are detected
- Specificity: \( \frac{TN}{NE} = \frac{TN}{FP+TN} \)
  - Fraction of non-events that are not detected
- Positive Predictivity: \( \frac{TP}{ND} = \frac{TP}{TP+FP} \)
  - Fraction of detected events that are events
- Negative Predictivity: \( \frac{TN}{ND} = \frac{TN}{FN+TN} \)
  - Fraction of nondetected events that are not events

Remaining Tasks

- Add an example with a synthetic signal and known system much like Example 6.7.1 in the text
- Add a practical example
**Matched Filters: Metric Selection**

- Goal: maximize the detection accuracy
  - Minimize the number of false positives and false negatives
  - Maximize the sensitivity and specificity
  - Maximize the positive predictivity and negative predictivity
- There is almost always a tradeoff between each of these pairs
- Threshold controls the tradeoff
- For a given tradeoff, what is the best parameter vector \( c \)?
- Relationship is difficult to obtain in general

**Matched Filters: Problem Classes**

\[ \mathbf{x}(n) = \mathbf{s}(n) + \mathbf{v}(n) \]

- The signal may be
  - Random with autocorrelation matrix \( \mathbf{R}_s(n) \)
  - Deterministic with random amplitude, \( \mathbf{s}(n) = \alpha \mathbf{s}_0 \)
- Similarly, the noise may be colored or white
- Four problems altogether
- Consider one at a time

**Matched Filters: Signal-to-Noise Ratio**

\[ y(n) = c^H \mathbf{x}(n) = c^H \mathbf{s}(n) c^H \mathbf{v}(n) \]
\[ P_y(n) = c^H \mathbf{R}_x(n) c = c^H \mathbf{R}_s(n) c + c^H \mathbf{R}_v(n) c \]

- Proxy goal: Find \( c \) such that the signal-to-noise ratio is maximized
\[ \text{SNR} = \frac{c^H \mathbf{R}_s(n) c}{c^H \mathbf{R}_v(n) c} \]
- Rationale: detection will be easier if
  - \( y(n) \) is large when signal is present
  - \( y(n) \) is small when only noise is present

**Matched Filters: Deterministic Signal in White Noise**

\[ \mathbf{s}(n) = \alpha \mathbf{s}_0 \]
\[ P_s(n) = E [ |c^H \alpha \mathbf{s}_0|^2 ] = P_\alpha |c^H \mathbf{s}_0|^2 \]
\[ \text{SNR}(c) = P_\alpha |c^H \mathbf{s}_0|^2 P_v c^H \mathbf{R}_v c \]

Now if \( \mathbf{R}_v(n) = \sigma_v^2 \mathbf{I} \)
\[ \text{SNR}(c) = \frac{P_\alpha |c^H \mathbf{s}_0|^2}{P_v c^H \mathbf{c}} \]

From the Cauchy-Schwartz inequality
\[ c^H \mathbf{s}_0 \leq \sqrt{(c^H \mathbf{c})(\mathbf{s}_0^H \mathbf{s}_0)} \]
\[ \text{SNR}(c) = \frac{P_\alpha |c^H \mathbf{s}_0|^2}{P_v c^H \mathbf{c}} \leq \frac{P_\alpha (c^H \mathbf{c})(\mathbf{s}_0^H \mathbf{s}_0)}{P_v c^H \mathbf{c}} = \frac{P_\alpha \mathbf{s}_0^H \mathbf{s}_0}{P_v} \]
Matched Filters: Whitening

Let us define

\[ \tilde{v}(n) \triangleq L_v^{-1} v(n) \]
\[ R_{\tilde{v}} = E \left[ (L_v^{-1} v(n))(L_v^{-1} v(n))^\text{H} \right] \]
\[ = E \left[ L_v^{-1} v(n)v^\text{H}(n)L_v^{-\text{H}} \right] \]
\[ = L_v^{-1} E \left[ v(n)v^\text{H}(n) \right] L_v^{-\text{H}} \]
\[ = L_v^{-1} R_v L_v^{-\text{H}}(n) \]
\[ = L_v^{-1} L_v^\text{H} L_v^{-\text{H}}(n) \]
\[ = I \]

The SNR is maximized when \( c \) is the same as the expected event shape.
This is why it is called a \textit{matched filter}.
SNR is scale invariant \( \text{SNR}(c) = \text{SNR}(\beta c) \)

Matched Filters: Deterministic Signal in White Noise

\[ c_o = \beta s_0 \]
\[ \text{SNR}_o(c) = \frac{P_\alpha}{F_v}s_0^\text{H} s_0 \]

• The SNR is maximized when \( c \) is the same as the expected event shape.
• This is why it is called a \textit{matched filter}.
• SNR is scale invariant \( \text{SNR}(c) = \text{SNR}(\beta c) \)

Matched Filters: Deterministic Signal in Colored Noise

If the noise covariance matrix \( R_v(n) \) is positive definite, then there exists a square root such that
\[ R_v = L_v L_v^\text{H} \]

• The text assumes that \( L_v \) is a lower-upper Cholesky decomposition, but it could be any square root of \( R_v(n) \) for this application.
• Here I am assuming the signal and noise are jointly stationary to simplify notation, but it works in the nonstationary case as well.

