Observed Signals

Generally I will call \( y(n) \) the target or desired response

The RVs that we observe will often be called the input variables (to the estimator)

These may be of several types

- No temporal component, just a set of observations for each realization: \( x = [x_1, x_2, \ldots, x_M]^T \)
- Separate observations with a temporal component:  
  \[
  x(n) = [x_1(n), x_2(n), \ldots, x_M(n)]^T
  \]
- Samples from signal segment:  
  \[
  x(n) = [x(n), x(n-1), \ldots, x(n-M)]^T
  \]

Again, in the last case the book assumes the signal \( x(n) \in \mathbb{C}^{1 \times 1} \)

is univariate, but everything can be generalized to the multivariate case

Many applications (see Chapter 1): find one

Problem Formulation

Goal for much of this class is to estimate a random variable

Usual assumptions

- Is a signal, \( y(n) \in \mathbb{C}^{1 \times 1} \)

Much of what we discuss is easily generalized to the multivariate case

Not clear why books focus on univariate signal

Also assume we can observe other random variables, collected in a vector \( x(n) \), to estimate \( y(n) \) with
Ensemble versus Realizations Estimators

- Fundamentally, the book discusses two approaches to estimation
  - Ensemble Performance Metrics
    - $P$ is a function of the joint distribution of $[y(n), x(n)^T]$ 
      - Estimator performs well on the ensemble 
        - Chapters 6 and 7 
        - Example: Minimum Mean Square Error (MMSE)
  - Realization Performance Metrics
    - $P$ is a function of observed data (a realization) 
      - Estimator performs well on that data set 
        - Chapters 8 and 9 
        - Example: Least square error (LSE)
- The two approaches are closely related

Error Signal

$$\tilde{y}(n) \triangleq y(n) - \hat{y}(n) \triangleq e(n)$$

- We want $\hat{y}(n)$ to be as close to $y(n)$ as possible
- In order to find the "optimal" estimator, we must have a definition of optimality
- Most definitions are functions of the error signal, $e(n)$
- They are not equivalent, in general
- Let $P$ denote the performance criterion
- Also called the performance metric or cost function
- May wish to maximize or minimize depending on the definition

Estimation Problem

Given a random vector $x(n) \in \mathbb{C}^{M \times 1}$, determine an estimate $\hat{y}(n)$ using a "rule"

$$\hat{y}(n) \triangleq h[x(n)]$$

- The "rule" $h[x(n)]$ is called the estimator
- In general it is (could be) a nonlinear function
- If $x(n) = [x(n), x(n-1), \ldots, x(n-M)]^T$, the estimator is called a discrete-time filter
- The estimator could be
  - Linear or nonlinear
  - Time-invariant or time-varying, $h_n[x(n)]$
  - FIR or IIR

Optimum Estimator Design

1. Select a structure for the estimator (usually parametric)
2. Select a performance criterion
3. Optimize the estimator (solve for the best parameter values)
4. Assess the estimator performance
Selection of Performance Criterion

- If approximation to subjective criteria (e.g., health, sound quality, image quality) is irrelevant or unimportant, two other factors motivate the choice:
  1. Sensitivity to outliers
  2. Mathematical tractability
- Fundamental tradeoff: tractability of the solution versus accuracy of subjective quality
- Is usually a function of the estimation error
- Often has even symmetry: \( p[e(n)] = p[e(-n)] \)
  - Positive errors are equally harmful as negative errors

It is often difficult to express subjective criteria (e.g., health, sound quality, image quality) mathematically
- This is where your judgement, experience are required to make a design decision that is suited to your application
- Fundamental tradeoff: tractability of the solution versus accuracy of subjective quality
- Is usually a function of the estimation error
- Often has even symmetry: \( p[e(n)] = p[e(-n)] \)
  - Positive errors are equally harmful as negative errors

Selection of Performance Criterion

- Much of the known theory is limited to
  - MSE or ASE/SSE performance metrics
  - Linear estimators
- Why?
  - Thorough and elegant optimal results are known
  - Only requires knowledge of second-order moments, which can be estimated (ECE 5/638)
  - Many of the other cases are generalizations of the linear case
- This is a critical foundation for a career in signal processing
Zero Mean Assumption

