Chapter 3

Representation and Approximation in Vector Spaces

Any good mathematical commodity is worth generalizing.
— Michael Spivak
Calculus on Manifolds

3.1 The approximation problem in Hilbert space

Let \((S, \|\cdot\|)\) be a normed linear vector space for some norm \(\|\cdot\|\). Let \(T = \{p_1, p_2, \ldots, p_m\} \subset S\) be a set of linearly independent vectors in a vector space \(S\) and let \(V = \text{span}(T)\). The analysis problem is this: given a vector \(x \in S\), find the coefficients \(c_1, c_2, \ldots, c_m\) so that

\[
\hat{x} = c_1p_1 + c_2p_2 + \cdots + c_mp_m
\]  

(3.1)

approximates \(x\) as closely as possible. The \(^\wedge\) (caret) indicates that this is (or may be) an approximation. That is, we wish to write

\[
x = \hat{x} + e = c_1p_1 + c_2p_2 + \cdots + c_mp_m + e,
\]

where \(e\) is the approximation error, so that

\[
\|x - \hat{x}\| = \|e\|
\]

is as small as possible. The problem is diagrammed in figure 3.1 for \(m = 2\). Of course,

![Figure 3.1: The approximation problem](image)

if \(x \in V\) then it is possible to find coefficients so that \(\|x - \hat{x}\| = 0\). The particular norm chosen in performing the minimization affects the analytic approach to the problem and the
The Approximation Problem in Hilbert Space

If the $l_1$ (or $L_1$) norm is chosen, then the analysis involves absolute values, which makes an analytical solution involving derivatives difficult. If the $l_\infty$ (or $L_\infty$) norm is chosen, the analysis may involve derivatives of the max function, which is also difficult. If the $l_2$ (or $L_2$) norm is chosen, many of the analytical difficulties disappear. The norm is the induced norm, and the properties of the projection theorem can be used to formulate the solution. Alternatively, the solution can be obtained using calculus techniques. (Actually, for problems posed using the $l_p$ norms, a generalization of the projection theorem can be used, optimizing in Banach space rather than Hilbert space, but this lies beyond the scope of this book.) Choosing the $l_2$ norm allows familiar Euclidean geometry to be used to develop insight. The approximation problem when the induced norm is used (for example, either an $l_2$ or $L_2$ norm) is known as the Hilbert space approximation problem.

To develop geometric insight into the approximation problem, the analysis formulas are presented by starting with the approximation problem with one element in $T$, aided by a key observation: the error is orthogonal to the data. The analysis is then extended to two dimensions, then to arbitrary dimensions. We will begin first with geometric plausibility and calculus, then prove the result using the Cauchy–Schwarz inequality.

To begin, let $T \in \mathbb{R}^2$ consist of only one vector, $T = \{p_1\}$. For a vector $x \in \mathbb{R}^2$, we wish to represent $x$ as a linear combination of $T$,

$$x = c_1 p_1 + e,$$

in such a way as to minimize the norm of the approximation error $\|e\|$. In this simplest case, there is only the parameter $c_1$ to identify. The situation is illustrated in figure 3.2(a). If the $l_2$ or $L_2$ norm is used, it may be observed geometrically that the error is minimized when the error is orthogonal to $V$; that is, when the error is orthogonal to the data that forms our estimate. Written mathematically, the norm of the error $\|e\|$ is minimized when

$$e \perp p_1,$$

or

$$\langle x - c_1 p_1, p_1 \rangle = 0.$$

Using the properties of inner products,

$$c_1 = \frac{\langle x, p_1 \rangle}{\|p_1\|^2}.$$  \hspace{1cm} (3.2)

Geometrically, the quantity

$$\frac{\langle x, p_1 \rangle}{\|p_1\|^2}$$

(a) One vector in $T$  \hspace{1cm} (b) Two vectors in $T$

Figure 3.2: Approximation with one and two vectors
is the projection of the vector $x$ in the direction of $p_1$; it is the length of the shadow that $x$ casts onto $p_1$ (expressed as a proportion of the length of $p_1$).

The same approximation formula may also be obtained by calculus. We find $c_1$ to minimize

$$
\|x - c_1 p_1\|_2^2 = (x - c_1 p_1, p - c_1 x_1)
$$

by taking the derivative with respect to $c_1$ and equating the result to zero. This gives the same answer as (3.2).

Continuing our development, when $T$ contains two vectors we can write the approximation as

$$
x = c_1 p_1 + c_2 p_2 + e.
$$

Figure 3.2(b) illustrates the concept for vectors in $\mathbb{R}^3$. It is clear from this figure that if Euclidean distance is used, the error is orthogonal to the data $p_1$ and $p_2$. This gives the following orthogonality conditions:

$$
\langle x - (c_1 p_1 + c_2 p_2), p_1 \rangle = 0,
$$
$$
\langle x - (c_1 p_1 + c_2 p_2), p_2 \rangle = 0.
$$

Expanding these using the properties of inner products gives

$$
\langle x, p_1 \rangle = c_1 \langle p_1, p_1 \rangle + c_2 \langle p_2, p_1 \rangle,
$$
$$
\langle x, p_2 \rangle = c_1 \langle p_1, p_2 \rangle + c_2 \langle p_2, p_2 \rangle,
$$

which can be written more concisely in matrix form as

$$
\begin{bmatrix}
\langle p_1, p_1 \rangle & \langle p_2, p_1 \rangle \\
\langle p_1, p_2 \rangle & \langle p_2, p_2 \rangle
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix} =
\begin{bmatrix}
\langle x, p_1 \rangle \\
\langle x, p_2 \rangle
\end{bmatrix}.
$$

(3.3)

Solution of this matrix equation provides the desired coefficients.

**Example 3.1.1** Suppose $x = [1, 2, 3]^T$, $p_1 = [1, 1, 0]^T$, and $p_2 = [2, 1, 0]^T$. It is clear that

$$
\hat{x} = c_1 p_1 + c_2 p_2
$$

cannot be an exact representation of $x$ since there is no way to match the third element. Using (3.3), we obtain

$$
\begin{bmatrix}
2 & 3 \\
3 & 5
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix} =
\begin{bmatrix}
3 \\
4
\end{bmatrix}.
$$

This can be solved to give

$$
c_1 = 3 \quad c_2 = -1.
$$

Then the approximation vector is

$$
\hat{x} = c_1 p_1 + c_2 p_2 = 3[1, 1, 0]^T - [2, 1, 0]^T = [1, 2, 0]^T.
$$

Note that the approximation $\hat{x}$ is the same as $x$ in the first two coefficients. The vector has been projected onto the plane formed by the vectors $p_1$ and $p_2$. The error in this case has length 3.

Jumping now to higher numbers of vectors, what we can do for two vectors in $T$, we can do for $m$ ingredient vectors. We approximate $x$ as

$$
x = \sum_{i=1}^{m} c_i p_i + e = \hat{x} + e
$$
to minimize \( \| e \| = \| x - \hat{x} \| \). If the norm used is the \( l_2 \) or \( L_2 \) norm, this is the \textit{linear least-squares} problem. Whenever the norm measuring the approximation error \( \| e \| \) is induced from an inner product, we can express the minimization in terms of an orthogonality condition: the minimum-norm error must be orthogonal to each vector \( p_j \):

\[
\left( x - \sum_{i=1}^{m} c_i p_i, p_j \right) = 0, \quad j = 1, 2, \ldots, m.
\]

This gives \( m \) equations in the \( m \) unknowns, which may be written as

\[
\begin{bmatrix}
\langle p_1, p_1 \rangle & \langle p_2, p_1 \rangle & \cdots & \langle p_m, p_1 \rangle \\
\langle p_1, p_2 \rangle & \langle p_2, p_2 \rangle & \cdots & \langle p_m, p_2 \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle p_1, p_m \rangle & \langle p_2, p_m \rangle & \cdots & \langle p_m, p_m \rangle 
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_m 
\end{bmatrix}
= 
\begin{bmatrix}
\langle x, p_1 \rangle \\
\langle x, p_2 \rangle \\
\vdots \\
\langle x, p_m \rangle 
\end{bmatrix}.
\]

We define the vector

\[
p = 
\begin{bmatrix}
\langle x, p_1 \rangle \\
\langle x, p_2 \rangle \\
\vdots \\
\langle x, p_m \rangle 
\end{bmatrix}
\]

as the \textit{cross-correlation vector}, and

\[
c = 
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_m 
\end{bmatrix}
\]

as the vector of coefficients. Then (3.4) can be written as

\[Rc = p,
\]

where \( R \) is the matrix of inner products in (3.4). Equations of this form are known as the \textbf{normal equations}. Since the solution minimizes the square of the error, it is known as a \textit{least-square} or \textit{minimum mean-square} solution, depending on the particular inner product used.

\subsection{3.1.1 The Grammian matrix}

The \( m \times m \) matrix

\[
R = 
\begin{bmatrix}
\langle p_1, p_1 \rangle & \langle p_2, p_1 \rangle & \cdots & \langle p_m, p_1 \rangle \\
\langle p_1, p_2 \rangle & \langle p_2, p_2 \rangle & \cdots & \langle p_m, p_2 \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle p_1, p_m \rangle & \langle p_2, p_m \rangle & \cdots & \langle p_m, p_m \rangle 
\end{bmatrix}
\]

in the left hand side of (3.4) is said to be the \textbf{Grammian} of the set \( T \). Since the \((i, j)\)th element of the matrix is

\[R_{ij} = \langle p_j, p_i \rangle,
\]

it follows that the Grammian is a Hermitian symmetric matrix; that is,

\[R^H = R
\]

(where \( ^H \) indicates conjugate-transpose). Some implications of the Hermitian structure are examined in section 6.2. Solution of (3.4) requires that \( R \) be invertible. The following theorem determines conditions under which \( R \) is invertible. Recall that a matrix \( R \) for which

\[x^H Rx > 0
\]
Box 3.1: Positive-definite matrices

We will encounter several times in the course of this book the notion of positive-definite matrices. We collect together here several important facts related to positive definite matrices.

**Definition 3.1** A matrix $A$ is said to be positive definite (PD) if $x^H Ax > 0$ for all $x \neq 0$. This is sometimes denoted as $A > 0$. (Caution: the notation $A > 0$ is also sometimes used to indicate that all the elements of $A$ are greater than zero, which is not the same as being PD.) If $x^H Ax \geq 0$ for all $x$, then $A$ is positive semidefinite (PSD). If $>$ is replaced by $\leq$, the matrix is said to be negative definite (ND), and if $\geq$ is replaced by $\leq$, the matrix is negative semidefinite (NSD).