Matched Filters: Colored Noise SNR

\[ \text{SNR}(c) = P_\alpha |c^\text{H} s_0|^2 \]
\[ = P_\alpha |c^\text{H} L_v L_v^{-1} s_0|^2 \]
\[ = P_\alpha |c^\text{H} s_0|^2 \]

where
\[ \tilde{c} \triangleq L_v^\text{H} c \]
\[ \tilde{s}_0 \triangleq L_v^{-1} s_0 \]

Thus, this reduces to the same problem as the white noise case and the solution follows immediately
\[ \tilde{c}_o = \beta \tilde{s}_0 \]
\[ c_o = \beta L_v^{-\text{H}} L_v^{-1} s_0 = \beta R_v^{-1} s_0 \]
\[ \text{SNR}_o(c) = \frac{P_\alpha}{F_v}s_0^\text{H} \tilde{s}_0 \]
\[ \text{SNR}_o(c) = \frac{P_\alpha}{F_v}s_0^\text{H} R_v^{-1} s_0 \]
Matched Filters: Random Signal & White Noise Case

Suppose \( R_v = \sigma_v^2 I \). Then

\[
SNR(c) = \frac{1}{\sigma_v^2} \frac{c^H R_s c}{c^H c}
\]

Now let us define \( \tilde{c} = Q \Lambda^{\frac{1}{2}} c \) so that

\[
SNR(c) = \frac{1}{\sigma_v^2} \frac{c^H Q \Lambda Q^H \tilde{c}}{c^H \tilde{c}} = \frac{1}{\sigma_v^2} \frac{c^H \tilde{c}}{c^H \tilde{c}} \sum_{k=1}^{M} |\tilde{c}_k|^2 \lambda_k
\]

where \( 0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_M \). Now since this is a weighted average

\[
\lambda_{\text{min}} \leq \frac{\sum_{k=1}^{M} |\tilde{c}_k|^2 \lambda_k}{\sum_{k=1}^{M} |\tilde{c}_k|^2} \leq \lambda_{\text{max}}
\]

We can maximize this sum by picking \( \tilde{c}_k = 0 \) for \( k \neq M \)

Matched Filters: Random Signal Case

In this case the signal \( s(n) \) is random with covariance matrix \( R_s \). Again we assume the signal and noise are jointly stationary, merely to simplify the notation. Consider now the signal power portion of \( y(n) \)

\[
P_s = E[|c^H s(n)|^2] = E[(c^H s(n))(c^H s(n))^*] = E[c^H s(n)s^H(n)c] = c^H E[s(n)s^H(n)] c = c^H R_s c
\]

So the SNR is given by

\[
SNR(c) = \frac{c^H R_s c}{c^H R_v c}
\]

Matched Filters: Deterministic Signal & Colored Noise

\[
c_o = \beta R_v^{-1} s_0 \quad SNR_o(c) = \frac{P_n}{P_v} s_0^H R_v^{-1} s_0
\]

- If we were to preprocess the signal with \( L_v^{-1} \), the observed signals would be

\[
L_v^{-1} x(n) = L_v^{-1} s(n) + L_v^{-1} v(n)
\]

= \( \tilde{s}(n) + \tilde{v}(n) \)

- This essentially converts the colored noise problem into the white noise problem since \( R_v = I \)

- So again (third time?) we see that the whitening filter is a useful tool

Matched Filters: Random Signal & Colored Noise

The optimum filter is therefore

\[
c_o = q_{\text{max}} \quad SNR_o(c) = \frac{\lambda_{\text{max}}}{\sigma_v^2}
\]

- This is why these are sometimes called eigenfilters

- Again, the colored noise case can be solved by applying a whitening filter as a preprocessing step to reduce it to the white noise case
Example 1: MER Example

Use an matched filter to detect the times of action potentials in a microelectrode recording.
idi = ceil(ed*fs/2); % Duration before and after each event
did = (-idi:idi)'; % Segment indices (samples)
t = did/fs; % Segment indices (seconds)
I = did*ones(1,length(si)) + ones(length(did),1)*si'; % Matrix of segment indices
I(I<=0) = 1; % Handle left edge condition
I(I>nx) = nx; % Handle right edge condition
co = median(x(I),2);

figure;
FigureSet(1,'LTX');
h = plot(t,x(I),'b',t,median(x(I),2),'g');
set(h,'Color',0.8*[1 1 1]);set(h(end),'Color',0.5*[0 1 0]);
set(h,'LineWidth',0.1);set(h(end),'LineWidth',2);
xlabel('Time (s)');ylabel('Signal ($\mu$V?)');xlim([t(1) t(end)]);box off;AxisSet;
print('MERTemplate','-depsc');

%================================================
% Plot a Segment of the Matched Filter Output%================================================
es = ceil(ed*fs/2);y = filter(flipud(co),1,[x;zeros(es,1)]);y = y(es+(0:nx-1)); % HERE - I don't understand this
figure;
FigureSet(1,'LTX');k = 1:length(x);
t = (k-0.5)/fs;h = plot(t,y,'b',t(si),y(si),'r.');set(h,'LineWidth',0.5);xlabel('Time (s)');ylabel('Matched Filter Output');xlim([0.15 0.23]);box off;AxisSet;
print('MEROutputSegment','-depsc');

%================================================
% Plot a Segment of the Matched Filter Output%================================================
DetectionPlot(x,fs,si,2);
FigureSet(1,'LTX');AxisSet;
print('MERDetectionHistogram','-depsc');

clear all;
close all;

% User-Specified Parameters
ed = 1e-3; % Event duration (seconds)

% Load the Signal
load MER;
x = x(1:ceil(5*fs));
si = si(si<ceil(5*fs));

% Plot A Segment of the Signal
figure;
FigureSet(1,'LTX');k = 1:length(x);
t = (k-0.5)/fs;
h = plot(t,x,'b',t(si),x(si),'r.);
set(h,'LineWidth',0.5);
xlabel('Time (s)');ylabel('Signal ($\mu$V?)');xlim([0.15 0.23]);box off;AxisSet;
print('MERSignalSegment','-depsc');

nx = length(x);
Remaining Tasks

- Add slide on ROC curves
- Fix bugs, show result of second detection with real example
- Synthetic example of a matched filter