All RVs are assumed to have zero mean

- Greatly simplifies the math
- Means that all covariances are simply correlations
- In practice, is enforced by removing the mean and/or trend
  - Subtract the sample average and assume statistical impact is negligible
  - Highpass filter, but watch for edge effects
  - First order difference
  - Other (detrend)
- This is a gotcha, so watch this carefully

Error Performance Surface

The error performance surface is simply the multivariate error criterion expressed as a function of the parameter vector

\[ P(c) = E[|e|^2] \]
\[ = E \left[ (y - c^H x)(y^* - x^H c) \right] \]
\[ = E[|y|^2] - c^H E[xy^*] - E[yy^H]c + c^H E[xx^H]c \]
\[ = P_y - c^H d - d^H c + c^H Rc \]

where

\[ P_y \triangleq E[|y|^2] \]
\[ d \triangleq E[xy^*] \]
\[ R \triangleq E[xx^H] \]

You should be able to show that \( R \) is Hermitian and nonnegative definite. In virtually all practical applications, it is positive definite and invertible.

Linear MSE Estimation

\[ \hat{y}(n) \triangleq c(n)^H x = \sum_{k=1}^{M} c_k^* x_k(n) \]
\[ P(c(n)) \triangleq E[|e(n)|^2] \]

- Design goal: design a linear estimator of \( y(n) \) that minimizes the MSE
- Equivalent to solving for the model coefficients \( c \) such that the MSE is minimized
- We assume that the RVs \( \{y(n), x(n)^T\} \) are realizations of a stochastic process
- If jointly nonstationary, the optimal coefficients are time-varying, \( c(n) \)
- The time index will often be dropped to simplify notation

\[ \hat{y}(n) \triangleq c^H x = \sum_{k=1}^{M} c_k^* x_k(n) \]
\[ P(c) \triangleq E[|e|^2] \]

Notation

I believe the book chose the linear estimator to use the complex conjugate of the coefficients,

\[ \hat{y}(n) = \sum_{k=1}^{M} c_k^* x_k(n) \]

so that this could be defined as an inner product of the parameter or coefficient vector \( c \in \mathbb{C}^{M \times 1} \) and the input data vector \( x(n) \in \mathbb{C}^{M \times 1} \) as

\[ c^H x = \sum_{k=1}^{M} c_k x_k = (c, x) \]

The parameter vector that minimizes the MSE, denoted as \( c_o \) is called the linear MMSE (LMMSE) estimator

\( \hat{y}_o \) is called the LMMSE estimate
Error Performance Surface

\[ P(c) = P_y - c^H d - d^H c + c^H R c \]

- \( P(c) \) is called the **error performance surface**
- It is a quadratic function of the parameter vector \( c \)
- If \( R \) is positive definite, it is strictly convex with a single minimum
  at the optimal solution \( c_0 \)
- Can think of as a quadratic bowl in an \( M \) dimensional space
- Our goal is to find the bottom of the bowl

Example 1: Error Performance Surface

Plot the error surface for the following covariance matrices. What is the optimal solution?
Nonlinear Functions

- If the estimator is nonlinear in the parameters or the performance
criterion is not MSE
  - The error surface is not quadratic
  - If nonlinear, the error surface may contain multiple local
minima

- There is no guaranteed algorithm to find a global minimum when
there are multiple local minima

- Many heuristic algorithms (genetic algorithms, evolutionary
programming, etc.)