Here are some properties of positive-definite (or semidefinite) matrices.

1. All diagonal elements of a PD (PSD) matrix are positive (nonnegative). (Caution: this does not mean that positive diagonal elements imply that a matrix is PD).
2. A Hermitian matrix $A$ is PD (PSD) if and only if all of the eigenvalues are positive (nonnegative). Hence, a PD matrix has a positive determinant. Hence, a PD matrix is invertible.
3. A Hermitian matrix $P$ is PD if and only if all principal minors are positive.
4. If $A$ is PD, then the pivots obtained in the LU factorization are positive.
5. If $A > 0$ and $B > 0$, then $A + B > 0$. If $A$ is PD and $B$ is PSD, then $A + B$ is PD.
6. A Hermitian PD matrix $A$ can be factored as $A = B^H B$ (using the Cholesky factorization, for instance), where $B$ is full rank. This is a matrix square root.

for any nonzero vector $x$ is said to be positive-definite (see box 3.1). An important aspect of positive-definite matrices is that they are always invertible. If $R$ is such that

$$x^H Rx \geq 0$$

for any nonzero vector $x$, then $R$ is said to be positive-semidefinite.

**Theorem 3.1** A Grammian matrix $R$ is always positive-semidefinite (that is, $x^H Rx \geq 0$ for any $x \in \mathbb{C}^m$). It is positive-definite if and only if the vectors $p_1, p_2, \ldots, p_m$ are linearly independent.

**Proof** Let $y = [y_1, y_2, \ldots, y_m]^T$ be an arbitrary vector. Then

$$y^H Ry = \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \langle p_j, p_i \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \langle y_j p_j, y_i p_i \rangle$$

$$= \left( \sum_{j=1}^{m} y_j p_j, \sum_{j=1}^{m} y_j p_i \right) = \left\| \sum_{j=1}^{m} y_j p_j \right\|^2 \geq 0. \quad (3.8)$$

Hence $R$ is positive-semidefinite.
If $R$ is not positive-definite, then there is a nonzero vector $y$ such that
\[ y^H R y = 0, \]
so that (by (3.8))
\[ \sum_{i=1}^{m} y_i p_i = 0; \]
thus, the $\{p_i\}$ are linearly dependent.

Conversely, if $R$ is positive-definite, then
\[ y^H R y > 0 \]
for all nonzero $y$ and by (3.8)
\[ \sum_{i=1}^{m} y_i p_i \neq 0. \]
This means that the $\{p_i\}$ are linearly independent.

As a corollary to this theorem, we get another proof of the Cauchy–Schwarz inequality. The $2 \times 2$ Grammian
\[
R = \begin{bmatrix}
\langle x, x \rangle & \langle x, y \rangle \\
\langle y, x \rangle & \langle y, y \rangle
\end{bmatrix}
\]
is positive-semidefinite, which means that its determinant is nonnegative:
\[
\langle x, x \rangle \langle y, y \rangle - \langle x, y \rangle \langle y, x \rangle \geq 0,
\]
which is equivalent to (2.13).

The concept of using orthogonality for the Euclidean inner product to find the minimum norm solution generalizes to any induced norm and its associated inner product.

If the set of vectors $\{p_1, p_2, \ldots, p_m\}$ are orthogonal, then the Grammian in (3.7) is diagonal, significantly reducing the amount of computation required to find the coefficients of the vector representation. In this case, the coefficients are obtained simply by
\[
c_j = \frac{\langle x, p_j \rangle}{\langle p_j, p_j \rangle}. \tag{3.9}
\]
Each coefficient uses the same projection formula that was used in (3.3) for a single dimension. The coefficients can also be readily interpreted: for orthogonal vectors, the coefficient of each vector indicates the strength of the vector component in the signal representation.

### 3.2 The Orthogonality Principle

The **orthogonality principle** for least-squares (LS) optimization introduced in section 3.1 is now formalized.

**Theorem 3.2 (The orthogonality principle)** Let $p_1, p_2, \ldots, p_m$ be data vectors in a vector space $S$. Let $x$ be any vector in $S$. In the representation
\[
x = \sum_{i=1}^{m} c_i p_i + e = \hat{x} + e,
\]
the induced norm of the error vector $\|e\|$ is minimized when the error $e = x - \hat{x}$ is orthogonal to each of the data vectors,
\[
\langle x - \sum_{i=1}^{m} c_i p_i, p_j \rangle = 0 \quad j = 1, 2, \ldots, m.
\]
Proof One proof relies on the projection theorem, theorem 2.8, with the observation that 
\( V = \text{span}(p_1, p_2, \ldots, p_m) \) is a subspace of \( S \). We present a more direct proof using the Cauchy-Schwarz inequality.

In the case that \( x \in \text{span}(p_1, p_2, \ldots, p_m) \), the error is zero and hence is orthogonal to the data vectors. This case is therefore trivial and is excluded from what follows.

If \( x \notin \text{span}(p_1, p_2, \ldots, p_m) \), let \( y \) be a fixed vector that is orthogonal to all of the data vectors, 
\[
\langle y, p_i \rangle = 0 \quad i = 1, 2, \ldots, m,
\]
such that
\[
x = \sum_{i=1}^{m} a_i p_i + y
\]
for some set of coefficients \( \{a_1, a_2, \ldots, a_m\} \). Let \( e \) be a vector satisfying
\[
x = \sum_{i=1}^{m} c_i p_i + e
\]
for some set of coefficients \( \{c_1, c_2, \ldots, c_m\} \). Then by the Cauchy-Schwarz inequality,
\[
\|e\|^2 \|y\|^2 \geq |\langle e, y \rangle|^2 \quad \text{(Cauchy-Schwarz)}
\]
\[
= \left| \langle x, y \rangle - \left( \sum_{i=1}^{m} c_i p_i, y \right) \right|^2
\]
\[
= |\langle x, y \rangle|^2 \quad \text{(orthogonality of } y) \). \quad (3.11)
\]
The lower bound is independent of the coefficients \( \{c_i\} \), and hence no set of coefficients can make the bound smaller. By the equality condition for the Cauchy-Schwarz inequality, the lower bound is achieved—implying the minimum \( \|e\| \)—when
\[
e = \alpha y
\]
for some scalar \( \alpha \). Since \( e \) must satisfy (3.10), it must be the case that \( e = y \), hence the error is orthogonal to the data.$\square$

When \( c \) is obtained via the principle of orthogonality, the optimal estimate
\[
\hat{x} = \sum_{i=1}^{m} c_i p_i
\]
is also orthogonal to the error \( e = x - \hat{x} \), since it is a linear combination of the data vectors \( \{p_i\} \). Thus,
\[
\langle \hat{x}, e \rangle = 0. \quad (3.12)
\]

3.2.1 Representations in infinite-dimensional space

If there are an infinite number of vectors in \( T = \{p_1, p_2, \ldots, \} \), then the representation
\[
\hat{x} = \sum_{i=1}^{\infty} c_i p_i
\]
is suspect, because a linear combination is defined, technically, only in terms of a finite
sum. The convergence of this infinite sum must therefore be examined carefully. However, if \( T \) is an orthonormal set, then the representation can be shown to converge.

### 3.3 Error minimization via gradients

While the orthogonality theorem is used principally throughout this chapter as the geometrical basis for finding a minimum error approximation under an induced norm, it is pedagogically worthwhile to consider another approach based on gradients, which reaffirms what we already know but demonstrates the use of some new tools.

Minimizing \( \|e\|^2 \) for the induced norm in

\[
x = \sum_{i=1}^{m} c_i p_i + e
\]

requires minimizing

\[
J(c) = \left\langle x - \sum_{j=1}^{m} c_j p_j, x - \sum_{i=1}^{m} c_i p_i \right\rangle
\]

\[
= \langle x, x \rangle - 2 \text{Re} \left( \sum_{i=1}^{m} c_i (x, p_i) \right) + \sum_{i=1}^{m} \sum_{j=1}^{m} c_j \overline{c}_i (p_j, p_i). \tag{3.13}
\]

Using the vector notations defined in (3.5), (3.6), and (3.7), we can write (3.13) as

\[
J(c) = \|x\|^2 - 2 \text{Re}(e^H p) + e^H Rc. \tag{3.14}
\]

Gradient formulas appropriate for this optimization are presented in section E.1.1 of appendix E. In particular, the following gradient formulas are derived:

\[
\frac{\partial}{\partial e} d^H c = 0 \quad \frac{\partial}{\partial e} c^H d = d \quad \frac{\partial}{\partial e} \text{Re}(e^H d) = \frac{1}{2} d \quad \frac{\partial}{\partial e} e^H Rc = Rc.
\]

Taking the gradient of (3.14) using the last two of these, we obtain

\[
\frac{\partial}{\partial e} (\|x\|^2 - 2 \text{Re}(e^H p) + e^H Rc) = -p + Rc. \tag{3.15}
\]

Equating this result to zero we obtain

\[
Rc = p,
\]

giving us again the normal equations.

To determine whether the extremum we have obtained by the gradient is in fact a minimum, we compute the gradient a second time. We have the Hessian matrix

\[
\frac{\partial}{\partial e} Rc = R,
\]

which is a positive-semidefinite matrix, so the extremum must be a minimum.