- Usually computationally expensive

- Can’t apply in most online signal processing problems

- Many good solutions when one global minimum
  - Convex, pseudoconvex, and quasi-convex performance criteria

- Example 6.2.2 is an excellent example of this in a simple filtering
context

Example 1: MATLAB Code

```matlab
R = [1 0.2;0.2 1];
d = [3;2];
Py = 5;
np = 100;
co = inv(R)*d;
c1 = linspace(-5+co(1),5+co(1),np);
c2 = linspace(-5+co(2),5+co(2),np);
[C1,C2] = meshgrid(c1,c2);
P = zeros(np,np);
for i=1:np,
    for j=1:np,
        c = [C1(i,j);C2(i,j)];
        P(i,j) = Py - c'*d - d'*c + c'*R*c;
    end;
end;
figure;
h = imagesc(c1,c2,P);
set(gca,'YDir','Normal');
hold on;
h = plot(co(1),co(2),'ko');
set(h,'MarkerFaceColor','w');
set(h,'MarkerEdgeColor','k');
set(h,'MarkerSize',7);
set(h,'LineWidth',1);
hold off;
```

Example 2: Nonlinear Estimation Error Surface

Suppose we wish to model two ARMA processes.

\[ G_1(z) = \frac{1}{(1 - 0.9z^{-1})(1 + 0.9z^{-1})} \]
\[ G_2(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \]

Let us use a pole zero filter for our estimator,

\[ H(z) = \frac{b}{1 - az^{-1}} \]
\[ h(n) = ba^n u(n) \]

though this is clearly not optimal.

\[ y(n) = g(n) \ast x(n) \]
\[ \hat{y}(n) = h(n) \ast x(n) \]

Plot the nonlinear error surface and the transfer function of the
optimal estimate at all local minima.
Example 2: Nonlinear Estimation Error Surface Continued

\[ P_e = E[|y(n) - \hat{y}(n)|^2] \]
\[ = E[y(n)^2] - 2E[y(n)\hat{y}(n)] + E[\hat{y}(n)^2] \]
\[ E[y(n)^2] = \sigma_x^2 \sum_{n=0}^{\infty} |g(n)|^2 \]
\[ = \frac{\sigma_x^2}{2\pi} \int_{-\pi}^{\pi} |G(e^{j\omega})|^2 d\omega \]
\[ E[\hat{y}(n)^2] = \frac{\sigma_x^2}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega \]
\[ E[y(n)\hat{y}(n)] = \frac{\sigma_x^2}{2\pi} \int_{-\pi}^{\pi} G(e^{j\omega})H^*(e^{j\omega}) d\omega \]

These three integrals can be approximated with a Reimann sum with the FFT.
Example 2: Nonlinear Error Performance Surface

\[ H(e^{j\omega}) \]

Actual
Optimal
Local Minima

Example 2: MATLAB Code

```matlab
% Example 2: Nonlinear Error Performance Surface

% No. of points to evaluate in the frequency domain
nz = 2^9;
% No. points to evaluate in each dimension
np = 150;

for c0=1:2,
    switch c0
        case 1,
            [G,w] = freqz(1,poly([-0.9 0.9]),nz);
            br = [-2 4]; % Range of b
            vw = [-70 5]; % View
        case 2,
            [G,w] = freqz([0.05 -0.4],[1 -1.1314 0.25],nz);
            br = [-1 1]; % Range of b
            vw = [-85 2]; % View
    end;

    a = linspace(-0.99,0.99,np);
    b = linspace(br(1),br(2),np);
    Po = inf;
    P = zeros(np,np); % Memory allocation
    for c1=1:np,
        for c2=1:np,
            [H,w] = freqz(b(c2),[1 -a(c1)],nz);
            P(c1,c2) = sum(abs(G).^2) -2*sum(real(G.*conj(H))) + sum(abs(H).^2);
            P(c1,c2) = P(c1,c2)/sum(abs(G).^2);
            if abs(P(c1,c2))<Po,
                Po = abs(P(c1,c2));
                ao = a(c1);
                bo = b(c2);
            end;
        end;
    end;
```

Example 2: Nonlinear Error Fits

\[ H(e^{j\omega}) \]