Restricting attention for the moment to real variables, consider the plot of the norm of the error \( J(c) \) as a function of the variables \( c_1, c_2, \ldots, c_m \). Such a plot is called an error surface. Because \( J(c) \) is quadratic in \( c \) and \( R \) is positive semidefinite, the error surface is a parabolic bowl. Figure 3.3 illustrates such an error surface for two variables \( c_1 \) and \( c_2 \). Because of its parabolic shape, any extremum must be a minimum, and is in fact a global minimum.
3.4 Matrix representations of least-squares problems

While vector space methods apply to both infinite- and finite-dimensional vectors (signals), the notational power of matrices can be applied when the basis vectors are finite dimensional. The linear combination of the finite set of vectors \( \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m\} \) can be written as

\[
\mathbf{x} = \sum_{i=1}^{m} c_i \mathbf{p}_i = [\mathbf{p}_1 \, \mathbf{p}_2 \, \cdots \, \mathbf{p}_m] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}.
\]

This is the linear combination of the columns of the matrix \( \mathbf{A} \) defined by

\[
\mathbf{A} = [\mathbf{p}_1 \, \mathbf{p}_2 \, \cdots \, \mathbf{p}_m],
\]

which we compute by

\[
\mathbf{x} = \mathbf{A} \mathbf{c}.
\]

The approximation problem can be stated as follows:

Determine \( \mathbf{c} \) to minimize \( \|\mathbf{e}\|_2^2 \) in the problem \( \mathbf{x} = \mathbf{A} \mathbf{c} + \mathbf{e} = \hat{\mathbf{x}} + \mathbf{e} \). (3.16)

The minimum \( \|\mathbf{e}\|_2^2 = \|\mathbf{x} - \mathbf{A} \mathbf{c}\|_2^2 \) occurs when \( \mathbf{e} \) is orthogonal to each of the vectors

\[
\langle \mathbf{x} - \mathbf{A} \mathbf{c}, \mathbf{p}_j \rangle = 0, \quad j = 1, 2, \ldots, m.
\]

Stacking these orthogonality conditions, we obtain

\[
\begin{bmatrix}
\mathbf{p}_1^H \\
\mathbf{p}_2^H \\
\vdots \\
\mathbf{p}_m^H
\end{bmatrix}
\begin{bmatrix}
\mathbf{x} - \mathbf{A} \mathbf{c} = \mathbf{0}.
\end{bmatrix}
\]

Recognizing that the stack of vectors is simply \( \mathbf{A}^H \), we obtain

\[
\mathbf{A}^H \mathbf{A} \mathbf{c} = \mathbf{A}^H \mathbf{x}.
\] (3.17)
The matrix $A^H A$ is the Grammian $R$, and the vector $A^H x$ is the cross-correlation $p$. We can write (3.17) as

$$Rc = A^H x = p.$$  
(3.18)

These equations are the normal equations. Then the optimal (least-squares) coefficients are

$$e = (A^H A)^{-1} A^H x = R^{-1} p$$  
(3.19)

By theorem 3.1, $A^H A$ is positive definite if the $p_1, \ldots, p_m$ are linearly independent. The matrix $(A^H A)^{-1} A^H$ is called a pseudoinverse of $A$, and is often denoted $A^\dagger$. More is said about pseudoinverses in section 4.9. While (3.19) provides an analytical prescription for the optimal coefficients, it should rarely be computed explicitly as shown, since many problems are numerically unstable (subject to amplification of roundoff errors). Numerical stability is discussed in section 4.10. Stable methods for computing pseudoinverses are discussed in sections 5.3 and 7.3. In MATLAB, the pseudoinverse may be computed using the command pinv.

Using (3.19), the approximation is

$$\hat{x} = Ac = A(A^H A)^{-1} A^H x$$  
(3.20)

The matrix $P = A(A^H A)^{-1} A^H$ is a projection matrix, which we encountered in section 2.13. The matrix $P$ projects onto the range of $A$. Consider geometrically what is taking place: we wish to solve the equation $Ac = x$, but there is no exact solution, since $x$ is not in the range of $A$. So we project $x$ orthogonally down onto the range of $A$, and find the best solution in that range space. The idea is shown in figure 3.4.

![Figure 3.4: Projection solution](image)

A useful representation of the Grammian $R = A^H A$ can be obtained by considering $A$ as a stack of rows,

$$A = \begin{bmatrix} q_1^H \\ q_2^H \\ \vdots \\ q_n^H \end{bmatrix},$$  
(3.21)

so that $A^H = [q_1, q_2, \ldots, q_n]$ and

$$A^H A = [q_1 \ q_2 \ \cdots \ q_n] \begin{bmatrix} q_1^H \\ q_2^H \\ \vdots \\ q_n^H \end{bmatrix} = \sum_{i=1}^{n} q_i q_i^H .$$  
(3.22)
3.4.1 Weighted least-squares

A weight can also be applied to the data points, reflecting the confidence in the data, as illustrated by the next example. This is naturally incorporated into the inner product. Define a weighted inner product as

\[ (x, y)_w = y^H W x. \]

Then minimizing \( \|e\|_W^2 = \|Ac - x\|_W^2 \) leads to the weighted normal equations

\[ A^H W Ac = A^H W x, \tag{3.23} \]

so the coefficients which minimize the weighted squared error are

\[ c = (A^H W A)^{-1} A^H W x. \tag{3.24} \]

Another approach to (3.24) is to presume that we have a factorization of the weight \( W = S^H S \) (see section 5.2). Then we weight the equation

\[ SAC = Sy. \]

Multiplying through by \((SA)^H\) and solving for \(c\), we obtain

\[ c = ((SA)^H SA)^{-1}(SA)^H Sy, \]

which is equivalent to (3.24).

3.4.2 Statistical properties of the least-squares estimate

The matrix-least squares solution (3.20) has some useful statistical properties. Suppose that the signal \( x \) has the true model according to the equation

\[ x = Ac_0 + e, \tag{3.25} \]

for some “true” model parameter vector \( c_0 \); and that we assume a statistical model for the model error \( e \): assume that each component of \( e \) is a zero-mean, i.i.d. random variable with variance \( \sigma_e^2 \). The estimated parameter vector is

\[ c = (A^H A)^{-1} A^H x. \tag{3.26} \]

This least-squares estimate, being a function of the random vector \( x \), is itself a random vector. We will determine the mean and covariance matrix for this random vector.

**Mean of** \( c \). Substituting the “true” model of (3.25) into (3.26), we obtain

\[ c = (A^H A)^{-1} A^H Ac_0 + (A^H A)^{-1} A^H e \]

\[ = c_0 + (A^H A)^{-1} A^H e. \]

If we now take the expected value of our estimated parameter vector, we obtain

\[ E[c] = E[c_0 + (A^H A)^{-1} A^H e] = c_0, \]

since each component of \( e \) has zero mean. Thus, the expected value of the estimate is equal to the true value. Such an estimate is said to be **unbiased**.

**Covariance of** \( c \). The covariance can be written as

\[ \text{Cov}[c] = E[(c - c_0)(c - c_0)^H] \]

\[ = (A^H A)^{-1} A^H E[ee^H] A(A^H A)^{-1}. \]

Since the components of \( e \) are i.i.d., it follows that \( E[ee^H] = \sigma_e^2 I \), so that

\[ \text{Cov}[c] = \sigma_e^2 (A^H A)^{-1} = \sigma_e^2 R^{-1}. \]
Smallest covariance. Another interesting fact: of all possible unbiased linear estimates, the estimator (3.19) has the “smallest” covariance. Suppose we have another unbiased linear estimator \( \tilde{c} \) given by

\[ \tilde{c} = Lx, \]

where \( E[\tilde{c}] = c_0 \). Using our statistical model (3.25), we obtain

\[ \tilde{c} = LAc_0 + Le. \]

In order for the estimate \( \tilde{c} \) to be unbiased, we must have \( E[\tilde{c}] = c_0 \), so

\[ LA = I. \]

We therefore obtain \( \tilde{c} = c_0 + Le \). The covariance of \( \tilde{c} \) is

\[ \text{Cov}[\tilde{c}] = E[(\tilde{c} - c_0)(\tilde{c} - c_0)^H] = \sigma^2_\epsilon LL^H. \]

We will show that \( LL^H > R^{-1} \), in the sense that the matrix \( LL^H - R^{-1} \) is positive semidefinite. Let

\[ Z = L - R^{-1}A^H. \]

Then for any \( z \),

\[ 0 \leq \|Z^Hz\|^2 = \langle Z^Hz, Z^Hz \rangle = z^HZZ^Hz. \]

But

\[ ZZ^H = LL^H - R^{-1}, \]

where we have used the fact that \( LA = I \). Thus, for any \( z \),

\[ z^H(LL^H - R^{-1})z \geq 0, \]

so \( LL^H - R^{-1} \) is positive semidefinite, or \( R^{-1} \) is a smaller covariance matrix. The estimator \( c \) is said to be a best linear unbiased estimator (BLUE).

3.5 Minimum error in vector-space approximations

In this section we examine how much error is left when an optimal (minimal-norm) solution is obtained. Under the model that

\[ x = \sum_{i=1}^{m} c_i p_i + e, \]

when the coefficients are found so that the estimation error is orthogonal to the data, we have

\[ x = \hat{x} + e_{\text{min}}, \]

where \( e_{\text{min}} \) denotes the minimum achievable error. Taking the squared norm of both sides, we obtain

\[ \|x\|^2 = \|\hat{x}\|^2 + \|e_{\text{min}}\|^2. \] (3.27)
This result, sometimes called the statistician’s Pythagorean theorem, follows because $\hat{x}$ is orthogonal to the minimum-norm error,

$$\langle \hat{x}, e_{\text{min}} \rangle = 0.$$ 

The statistician’s Pythagorean theorem is illustrated in figure 3.5. (See also lemma 2.2.)

![Figure 3.5: Statistician’s Pythagorean theorem](image)

The squared norm of the minimum error is

$$\|e_{\text{min}}\|^2 = \|x\|^2 - \|\hat{x}\|^2.$$ 

When we use the matrix formulation, we can obtain a more explicit representation for the minimum error. Then $\hat{x} = Ac$, so

$$\|\hat{x}\|^2 = c^H A^H A c = c^H R c = c^H p.$$  

(3.28)

where $p$ from (3.18) has been employed. This gives

$$\|e_{\text{min}}\|^2 = x^H x - c^H p.$$ 

Another form for $\|\hat{x}\|^2$ is obtained from (3.20),

$$\|\hat{x}\|^2 = (Ac)^H (Ac) = x^H A (A^H A)^{-1} A^H x.$$  

(3.29)

Then

$$\|e_{\text{min}}\|^2 = x^H x - x^H A (A^H A)^{-1} A^H x$$ 

$$= x^H (I - A (A^H A)^{-1} A^H) x.$$ 

It can be shown (see exercise 3.5-2) that

$$(I - A (A^H A)^{-1} A^H)$$

(3.30)

is a positive-semidefinite matrix, from which we can conclude that $\|e_{\text{min}}\|^2$ is smaller than $\|x\|^2$.

---

**Applications of the orthogonality theorem**

Because a number of vector spaces and inner products can be formulated, the orthogonality principle is used in a variety of applications. The orthogonality theorem provides the foundation for a good part of signal processing theory, since it provides a prescription for an optimum estimator: **in the optimum (least-squares) estimator, the error is orthogonal to the data.** The theorem is applied by defining an inner product, and hence the induced norm, to match the needs of the problem. Under various inner-product definitions, much of approximation theory, estimation theory, and prediction theory can be accommodated. Examples are given in the next several sections.
3.6 Approximation by continuous polynomials

Suppose we want to find the best polynomial approximation of a real continuous function $f(t)$ over an interval $t \in [a, b]$, in the sense that

$$\int_a^b (f(t) - p(t))^2 \, dt$$

is minimized for a polynomial $p(t)$ of degree $m - 1$. The vector space underlying the problem is $S = C[a, b]$. We will (naively) take as basis vectors the functions $\{1, t, t^2, \ldots, t^{m-1}\}$, so that

$$p(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_{m-1} t^{m-1}.$$ 

The optimal coefficients can be determined (for example) directly by calculus, but the orthogonality theorem applies, using the inner product

$$\langle f, g \rangle = \int_a^b f(t)g(t) \, dt.$$ 

Then, using (3.4) we obtain

$$\begin{bmatrix} \langle 1, 1 \rangle & \langle 1, t \rangle & \cdots & \langle 1, t^{m-1} \rangle \\ \langle t, 1 \rangle & \langle t, t \rangle & \cdots & \langle t, t^{m-1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle t^{m-1}, 1 \rangle & \langle t^{m-1}, t \rangle & \cdots & \langle t^{m-1}, t^{m-1} \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} \langle f, 1 \rangle \\ \langle f, t \rangle \\ \vdots \\ \langle f, t^{m-1} \rangle \end{bmatrix}. \tag{3.31}$$

If we take the specific case that the function is to be approximated over the interval $[0, 1]$, then the Grammian matrix in (3.31) can be computed explicitly as

$$\langle t^i, t^j \rangle = \int_0^1 t^{i+j} \, dt = \frac{1}{i+j+1}, \quad i, j = 0, 1, \ldots, m-1,$$

so that

$$R = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{m} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m} & \frac{1}{m+1} & \frac{1}{m+2} & \cdots & \frac{1}{2m} \end{bmatrix}. \tag{3.32}$$

A matrix of this particular form is known as a Hilbert matrix. The Hilbert matrix is famous as a classic example of a matrix that is ill conditioned: as $m$ increases, the matrix becomes ill conditioned exponentially fast, which means (as discussed in section 4.10) that it will suffer from severe numerical problems if $m$ is even moderately large, no matter how it is inverted. Because of this, the particular set of basis functions chosen is not recommended. The use of
the Legendre polynomials described in example 2.15.1, or other orthogonal polynomials, is preferred for polynomial approximation.

**Example 3.6.1** Let \( f(t) = e^t \) and \( m = 3 \). (For only three parameters, the Hilbert matrix (3.32) is still well conditioned.) The vector on the right hand of (3.31) is

\[
b = \begin{bmatrix} e - 1 \\ 1 \\ e - 2 \end{bmatrix},
\]

and the coefficients in (3.31) are computed as

\[
\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = R^{-1}b = \begin{bmatrix} 1.0130 \\ 0.8511 \\ 0.8392 \end{bmatrix}.
\]

The approximating polynomial is

\[
e^t \approx 1.0130 + 0.8511t + 0.8392t^2.
\]

![Figure 3.6: Comparison of LS, WLS, and Taylor series approximations to \( e^t \)](image)

Figure 3.6 shows the absolute error \( |e^t - p(t)| \) for this polynomial for \( t \in [0, 1] \). For comparison, the error we would get by approximating \( e^t \) by the first three terms of the Taylor series expansion,

\[
e^t \approx 1 + t + t^2/2,
\]

is also shown, as is the weighed least-squares (WLS) approximation discussed subsequently. The error in the Taylor series starts small, but increases to a larger value than does the least-squares approximation. (How would the Taylor series have compared if the series had been expanded about the midpoint of the region, at \( t_0 = \frac{1}{2} \)?)

The basis functions of the previous example give rise to the Hilbert matrix as the Grammian. However, a set of orthogonal polynomials can be used that has a diagonal (and hence well-conditioned) Grammian.

Now suppose that for some reason it is more important to get the approximation more correct on the extremes of the interval of approximation. We will denote the approximating polynomial in this case by \( p_w(t) \). To attempt to make the approximation more exact on the
extremes of the interval of approximation, we use a weighted norm
\[ \int_a^b w(t)(f(t) - p_w(t))^2 dt, \]
which is induced from the inner product
\[ \langle f, g \rangle = \int_a^b \sqrt{w(t)} f(t) g(t) dt. \]

**Example 3.6.2** Continuing the example above with \( f(t) = e^t \) over \([0, 1]\), take the weighting function as
\[ w(t) = 10(t - 0.5)^2. \]
Then the Grammian matrix is
\[ R = \begin{pmatrix} \frac{1}{2} \sqrt{5/2} & \frac{1}{4} \sqrt{5/2} & \frac{1}{8} \sqrt{5/2} \\ \frac{1}{4} \sqrt{5/2} & \frac{1}{8} \sqrt{5/2} & \frac{5}{32} \sqrt{5/2} \\ \frac{1}{8} \sqrt{5/2} & \frac{5}{32} \sqrt{5/2} & \frac{1}{64} \sqrt{5/2} \end{pmatrix} \]
and the right hand vector (computed numerically) is
\[ b = [1.38603 \ 0.860513 \ 0.690724]^T. \]
The approximating polynomial is now
\[ p_w(t) = 1.0109 + .8535t + .8415t^2. \]
Figure 3.6 shows the error \( e' - p_w(t) \) and \( e' - p(t) \). As expected, the error is smaller (though only slightly) for \( p_w(t) \) near the endpoints, but larger in between.

As various weightings are imposed, the error at some values of \( t \) is reduced, while error for other values of \( t \) may increase. This raises the following interesting (and important) question: Is there some way to design the approximation so that the maximum error is minimized? This is what \( L_\infty \) approximation is all about:
\[ \min \| f(t) - p(t) \|_\infty. \]
The approximation is chosen so that the maximum error is minimized.

### 3.7 Approximation by discrete polynomials

We can approximate discrete (sampled) data using polynomials in a manner similar to the continuous polynomial approximations of section 3.6 using a set of discrete-time basis functions \( \{1, k, \ldots, k^{m-1}\} \). We desire to fit an \((m - 1)\)st order polynomial through the data points \( x_1, x_2, \ldots, x_n \), so that
\[ x_k \approx p(k), \quad k = 1, 2, \ldots, n, \]
where
\[ p(k) = c_0 + c_1 k + c_2 k^2 + \cdots + c_{m-1} k^{m-1}. \]
The polynomial \( p(k) \) can be written as
\[ p(k) = [1 \ k \ k^2 \ \cdots \ k^{m-1}] \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{m-1} \end{bmatrix}. \]
If \( m = n \) and the \( x_k \) are distinct, then there exists a polynomial, the interpolating polynomial, passing exactly through all \( n \) points. If \( m < n \), then there is probably not a polynomial that will pass through all \( n \) points, in which case we desire to find the polynomial to minimize the squared error,

\[
\sum_{k=1}^{n} |x_k - p(k)|^2.
\]

This can be expressed as a vector norm

\[
\|x - p\|_2,
\]

which is induced from the Euclidean inner product \( \langle x, y \rangle = x^H y \), where

\[
x = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}
\text{ and } \quad p = \begin{bmatrix}
p(1) \\
p(2) \\
\vdots \\
p(n)
\end{bmatrix}.
\]

We can write \( p \) in terms of the coefficients of the polynomial as

\[
p = \begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & 2 & 4 & \ldots & 2^{m-1} \\
1 & 3 & 9 & \ldots & 3^{m-1} \\
\vdots \\
1 & n & n^2 & \ldots & n^{m-1}
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 \\
\vdots \\
c_{m-1}
\end{bmatrix} = P \begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\vdots \\
p_m
\end{bmatrix} = Pa.
\]

The vectors \( p_i, i = 1, 2, \ldots, m \) represent the data in this approximation problem. If \( P \) is square, it is called a Vandermonde matrix, about which more is presented in section 8.4. As with the continuous-time polynomial approximation, there may be better basis functions for this problem from a numerical point of view.

Using this notation, the approximation problem becomes

\[
x = Pe + e,
\]

which is a problem in the same form as (3.2), from which observe that the \( e \) which minimizes \( \|e\|_2^2 \) is

\[
e = (P^T P)^{-1} P^T x.
\]

The approximated vector \( p \) is thus

\[
p = Pe = P(P^T P)^{-1} P^T x.
\]

**Example 3.7.1** We desire to approximate the function

\[
x[k] = \sin(k\pi/7)
\]

using a quadratic polynomial \( (m = 3) \) to obtain the best match for \( k = 1:7 \). The \( P \) matrix is

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 2 & 4 \\
1 & 3 & 9 \\
1 & 4 & 16 \\
1 & 5 & 25 \\
1 & 6 & 36 \\
1 & 7 & 49
\end{bmatrix}
\]
3.8 Linear Regression

From the data in figure 3.8(a), where there are \( n \) points \( x_i, i = 1, 2, \ldots, n \) with each \( x_i = [x_i, y_i]^T \), it would appear that we can approximately fit a line of the form

\[
y_i \approx ax_i + b, \quad i = 1, 2, \ldots, n
\]

(3.33)

for suitably chosen slope \( a \) and intercept \( b \). As stated, this is a linear regression problem; that is, a problem of determining a functional relation between the measured variables \( x_i \) and \( y_i \). Nonlinear regressions are also used, such as the quadratic regression,

\[
y_i \approx a_0 + a_1 x_i + a_2 x_i^2.
\]

(3.34)

Or we may have data vectors \( x_i \in \mathbb{R}^3 \), with \( x_i = [x_i, y_i, z_i]^T \), and we may regress among the points as

\[
z_i \approx ax_i + by_i + c.
\]

(3.35)

Figure 3.8: Data for regression
In all such regression problems, we desire to choose the regression parameters so that the Right Hand Side of the regression equations provides a good representation of the Left Hand Side.

We will consider in detail the linear regression problem (3.33). We can stack the equations to obtain

\[
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{bmatrix} = \begin{bmatrix}
ax_1 + b \\
ax_2 + b \\
\vdots \\
ax_n + b
\end{bmatrix} + \begin{bmatrix}
e_1 \\
e_2 \\
\vdots \\
e_n
\end{bmatrix}
\]

for some error terms \(e_i\). Let

\[
y = [y_1, y_2, \ldots, y_n]^T \quad e = [e_1, e_2, \ldots, e_n]^T \quad c = \begin{bmatrix} a \\ b \end{bmatrix}
\]

and

\[
A = \begin{bmatrix}
x_1 & 1 \\
x_2 & 1 \\
\vdots \\
x_n & 1
\end{bmatrix}
\]

Then (3.36) is of the form

\[
y = Ac + e,
\]

which again is in the form (3.16), so the best (in the least-squares sense) estimate of \(c\) is

\[
c = (A^T A)^{-1} A^T y.
\]

The line found by (3.38) minimizes the sums of the squares of the vertical distances between the data abscissas and the line, as shown in figure 3.8(b). To minimize shortest distances of the data to the interpolating line, the method of total least squares discussed in section 7.7 must be used.