Actual
Optimal
Local Minima

Example 2: MATLAB Code

```matlab
% Example 2: MATLAB Code

% No. of points to evaluate in the frequency domain
nz = 2^9;
% No. points to evaluate in each dimension
np = 150;

for c0=1:2,
    switch c0
        case 1,
            [G,w] = freqz(1,poly([-0.9 0.9]),nz);
            br = [-2 4]; % Range of b
            vw = [-70 5]; % View
        case 2,
            [G,w] = freqz([0.05 -0.4],[1 -1.1314 0.25],nz);
            br = [-1 1]; % Range of b
            vw = [-85 2]; % View
    end;

    a = linspace(-0.99,0.99,np);
    b = linspace(br(1),br(2),np);
    Po = inf;
    P = zeros(np,np); % Memory allocation
    for c1=1:np,
        for c2=1:np,
            [H,w] = freqz(b(c2),[1 -a(c1)],nz);
            P(c1,c2) = sum(abs(G).^2) -2*sum(real(G.*conj(H))) + sum(abs(H).^2);
            P(c1,c2) = P(c1,c2)/sum(abs(G).^2);
            if abs(P(c1,c2))<Po,
                Po = abs(P(c1,c2));
                ao = a(c1);
                bo = b(c2);
            end;
        end;
    end;
```
Example 2: MATLAB Code Continued

```matlab
nl = 0; % No. local minima
for c1=2:np-1,
    for c2=2:np-1,
        if P(c1,c2)<min([P(c1+1,c2);P(c1-1,c2);P(c1,c2+1);P(c1,c2-1)]) & P(c1,c2)~=Po,
            nl = nl + 1;
            al(nl) = a(c1);
            bl(nl) = b(c2);
            Pl(nl) = P(c1,c2);
        end;
    end;
    al = al(1:nl);
    bl = bl(1:nl);
    Pl = Pl(1:nl);
end;
```

Example 2: MATLAB Code Continued

```matlab
figure;
subplot(2,1,1);
for c1=1:length(al),
    [H,w] = freqz(bl(c1),[1 -al(c1)],nz);
    hl = plot(w,abs(H),'r');
    hold on;end;
[H,w] = freqz(bo,[1 -ao],nz);
[h,l] = plot(w,abs(G),'g',w,abs(H),'b');
set(h,'LineWidth',1.5);hold off;
set(get(gca,'XLabel'),'Interpreter','LaTeX');set(get(gca,'YLabel'),'Interpreter','LaTeX');ylabel('$\angle H(e^{j\omega})$');xlabel('$\omega$ (radians/sample)');xlim([0 pi]);legend([h;hl(1)],'Actual','Optimal','Local Minima');
subplot(2,1,2);
for c1=1:length(al),
    [H,w] = freqz(bl(c1),[1 -al(c1)],nz);
    hl = plot(w,angle(H),'r');hold on;end;
[H,w] = freqz(bo,[1 -ao],nz);
[h,l] = plot(w,angle(G),'g',w,angle(H),'b');
set(h,'LineWidth',1.5);hold off;
set(get(gca,'XLabel'),'Interpreter','LaTeX');set(get(gca,'YLabel'),'Interpreter','LaTeX');ylabel('$\angle H(e^{j\omega})$');xlabel('$\omega$ (radians/sample)');xlim([0 pi]);
end;
```

Example 2: MATLAB Code Continued

```matlab
figure;
subplot(2,1,1);
for c1=1:length(al),
    [B,W] = freqz(bl(c1),[1 -al(c1)],nz);
    h1 = plot(W,abs(B),'r');
    hold on;
end;
[B,W] = freqz(bo,[1 -ao],nz);
h1 = plot(W,abs(B),'g',W,abs(B),'b');
set(h1,'LineWidth',1.5);hold off;
set(get(gca,'XLabel'),'Interpreter','LaTeX');set(get(gca,'YLabel'),'Interpreter','LaTeX');ylabel('$\angle H(e^{j\omega})$');xlabel('$\omega$ (radians/sample)');xlim([0 pi]);legend([h1;hl(1)],'Actual','Optimal','Local Minima');
subplot(2,1,2);
for c1=1:length(al),
    [B,W] = freqz(bl(c1),[1 -al(c1)],nz);
    h1 = plot(W,angle(B),'r');hold on;end;
[B,W] = freqz(bo,[1 -ao],nz);
h1 = plot(W,angle(B),'g',W,angle(B),'b');
set(h1,'LineWidth',1.5);hold off;
set(get(gca,'XLabel'),'Interpreter','LaTeX');set(get(gca,'YLabel'),'Interpreter','LaTeX');ylabel('$\angle H(e^{j\omega})$');xlabel('$\omega$ (radians/sample)');xlim([0 pi]);
end;
```
Optimal Linear MMSE Estimator