Since \(A^T A\) in (3.38) is a 2 x 2 matrix, explicit closed-form expressions for \(a\) and \(b\) in \(c\) can be found. The slope and intercept (for real data) are

\[
a = \frac{n \sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2},
\]

\[
b = \frac{(\sum_{i=1}^{n} x_i^2)(\sum_{i=1}^{n} y_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i y_i)}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}.
\]

**Example 3.8.1** (Weighted least-squares) Five measurements \((x_i, y_i), i = 1, 2, \ldots, 5\) are made in a system, of which the first three are believed to be fairly accurate, and two are known to be somewhat corrupted by measurement noise. The measurements are

\((1, 2.5) \quad (3, 3.5) \quad (6, 5) \quad (5, 3) \quad (3, 4)\).
From these five measurements, the data are to be fitted to a line according to the model \( y = ax + b \). The measurements stack up in the model equation as

\[
\begin{bmatrix}
1 & 1 \\
3 & 1 \\
6 & 1 \\
5 & 1 \\
3 & 1
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix} =
\begin{bmatrix}
2.5 \\
3.5 \\
5 \\
3 \\
4
\end{bmatrix} + e,
\]

or

\[ Ae = y + e. \]

In finding the best (minimum squared-error) solution to this problem, it is appropriate to weight most heavily those equations which are believed to be the most accurate. Let

\[ W = \text{diag}[10, 10, 10, 1, 1]. \]

Then using (3.24), we can determine the optimal (under the weighted inner product) set of coefficients. Figure 3.9 illustrates the data and the least-squares lines fitted to them. The accurate data are plotted with \( \times \), and the inaccurate data are plotted with \( \circ \). The weighted least-squares line fits more closely (on average) to the more accurate data, while the unweighted least-squares line is pulled off significantly by the inaccurate data at \( x = 5 \).

## 3.9 Least-squares filtering

In the least-squares filter problem, we desire to filter a sequence of input data \( \{f[t]\} \), using a filter with impulse response \( h[t] \) of length \( m \) to produce an output that matches a desired sequence \( \{d[t]\} \) as closely as possible. (Examples in which such a circumstance arises are given in section 1.5, in the context of adaptive filtering.) If we call the output of the filter \( y[t] \), we have the filter expression

\[ y[t] = \sum_{i=0}^{m-1} h[i] f[t-i]. \]
We can write \( d[t] = y[t] + e[t] \), where \( e[t] \) is the error between the filter output and the desired filter output,
\[
d[t] = \sum_{i=0}^{m-1} h[i] f[t - i] + e[t].
\]

We want to choose the filter coefficients \( h[i] \) in such a way that the error between the filter output and the desired signal should be as small as possible; that is, we want to make
\[
e[t] = d[t] - y[t]
\]
small for each \( t \).

When doing least-squares filtering, the criterion of minimal error is that the sum of the squared errors is as small as possible:
\[
\min \sum_{i=i_1}^{i_2} |e[i]|^2,
\]
where \( i_1 \) is the starting index and \( i_2 \) the ending index over which we desire to minimize. The squared norm in (3.40) is induced from the inner product defined by
\[
(x, y) = \sum_{i=i_1}^{i_2} x_i y_i
\]
Letting
\[
y = \begin{bmatrix} y[i_1] \\ y[i_1 + 1] \\ \vdots \\ y[i_2] \end{bmatrix}, \quad h = \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[m - 1] \end{bmatrix}, \quad x = \begin{bmatrix} x[i_1] \\ x[i_1 + 1] \\ \vdots \\ x[i_2] \end{bmatrix},
\]
the inner product (3.41) can be written as
\[
(x, y) = y^H x,
\]
and the filtered outputs can be written as
\[
y = Ah,
\]
where \( A \) is a matrix of the input data, \( f[t] \). The matrix \( A \) takes various forms, depending on the assumptions made on the data, as described in the following. Let
\[
d = \begin{bmatrix} d[i_1] \\ d[i_1 + 1] \\ \vdots \\ d[i_2] \end{bmatrix}
\]
be a vector of desired outputs. Then we want
\[
d \approx y = Ah.
\]
We can represent our approximation problem as
\[
d = Ah + e,
\]
where \( e \) is the difference between the output \( y \) and the desired output \( d \). We desire to find the filter coefficients \( h \) to minimize \( \|e\| \). By comparison with (3.16), observe that the solution is
\[
h = (A^H A)^{-1} A^H d.
\]

We now examine the form of the \( A \) matrix under various assumptions about the inputs. Assume that we have available to us, for the purpose of finding the coefficients, the data \( f[1], f[2], \ldots, f[N] \), with a total of \( N \) data points.
The "covariance" method. In this method, we use only data that is explicitly available, not making any assumptions about data outside this segment of observed data. The data matrix $A$ in this case is the $(N-m+1) \times m$ matrix

$$A = \begin{bmatrix} f[m] & f[m-1] & f[m-2] & \cdots & f[1] \\ f[m+1] & f[m] & f[m-1] & \cdots & f[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f[N] & f[N-1] & f[N-2] & \cdots & f[N-m+1] \end{bmatrix}.$$ 

Let $q[i]$ be the $m \times 1$ data vector corresponding to a (conjugated) row of $A$, as in (3.21); then

$$q[i] = \begin{bmatrix} \overline{f[i]} \\ \overline{f[i-1]} \\ \vdots \\ \overline{f[i-m+1]} \end{bmatrix}, \quad (3.43)$$

with the notation that $f[i] = 0$ where $i$ is outside the range 1 to $N$, and we can represent the data matrix as

$$A = \begin{bmatrix} q[m]^H \\ q[m+1]^H \\ \vdots \\ q[N]^H \end{bmatrix}.$$ 

The Grammian can be written as

$$R = A^H A = \sum_{i=0}^{N} q[i]q^H[i]. \quad (3.44)$$

The Grammian $R$ is a Hermitian matrix.

The "autocorrelation" method. In this case, we assume that data prior to $f[1]$ and after $f[N]$ are all zero, and fill up the data matrix $A$ with these assumed values. The output is taken from $i_1 = 1$ up through $i_2 = N + m - 1$. The data matrix is the $(N + m - 1) \times m$ matrix

$$A = \begin{bmatrix} f[1] & 0 & 0 & \cdots & 0 \\ f[2] & f[1] & 0 & \cdots & 0 \\ f[3] & f[2] & f[1] & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f[m] & f[m-1] & f[m-2] & \cdots & f[1] \\ f[m+1] & f[m] & f[m-2] & \cdots & f[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f[N] & f[N-1] & f[N-2] & \cdots & f[N-m+1] \\ 0 & f[N] & f[N-1] & \cdots & f[N-m+2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & f[N] \end{bmatrix}.$$ 

The terms "covariance method" and "autocorrelation method" do not produce, respectively, a covariance matrix and an autocorrelation matrix in the usual sense. Rather, these are the
terms for these methods commonly employed in the speech processing literature (see, e.g., [215]). Using the notation of (3.43), we can write the data matrix as

\[ A = \begin{bmatrix} q^H[1] \\ q^H[2] \\ \vdots \\ q^H[N + m - 1] \end{bmatrix}. \]

In a manner similar to (3.44), we can write

\[ R = A^H A = \sum_{i=1}^{N+m-1} q[i]q^H[i]. \]

This is a Toeplitz matrix.

**Pre-windowing method.** In this method we assume that \( f[t] = 0 \) for \( t < 1 \), and use data up to \( f[N] \), so that \( i_1 = 1 \) and \( i_2 = N \). Then the data matrix is the \( N \times q \) matrix


and

\[ R = \sum_{i=1}^{N} q[i]q^H[i]. \]

**Post-windowing method.** We begin with \( i_1 = m \), and assume that data after \( N \) are equal to zero. Then \( A \) is the \( N \times m \) matrix

\[ A = \begin{bmatrix} f[m] & f[m-1] & f[m-2] & \cdots & f[1] \\ f[m+1] & f[m] & f[m-2] & \cdots & f[2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f[N] & f[N-1] & f[N-2] & \cdots & f[N-m+1] \\ 0 & f[N] & F[N-1] & \cdots & f[N-m+2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & f[N] \end{bmatrix}, \]

and

\[ R = \sum_{i=m}^{m+N} q[i]q^H[i]. \]

**Example 3.9.1** Suppose we observe the data sequence

\( \{f[1], \ldots, f[5]\} = \{1, -2, 3, -4, 5\} \)

and want to filter these data with a filter of length \( m = 3 \). The data matrices corresponding to each interpretation, labeled respectively \( A_{\text{conv}}, A_{\text{pre}}, A_{\text{post}} \), and \( A_{\text{post}} \), with their corresponding Grammars.
Least-Squares Filtering

...are shown here:

\[ A_{\text{cov}} = \begin{bmatrix} 3 & -2 & 1 \\ -4 & 3 & -2 \\ 5 & -4 & 3 \end{bmatrix} \quad A_{\text{ac}} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -2 & 1 \end{bmatrix} \]

Observe that while all of the data matrices are Toeplitz (constant along the diagonals), the only Grammian which is Toeplitz is the one which arises from the autocovariance form of the data matrix.

MATLAB code to compute the least-squares filter coefficients is given in algorithm 3.1.

**Algorithm 3.1 Least-squares filter computation**

File: `lsfilt.m`

...Example 3.9.2 For the input data of the previous example, the following desired data are known:

\[ d = [2, -5, 11, -17, 23, -17, 15]^T. \]

We want to find a filter of length \( m = 3 \) that produces this data. Using the four different data sets in the example, with selections of \( d \) corresponding to the data used, we obtain from the MATLAB commands

\[ \text{hcv} = \text{lsfilt}(f,d(3:5),3,1) \]

\[ \text{hac} = \text{lsfilt}(f,d,3,2) \]

\[ \text{hpre} = \text{lsfilt}(f,d(1:5),3,3) \]

\[ \text{hpost} = \text{lsfilt}(f,d(3:7),3,4) \]

the filter coefficients

\[ h_{\text{cv}} = [1.5 \ -2 \ 2.5]^T \quad h_{\text{ac}} = [2 \ -1 \ 3]^T \]

\[ h_{\text{pre}} = [2 \ -1 \ 3]^T \quad h_{\text{post}} = [2 \ -1 \ 3]^T. \]

respectively.

**Example 3.9.3** An application of least-squares filtering is illustrated in figure 3.10 in a channel equalizer application. A sequence of bits \( \{b(t)\} \) is passed through a discrete-time channel with unknown...
impulse response, the output of which is corrupted by noise. To counteract the effect of the channel, the signal is passed through an equalizer, which in this case is an FIR filter whose coefficients have been determined using a least-squares criterion. In order to determine what the coefficients are, some set of known data—a *training sequence*—is used at the beginning of the transmission. This sequence is delayed and used as the desired signal $d[t]$. Using this training sequence, the filter coefficients $h[k]$ are computed by using (3.42), after which the coefficients are loaded into the equalizer filter.

This example is more a demonstration of a concept than a practical reality. While equalizers are common on modern modern technology, they are more commonly implemented using adaptive filters. Adaptive equalizers are examined in section 4.11.2 (RLS adaptive equalizer) and section 14.6 (LMS adaptive equalizer).

### 3.9.1 Least-squares prediction and AR spectrum estimation

Consider now the estimation problem in which we desire to predict $x[t]$ using a linear predictor based upon $x[t - 1], x[t - 2], \ldots, x[t - m]$. We then have

$$x[t] = -\sum_{i=1}^{m} a_i x[t - i] + f[t], \quad (3.46)$$

using $a_i = -h_i$, as the coefficients, where $f[t]$ is now used to denote the (forward) prediction error. The predictor of (3.46) is called a *forward predictor*. This is essentially the problem solved in the last section, in which the desired signal is the sample $d[t] = x[t]$, and the data used are the previous data samples. We can model the signal $x[t]$ as being the output of a signal with input $f[t]$, where the system function is

$$H(z) = \frac{X(z)}{F(z)} = \frac{1}{1 + \sum_{i=1}^{m} a_i z^{-i}} = \frac{1}{A(z)}.$$

If $f[t]$ is a random signal with power spectral density (PSD) $S_f(z)$, then the PSD of $x[t]$ is

$$S_x(z) = \frac{1}{(1 + \sum_{i=1}^{m} a_i z^{-i}) \left(1 + \sum_{i=1}^{m} \bar{a}_i z^i\right)} S_f(z) = \frac{1}{A(z)A(1/z)} S_f(z). \quad (3.47)$$

If $f[t]$ is assumed to be a white-noise sequence with variance $\sigma_f^2$, then the random process $x[t]$ has the PSD

$$S_x(z) = \frac{\sigma_f^2}{A(z)A(1/z)}.$$

Evaluating this on the unit circle $z = e^{j\omega}$, we obtain

$$S_x(\omega) \triangleq S_x(z)\big|_{z = e^{j\omega}} = \frac{\sigma_f^2}{|1 + \sum_{i=1}^{m} a_i e^{-j\omega i}|^2} = \frac{\sigma_f^2}{|A(\omega)|^2}. \quad (3.48)$$

Thus, by finding the coefficients of the linear predictor, we can determine an estimate of the spectrum, under the assumption that the signal is produced by the AR model (3.46).

We can obtain more data to put in our data matrix (and usually decrease the variance of the estimate) by using a *backward predictor* in addition to a forward predictor. In the
Least-Squares Filtering

The backward predictor, the \( m \) data points \( x[t], x[t-1], \ldots, x[t-m+1] \) are used to estimate \( x[t-m] \), by

\[
x[t-m] = - \sum_{i=1}^{m} a_i x[t-m+i] + b[t],
\]

where \( b[t] \) is the backward prediction error. As before, if we view \( x[t-m] \) as the output of a system driven by an input \( b[t] \), we obtain a system function

\[
H_b(z) = \frac{x(z)}{b(z)} = \frac{1}{z^{-m} \left( 1 + \sum_{i=1}^{m} a_i z^{-i} \right)} = \frac{1}{z^{-m} A(1/z)}.
\]

If \( b[t] \) is a white-noise sequence with variance \( \sigma_{b}^2 = \sigma_{j}^2 \), then the PSD of the signal \( x[t-m] \) is

\[
S_x(z) = \sigma_{b}^2 \frac{1}{A(1/z) A(z)},
\]

the same as in (3.47). Since both the forward predictor and the backward predictor use the same predictor coefficients (just conjugated and in a different order), we can use the backward predictor information to improve our estimate of the coefficients. If we have measured data \( x[1], x[2], \ldots, x[N] \), we write our prediction equations as follows (using the covariance method employing only measured data):

\[
\begin{bmatrix}
x[m] & x[m-1] & \cdots & x[1] \\
x[m+1] & x[m] & \cdots & x[2] \\
\vdots & \vdots & \ddots & \vdots \\
x[N-1] & x[N-2] & \cdots & x[N-m] \\
\vdots & \vdots & \ddots & \vdots \\
x[N-m+1] & x[N-m+2] & \cdots & x[N]
\end{bmatrix}
\begin{bmatrix}
-a_1 \\
-a_2 \\
\vdots \\
-a_m
\end{bmatrix}
\]

Let us write this as

\[
x = Ah + e,
\]

where \( x \) and \( e \) are now \( 2(N-m) \times 1 \) and \( A \) is \( 2(N-m) \times n \). In the data matrix, the first \( N-m \) rows correspond to the forward predictor and the second \( N-m \) rows correspond to the backward predictor. Our optimization criterion is to minimize

\[
\sum_{i=m+1}^{N} |f[i]|^2 + |b[i]|^2.
\]

As before, a least-squares solution is straightforward. This technique of spectrum estimation is known as the forward–backward linear prediction (FBLP) technique, or the modified
covariance technique. An estimate of the variance is
\[ \hat{\sigma}_e^2 = \hat{\sigma}_n^2 = \| e_{\text{min}} \|^2. \]
A MATLAB function that computes the AR parameters using the modified covariance technique is shown in algorithm 3.2.

Algorithm 3.2 Forward–backward linear predictor estimate
File: fblp.m

3.10 Minimum mean-square estimation

In the least-squares estimation of the preceding sections, we have not employed, nor assumed the existence of, any probabilistic model. The optimization criterion has been to minimize the sum of squared error. In this section, we change our viewpoint somewhat by introducing a probabilistic model for the data.

Let \( P_1, P_2, \ldots, P_m \) be zero mean random variables. We desire to find coefficients \( \{c_i\} \) to estimate the random variable \( X \), using
\[ X = c_1 P_1 + c_2 P_2 + \cdots + c_m P_m + e \]
in such a way that the norm of the squared error is minimized. Using the inner product
\[ \langle X, Y \rangle = E[XY], \]
the minimum mean-square estimate of \( e \) is given by
\[ Re = p, \]
where
\[ R = \begin{bmatrix} E[P_1 P_1] & E[P_2 P_1] & \cdots & E[P_m P_1] \\ E[P_1 P_2] & E[P_2 P_2] & \cdots & E[P_m P_2] \\ \vdots & \vdots & \ddots & \vdots \\ E[P_1 P_m] & E[P_2 P_m] & \cdots & E[P_m P_m] \end{bmatrix} \quad \text{and} \quad p = \begin{bmatrix} E[X P_1] \\ E[X P_2] \\ \vdots \\ E[X P_m] \end{bmatrix}. \]
The minimum mean-squared error in this case is given using (3.29) as
\[ \| e \|_{\text{min}}^2 = \sigma_e^2 - p^H R^{-1} p \]
\[ = \sigma_e^2 - p^H c. \]

Example 3.10.1 Suppose that
\[ Z = [X_1, X_2, X_3]^T \]
is a real Gaussian random vector with mean zero and covariance
\[ R_Z = \text{cov}(Z) = E[ZZ^T] = \begin{bmatrix} 1 & 2 & .1 \\ 2 & 2 & .3 \\ .1 & .3 & 4 \end{bmatrix}. \]

Given measurements of \( X_1 \) and \( X_2 \), we wish to estimate \( X_3 \) using a linear estimator,
\[ \hat{X}_3 = c_1 X_1 + c_2 X_2. \]
The necessary correlation values in (3.51) can be obtained from the covariance \( R_{zz} \),

\[
R = \begin{bmatrix}
E[X_1 X_1] & E[X_1 X_2] \\
E[X_2 X_1] & E[X_2 X_2]
\end{bmatrix} = \begin{bmatrix} 1 & 0.2 \\
0.2 & 2 \end{bmatrix}
\quad \text{and} \quad
p = \begin{bmatrix} E[X_2 X_1] \\
E[X_2 X_2]
\end{bmatrix} = \begin{bmatrix} 1.1 \\
3 \end{bmatrix},
\]

from which the optimal coefficients are

\[
\mathbf{c} = \begin{bmatrix} 0.0714 \\
0.1429 \end{bmatrix}.
\]

The minimum mean-squared error is

\[
\|\mathbf{e}\|_{\text{min}}^2 = 4 - \mathbf{p}^T R^{-1} \mathbf{p} = 3.95.
\]

### 3.11 Minimum mean-squared error (MMSE) filtering

A minimum mean-square (MMS) filter is called a Wiener filter. It is mathematically similar to a least-squares filter, except that the expectation operator is used as the inner product. Given a sequence of data \( \{ f[t] \} \), we desire to design a filter in such a way that we get as close as possible to some desired sequence \( d[t] \). In the interest of generality, we assume the possibility of an IIR filter,

\[
y[t] = \sum_{l=0}^{\infty} h[l] f[t - l].
\] (3.53)

In adopting a statistical model, we assume that the signals involved are wide-sense stationary so that, for example,

\[
E[x[t]] = E[x[t - l]] \quad \text{for all } l
\]

and

\[
E[x[t] \bar{x}[t - l]]
\]

depends only upon the time difference \( l \) and not upon the sample instant \( t \).

Using

\[
e[t] = d[t] - y[t]
\] (3.54)

as the estimator error, by the orthogonality principle, the squared norm of error, which in this case is termed the mean-squared error,

\[
\|e[t]\|^2 = E[e[t] \bar{e}[t]],
\]

is minimized when the error is orthogonal to the data. That is, the optimal estimator satisfies

\[
\left< d[t] - \sum_{l=0}^{\infty} h[l] f[t - l], f[t - i] \right> = 0
\]

for \( i = 0, 1, 2, \ldots \); or,

\[
(d[t], f[t - i]) = \sum_{l=0}^{\infty} h[l] (f[t - l], f[t - i]). \tag{3.55}
\]

Using the inner product (3.50), we obtain

\[
\sum_{l=0}^{\infty} h[l] E[f[t - l] \bar{f}[t - i]] = E[\bar{f}[t - i] d[t]]. \tag{3.56}
\]
Equation (3.56) is an infinite set of normal equations. For this case in which the inner product is defined using the expectation, the normal equations are referred to as the Wiener-Hopf equations. We can place the normal equations into a more standard form by expressing the Grammian in the form of an autocorrelation matrix. Define

\[ r(i - l) = E[f(t - l)f(t - i)] = \langle f(t - l), f(t - i) \rangle \]

and

\[ p(i) = E[f(t - i)d[t]] = \langle d[t], f(t - i) \rangle, \]

and observe that \( r(-k) = r(k) \). Then (3.56) can be written as

\[ \sum_{i=0}^{\infty} h[l]r(i - l) = p(i), \quad i = 0, 1, \ldots \] (3.57)

Solution of this problem for an IIR filter is reexamined in section 3.13.