If \( R \) is positive definite (and therefore invertible),
\[
P_e(c) = P_y - c^H d - d^H c + c^H R c
\]
\[
= P_y + (c^H) R (c - R^{-1} d) - d^H c
\]
\[
= P_y + (c^H - d^H R^{-1}) R (c - R^{-1} d) - d^H R^{-1} d
\]
\[
= P_y - d^H R^{-1} d + (c - R^{-1} d)^H R (c - R^{-1} d)
\]
\[
= P_y - d^H R^{-1} d + (Rc - d)^H R^{-1} (Rc - d)
\]
- There are several approaches to finding the optimal solution
- Completing the square is one of the most general, elegant, and insightful
- Only the third term depends on \( c \)

Optimal Linear MMSE Estimator Continued

\[
P(c) = P_y - d^H R^{-1} d + (Rc - d)^H R^{-1} (Rc - d)
\]

In general, an optimal solution is any solution to the normal equations
\[
Rc_o = d
\]
If \( R \) is invertible,
\[
c_o = R^{-1} d
\]
\[
\hat{y}_o(n) = c_o^H x(n)
\]
\[
P_o \triangleq P_e(c_o)
\]
\[
= P_y - d^H R^{-1} d
\]
\[
= P_y - d^H c_o
\]
- By the MSE criterion, this is the optimal solution
- Can solve exactly in a predictable number of operations
- Only requires the second order moments of \( y \) and \( x(n) \)

Normalized Mean Square Error

The normalized mean square error (MMSE) is defined as
\[
P_o \triangleq P(c_o)
\]
\[
\xi \triangleq \frac{P_o}{P_y} = 1 - \frac{P_o}{P_y}
\]
It has the nice property that it is bounded
\[
0 \leq \xi \leq 1
\]
- \( \xi = 0 \) when the estimates are exact, \( \hat{y}(n) = y(n) \)
- \( \xi = 1 \) when \( x(n) \) is uncorrelated with \( y(n) \), \( d = 0 \)
- Can loosely be interpreted as the square of a correlation coefficient
- Unlike MSE, is invariant to the scale of \( y(n) \) and \( x(n) \)

Principal Component Analysis

Additional insights can be gained from studying the shape of the error surface.
Let us perform an eigenvalue decomposition of \( R \),
\[
R = Q \Lambda Q^H = \sum_{i=1}^{M} \lambda_i q_i q_i^H
\]
\[
\Lambda = Q^H R Q
\]
\[
Q Q^H = Q^H Q = I
\]
\[
\Lambda = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}
\]
\( Q \) is unitary (consists of orthonormal vectors)
\[
Q = [q_1 \ q_2 \ \cdots \ q_M]
\]
where
\[
q_i^H q_j = \delta_{ij}
\]
Rotated Residual Vector

\[ R = Q\Lambda Q^H \]

Let us define

\[ \hat{c} = c - c_o \]

Then

\[
P_e(c) = P_o - d^H c_o + (c - R^{-1} d)^H R (c - R^{-1} d)
= P_o - d^H c_o + (c_o + \hat{c} - R^{-1} d)^H R (c_o + \hat{c} - R^{-1} d)
= P_o + \hat{c}^H R \hat{c}
= P_o + \hat{c}^H Q \Lambda Q^H \hat{c}
= P_o + (Q^H \hat{c})^H \Lambda (Q^H \hat{c})
= P_o + \hat{c}'^H \Lambda \hat{c}'
\]

where

\[ \hat{c}' \triangleq Q^H \hat{c} \]

Rotated Residual Vector Length

This linear transformation does not alter the length of \( c \)

\[
||\hat{c}'||^2 = \hat{c}'^H \hat{c}'
= (Q^H \hat{c})^H (Q^H \hat{c})
= \hat{c}^H Q Q^H \hat{c}
= \hat{c}^H \hat{c}
\]

It is more convenient to work with the rotated residual because it simplifies the expression for the MSE

\[
P_e(c) = P_o + \hat{c}'^H \Lambda \hat{c}'
= P_o + \sum_{i=1}^{M} \lambda_i |\hat{c}'_i|^2
\]

Thus the eigenvalues of \( R \) determine how much the MSE is increased by deviations from \( c_o \).