For now, we focus on the solution when \( \{h\} \) is an FIR filter with \( m \) coefficients. Then the filter output can be written as

\[ y[t] = f[t]^H h, \]

where

\[ f[t] = [f[t], f[t-1], \ldots, f[t-m+1]] \quad \text{(note the conjugates in this definition)} \]

and

\[ h = [h[0], h[1], \ldots, h[m-1]]^T. \]

Under the assumption of an FIR filter, (3.57) can be written as

\[ \sum_{i=0}^{m-1} h[l]r(i - l) = p(i), \quad i = 0, 1, \ldots \] (3.59)

which we can express in matrix form with \( R_{ll} = r(i - l) \) as

\[ Rh = p, \] (3.60)

where

\[ R = \begin{bmatrix}
  r(0) & \bar{r}(1) & \bar{r}(2) & \cdots & \bar{r}(m-1) \\
r(1) & r(0) & \bar{r}(1) & \cdots & \bar{r}(m-2) \\
r(2) & r(1) & r(0) & \cdots & \bar{r}(m-3) \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
r(m-1) & r(m-2) & r(m-3) & \cdots & r(0)
\end{bmatrix} = E[f[t]f^H[t]] \] (3.61)

and

\[ p = \begin{bmatrix}
p(0) \\
p(1) \\
p(2) \\
\vdots \\
p(m-1)
\end{bmatrix} = E[f[t]d[t]]. \] (3.62)

The optimal weights from (3.60) are \( h = R^{-1}p \).
The matrix $R$ is the Grammian matrix and has the special form of a Toeplitz matrix: the diagonals are equal to each other. Because of this special form, fast algorithms exist for inverting the matrix and solving for the optimum filter coefficients. Toeplitz matrices are discussed further in section 8.3. (We have already seen one example of the solution of Toeplitz equations with a special right-hand side, in Massey’s algorithm in section 1.9.)

The minimum mean-squared error can be determined using (3.52) to be

$$\|e\|^2_{\text{min}} = E[e^2]_{\text{min}} = \|d\|^2 - \|y\|^2.$$  

Using the notation $\|e\|^2 = \sigma_e^2$ and $\|d\|^2 = \sigma_d^2$, and noting that

$$\|y[t]\|^2 = E[y[t]y^*[t]] = E[h^H[r]x[r]x^H[r]]h = h^H Rh = p^H h,$$

we obtain

$$\left(\sigma_e^2\right)_{\text{min}} = \sigma_d^2 - p^H h.$$  

Example 3.11.1 In this example we explore a simple equalizer. Suppose we have a channel with transfer function

$$H_c(z) = \frac{1}{1 - .6z^{-1}}.$$  

Passing into the channel is a desired signal $d[t]$. The output of the channel is $u[t]$, so that we have

$$u[t] = 0.6u[t - 1] = d[t].$$  

However, we observe only a noise-corrupted version of the channel output,

$$f[t] = u[t] + n[t].$$

where $n[t]$ is a zero-mean white-noise sequence with variance $\sigma_n^2 = 0.16$, which is uncorrelated with $v[t]$. Suppose, furthermore, that we have a statistical model for the desired signal, in which we know that $d[t]$ is a first-order AR signal generated by

$$d[t] = -0.5d[t - 1] + v[t],$$

where $v[t]$ is a zero-mean white-noise signal with variance $\sigma_v^2 = 0.1$. Based on this information, we desire to find an optimal Wiener filter to estimate $d[t]$, using the observed sequence $f[t]$. The diagram is shown in figure 3.11. The cascade of the AR process and the channel gives the combined transfer function from $v[t]$ to $u[t]$ as

$$H(z) = \frac{1}{(1 + .5z^{-1})(1 - .6z^{-1})} = \frac{1}{1 - .1z^{-1} -.3z^{-2}} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$  

Figure 3.11: An equalizer problem
so that
\[ u[t] - .1u[t - 1] - .3u[t - 2] = v[t]. \]

In this example, since the channel output is an AR(2) process, the equalizer used is a two-tap FIR filter.

We need the matrix \( R \), containing autocorrelations of the signal \( f[t] \), and the cross-correlation vector \( p \). Since \( f[t] = u[t] + n[t] \), and since \( v[t] \) and \( n[t] \) are uncorrelated, we have
\[ R = R_{ff} = R_{uu} + R_{nn}. \]

where \( R_{uu} \) is the autocorrelation matrix for the signal \( u[t] \) and \( R_{nn} \) is the autocorrelation matrix for the signal \( n[t] \). Since \( n[t] \) is a white-noise sequence, \( R_{nn} = \sigma_n^2 I \), where \( I \) is the \( 2 \times 2 \) identity matrix. To find
\[ R_{uu} = \begin{bmatrix} r_u(0) & r_u(1) \\ r_u(1) & r_u(0) \end{bmatrix}, \]
we use the results from section 1.4.2. Specifically, from (1.79) and (1.80) we find
\[ \sigma_u^2 = r_u(0) = \frac{1 + a_2}{1 - a_2} \cdot \frac{\sigma_n^2}{(1 + a_2)^2 - a_1^2} = 0.1122 \]
\[ r_u(1) = \frac{-a_1 - \sigma_n^2}{1 + a_2} = 0.0160. \]

Thus,
\[ R = \begin{bmatrix} .16 & 0 \\ 0 & .16 \end{bmatrix} + \begin{bmatrix} .1122 & .0160 \\ .0160 & .1122 \end{bmatrix} = \begin{bmatrix} .2722 & .0160 \\ .0160 & .2722 \end{bmatrix}. \]

For the cross-correlation vector,
\[ p = E \begin{bmatrix} \bar{f}[t]d[t] \\ \bar{f}[t - 1]d[t] \end{bmatrix} = E \begin{bmatrix} (\bar{u}[t] + \bar{n}[t])d[t] \\ (\bar{u}[t - 1] + \bar{n}[t - 1])d[t] \end{bmatrix} = E \begin{bmatrix} \bar{u}[t]d[t] \\ \bar{u}[t - 1]d[t] \end{bmatrix}, \]
since \( d[t] \) is uncorrelated with \( n[t-n] \). Multiplying (3.64) through by \( \bar{u}[t-k] \) and taking expectations, we obtain
\[ p(k) = E[\bar{u}[t-k]d[t]] = r_u(k) - 0.6r_u(k-1), \]
from which we can determine
\[ p = \begin{bmatrix} 0.1026 \\ -0.0513 \end{bmatrix}. \]

The optimal filter coefficients are
\[ h = R^{-1}p = \begin{bmatrix} 0.3893 \\ -0.2113 \end{bmatrix}. \]

To compute the minimum mean-squared error from (3.63) we need \( \sigma_d^2 \). This is found using (1.75) as
\[ \sigma_d^2 = \frac{\sigma_n^2}{1 - S^2}. \]

Then
\[ \sigma_d^2 = 0.0826. \]

The error surface is obtained by plotting (see (3.14))
\[ J(h) = \sigma_d^2 - 2p^T \begin{bmatrix} h[0] \\ h[1] \end{bmatrix} + |h[0], h[1]|^2 R \begin{bmatrix} h[0] \\ h[1] \end{bmatrix}. \]
3.12 Comparison of Least-Squares and Minimum Mean-Squares

![Contour plot of an error surface](image)

Figure 3.12: Contour plot of an error surface as a function of \( [h[0], h[1]] \). Figure 3.12 shows a contour plot of the error surface. Algorithm 3.3 is MATLAB code demonstrating these computations.

---

Algorithm 3.3 Two-tap channel equalizer  
File: \texttt{wftest.m}

---

Another example of MMSE filter design is given in conjunction with the RLS filter in 4.1.1.2.

3.12 Comparison of least-squares and minimum mean-squares

It is interesting to contrast the method of least-squares and the method of minimum mean squares, both of which are widely used in signal processing. For the method of least-squares, we make the following observations:

1. Only the sequence of data observed at the time of the estimate is used in forming the estimate.
2. Depending upon assumptions made about the data before and after the observation interval, the Grammian matrix may not be Toeplitz.
3. No statistical model is necessarily assumed.

For the method of minimum mean-squares, we make the following observations:

1. A statistical model for the correlations and cross-correlations is necessary. This must be obtained either from explicit knowledge of the channel and signal (as was seen in example 3.11.1), or on the basis of the multivariable distribution of the data (as was seen in example 3.10.1). In the absence of such knowledge, it is common to estimate the necessary autocorrelation and cross-correlation values. An example of an estimate of the autocorrelation \( r(n) = E[x(k)x(k-n)] \) using the data \( \{x(1), x(2), \ldots, x(N)\} \)
This is actually a biased estimate of \( r(n) \) (see exercise 3.12-211, but it has been found (see, e.g., [38]) to produce a lower variance when the lag \( n \) is close to \( N \).

In order for (3.65) to be a reasonable estimate of \( r(z^n) \), the random process \( x(k) \) must be ergodic, so that the time average asymptotically approaches the ensemble average. This assumption of ergodicity is usually made tacitly, but it is vital.

When the data sequence used to compute the estimate of the correlations’ parameters is the same as the data sequence for which the filter coefficients are computed, the minimum mean-squared error technique is essentially the same as the least-squares technique.

2. Commonly, the coefficients of the MMS technique are computed using a separate set of data whose statistics are assumed to be the same as those of the real data set of interest. This set of data is used as a training set to find the autocorrelation functions and the filter coefficients. Provided that the training data does have the same (or very similar) statistics as the data set of interest, this works well. However, if the training data is significantly different from the data set of interest, finding the optimum filter coefficients can actually lead to poor performance, because one has found the best solution to the wrong problem.

3. We also note that the (true) Grammian matrix \( R \) used in prediction and optimal FIR filtering problems is always a Toeplitz matrix, and hence fast algorithms apply to finding the coefficients.

In section 4.11.1 we examine how the coefficients of the LS filter can be updated adaptively, so that the coefficients are modified as new data arrives. In section 14.6, we develop an algorithm so that the coefficients of the MMS filter can be updated adaptively by approximating the expectation. These two concepts form the heart of adaptive filtering theory.