Example 3: Nonlinear Error Performance Surface

Plot the error surface contours and the principal axes of the ellipse.
Occasionally it is useful to work with a generalization of the problem. Specifically, let us define an inner product space such that two vectors $x$ and $y$

$$\langle x, y \rangle \triangleq \mathbb{E} [x^* y]$$

$$||x||^2 \triangleq \langle x, x \rangle = \mathbb{E} [|x|^2] < \infty$$

The Cauchy-Schwartz inequality in this case is given by

$$|\langle x, y \rangle|^2 \leq ||x|| \ |y||$$

though the proof is somewhat involved

Two vectors in this space are said to be orthogonal if

$$\langle x, y \rangle = 0$$

The Projection Theorem

A projection of $y$ onto a linear space spanned by all possible linear combinations of the observed random variables $x$ is the unique element $\hat{y}$ such that

$$\langle y - \hat{y}_P, x_k \rangle = 0 \text{ for all } x_k$$

$$\mathbb{E} [(y - \hat{y}_P) x_k] = 0$$

In other words, the residual or error is orthogonal (uncorrelated) to all of the RVs in $x$.

**Projection Theorem**

The projection theorem states that $||e(n)||^2$ is minimized when $\hat{y}(n)$ is the projection if $y(n)$ onto the linear space spanned by $x$. Mathematically,

$$||y - \hat{y}_P|| \leq ||y - c^H x||$$

Thus

$$\hat{y}_P = \hat{y}_o = c_0^H x$$
Power Decomposition

Due to orthogonality, you should also be able to show that

\[ P_y = P_o + P_y \]

The signal power composed of the power of the estimate plus the power of the error

Projection Theorem and Orthogonality

The most important consequence of the projection theorem is that it indicates the observed RVs \( x \) are orthogonal to the error. We can use this to solve for the parameter vector

\[
E[x(y - \hat{y}_o)^H] = 0 \\
E[x(y - c_o^H x)^H x] = 0 \\
E[xy^* - xx^H c_o] = 0 \\
E[xx^H c_o] = E[xy^*] \\
Rc_o = d
\]

- Thus we can obtain the normal equations by solving for the coefficients that make the observed RVs orthogonal to the error
- The projection theorem tells us these are the same coefficients that minimize the MSE

Using the Projection Theorem

Note that we can also then simplify our expression for MSE

\[
P_o = E[|e(n)|^2] \\
= E[(y - \hat{y}_o)^H (y - \hat{y}_o)] \\
= E[(y - \hat{y}_o)^H (y - c_o^H x)] \\
= E[(y - \hat{y}_o)^H y] \\
= E[(y - c_o^H x)^H y] \\
= P_y - E[xx^H c_oy] \\
= P_y - E[xx^H y]c_o \\
= P_y - d^H c_o
\]

Geometric Interpretations

- Our book and others make an explicit diagram of the orthogonal vectors
- This is a conceptual diagram only
- Note that \( x(n)e_o(n) \neq 0 \) in general

\[
E[x(n)e_o(n)] = 0
\]

- I think it is easier to simply understand the orthogonality equation

\[
E[xe_o^*] = 0
\]
Summary

- We will only discuss linear estimators for most of the term
  \( \hat{y} = c^H x \)
- Our objective criterion is \( \text{MSE} = P_c = \mathbb{E}[||y - \hat{y}||^2] \)
- Many advantages of this approach
  - Solution only depends on second order moments of the joint distribution of \( x \) and \( y \)
  - Error surface is quadratic
  - Unique minimum that can be found by solving the linear normal equations
  - The error is orthogonal (uncorrelated) with the observed RVs, \( x \)
  - If \( x \) and \( y \) are jointly Gaussian, the linear estimator is optimal out of all possible estimators (including nonlinear estimators)