### 3.13 Frequency-domain optimal filtering

We have seen several examples of FIR minimum mean-squared filters, in which the equations obtained involve a finite number of unknowns. In this section, we take a different viewpoint, and develop optimal filtering techniques for scalar signals in the frequency domain. This allows us to extend the minimum mean-squared error filters of section 3.11 to IIR filters. Following a brief review of stochastic processes and their processing by linear systems, we present the notion of two-sided Laplace transforms, and some decompositions of these that are critical to the solution of the Wiener filter equations. This is followed by the development of the continuous-time Wiener filter. Finally, we present analogous results for discrete-time Wiener filters.

#### 3.13.1 Brief review of stochastic processes and Laplace transforms

To expedite our development of frequency-domain filtering, it will be helpful to review briefly some fundamental results from stochastic processes associated with linear systems (see also appendix D).
Power spectral density functions and filtering stochastic processes

Let \( \{x_t, -\infty < t < \infty\} \) and \( \{y_t, -\infty < t < \infty\} \) be two wide-sense stationary, zero-mean, scalar stochastic processes. Throughout this development, we will assume that all processes are real. The auto- and cross-correlation functions are

\[
R_x(t) = E[x_{t+\tau}x_\tau] \quad R_y(t) = E[y_{t+\tau}y_\tau] \\
R_{xy}(t) = E[x_{t+\tau}y_\tau] \quad R_{yx}(t) = E[y_{t+\tau}x_\tau].
\]

The bilateral Laplace transforms of these functions are denoted by

\[
S_x(s) = \int_{-\infty}^{\infty} R_x(t)e^{-st}dt \quad S_y(s) = \int_{-\infty}^{\infty} R_y(t)e^{-st}dt \\
S_{xy}(s) = \int_{-\infty}^{\infty} R_{xy}(t)e^{-st}dt \quad S_{yx}(s) = \int_{-\infty}^{\infty} R_{yx}(t)e^{-st}dt,
\]

where \( s = \sigma + j\omega \) is a complex variable. These bilateral Laplace transforms exist whenever \( s \) is in the region of convergence. For all of our applications, the region of convergence will include the imaginary axis, and we may obtain the Fourier transform of these functions by restricting \( s \) to the imaginary axis, that is, setting \( s = j\omega \). The resulting function, \( S_x(j\omega), \) etc., is the usual power spectral density function. By an abuse of notation, we will usually drop the explicit inclusion of the imaginary unit in the argument, and simply refer to the power spectral density as \( S_x(\omega), \) and so on.

We observe that, since the autocovariance is real and even, its bilateral Laplace transform is even; that is,

\[
S_x(s) = S_x(-s).
\]

Furthermore, when \( s = j\omega \), the power spectral density has the property

\[
S_x(-\omega) = S_x^*(\omega).
\]

Filtering of stochastic processes

Let \( h(t) \) be the impulse response function of a time-invariant linear system Laplace transform \( H(s) \). We will be concerned (as usual) mainly with causal systems, in which \( h(t) = 0 \) for \( t < 0 \).

Let \( y_t \) be the output of a system with impulse response driven by the wide-sense stationary stochastic process \( \{x_t, -\infty < t < \infty\} \). The output of this system, denoted \( \{y_t, -\infty < t < \infty\}, \) is also a wide-sense stationary stochastic processes. The correlation functions \( R_{xy}(\tau), R_{yx}(\tau) \) and \( R_y(\tau) \) are given by

\[
R_{xy}(\tau) = R_x(\tau) * h(-\tau) \quad R_{yx}(\tau) = h(\tau) * R_x(\tau) \\
R_y(\tau) = h(\tau) * R_x(\tau) * h(-\tau).
\]

The equivalent relationships in the spectral domain are

\[
S_{xy}(s) = S_x(s)H(-s) \quad S_{yx}(s) = H(s)S_y(s) \\
S_y(s) = H(s)S_x(s)H(-s).
\]

Lumped systems and processes

A linear system is said to be lumped if it has a rational transfer function; that is, its transfer function is a ratio of polynomials in \( s \). Thus, if \( G(s) \) is a rational transfer function, then it
is of the form
\[ G(s) = \prod_{i=1}^{m} \frac{(s - z_i)}{\prod_{i=1}^{n} (s - p_i)}, \]
where \(z_i\) and \(p_i\) are the roots of the numerator (the zeros) and the denominator (the poles), respectively. We require that \(n \geq m\).

A stochastic process is said to be \textit{lumped} if its power spectral density is a rational function. Let \(\{y_t\}\) be a lumped stochastic process. Its spectral density function, \(S_y(\omega)\), is even and nonnegative. We will sometimes refer to \(S_y(s)\) as the power spectral density, although the nonnegativeness only holds for \(s = j\omega\). The evenness and nonnegativeness of \(S_y(\omega)\), however, means that the poles and zeros of \(S_y(s)\) have a particular quadrantal symmetry:

- The poles and zeros are symmetric about the real axis of the complex plane, because \(S_y(\omega)\) is real.
- The poles and zeros are symmetric about the imaginary axis of the complex plane, because \(S_y(\omega)\) is even.
- The imaginary axis of the complex plane has zeros of even multiplicity, because \(S_y(\omega)\) is nonnegative.
- There are no poles on the imaginary axis, because the inverse Fourier transform cannot be a covariance function.

Figure 3.13 illustrates the pole-zero structure of a rational power spectral density function.

The region of convergence for a stable inverse of \(S_y(s)\) is a strip in the complex plane containing the \(j\omega\) axis. The inverse Laplace transform of \(S_y(s)\) is of the form
\[ R_y(t) = \sum_{i=1}^{n} c_i e^{-d_i|t|}, \]
which is a sum of damped exponentials for positive as well as for negative \(t\). If any coefficient \(d_i\) is complex, then its complex conjugate, \(d_i^*\), must also be one of the coefficients. Purely imaginary \(d_i\) are excluded since \(R_y\) must be a correlation function. The coefficients \(c_i\) must be real.

---

![Figure 3.13: Pole-zero plot of rational \(S_y(s)\) (\(\times = \text{poles}, \circ = \text{zeros}\).](image-url)
3.13.2 Two-sided Laplace transforms and their decompositions

The one-sided Laplace transform should be familiar to students of signal processing. Less familiar, but applicable to our current study, is the two-sided, or bilateral, Laplace transform, defined as

\[ F(s) = \mathcal{L}_{\text{two-sided}}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-st} dt. \]

Of course, for a causal function \( f(t) \), the bilateral transform is equivalent to the one-sided transform.

Like the two-sided \( Z \)-transform (which should be somewhat more familiar), different inverses of a given function \( F(s) \) can be obtained depending upon the region of convergence that is selected. We make the following summarizing observations, where \( f(t) \) and \( F(s) \) are Laplace transform pairs.

1. If the region of convergence includes the \( j\omega \) axis, then the inverse transform \( f(t) \) is stable.
2. If the region of convergence is to the right of all poles of \( F(s) \), then the inverse \( f(t) \) is causal. That is, the region of convergence is a region of the form \( \Re(s) > \Re(p) \), for all poles \( p \) of \( F(s) \). Conversely, if \( f(t) \) is a causal, stable function, then there are no poles in the RHP.
3. If the region of convergence is to the left of all poles of \( F(s) \), then the inverse \( f(t) \) is anticausal. Conversely, if \( f(t) \) is an anticausal, stable, function, then there are no poles in the RHP.
4. If the region of convergence is neither to the right nor to the left of all of the poles, the inverse transform is two-sided.

Some simple examples will demonstrate these concepts.

**Example 3.13.1**

1. The transform \( F(s) = \frac{1}{s + \alpha} \), \( \alpha > 0 \), has its poles in the LHP, and the region of convergence to the right of the poles contains the \( j\omega \) axis, indicating that \( f(t) \) is stable. In fact, the inverse (one-sided) Laplace transform is \( f(t) = e^{-\alpha t}u(t) \), a stable, causal function.
2. Let \( f(t) = -e^{\alpha t}u(-t) \). Then the two-sided transform of \( F(s) \) is

\[ F(s) = \frac{1}{s - \alpha}, \]

with region of convergence \( \Re(s) < \Re(\alpha) \). If \( \Re(\alpha) > 0 \), then \( f(t) \) is stable, and \( F(s) \) has no poles in the LHP.
3. Let

\[ F(s) = \frac{2\alpha}{\alpha^2 - s^2} = \frac{1}{s + \alpha} - \frac{1}{s - \alpha}. \]

The region of convergence of \( F(s) \) containing the \( j\omega \) axis has poles both to the right and to the left, hence the inverse using this region of convergence is stable, but not causal. In fact, it can be verified that the inverse corresponding to this region of convergence is \( f(t) = e^{-\alpha t} \).
4. Suppose

\[ F(s) = \frac{1}{\alpha + s}e^{i\lambda}, \quad \alpha > 0, \ \lambda > 0. \]

This is not the transfer function of a lumped system. Let the region of convergence be \( \Re(s) > -\alpha \), which includes the \( j\omega \) axis and hence is stable. The stable inverse transform
of $F(s)$ is

$$f(t) = e^{-u(t+\lambda)}u(t+\lambda) = e^{-u\lambda}e^{u(t+\lambda)},$$

which is not causal. The causal portion of this function is

$$f(t)u(t) = e^{-u\lambda}e^{u(t)}.$$

This causal function has Laplace transform

$$f(t)u(t) \leftrightarrow \frac{e^{-u\lambda}}{s+\alpha}.$$  

### Canonical factorizations

Let $\{z_1, \ldots, z_m\}$ be the LHP zeros of a lumped system $F(s)$, and let $\{p_1, \ldots, p_m\}$ be the LHP poles of $F(s)$ for some Laplace transform function $F(s)$. We may then express $F(s)$ as

$$F(s) = F^+(s)F^-(s),$$  

where

$$F^+(s) = \prod_{i=1}^{m}(s-z_i) \prod_{i=1}^{m}(s-p_i).$$

Then, since the $j\omega$ axis is to the right of all the poles of $F^+(s)$, the stable inverse Laplace transform of $F^+(s)$ is causal.

For a power spectral density $S_y(s)$ with LHP zeros and poles $\{z_1, \ldots, z_m\}$ and $\{p_1, \ldots, p_m\}$, respectively, the zeros and poles occur in mirror images, so that $S_y(s)$ has the canonical factorization

$$S_y(s) = S^+_y(s)S^-_y(s),$$

where

$$S^+_y(s) = \frac{\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{m}(s-p_i)} \quad S^-_y(s) = S^+_y(-s).$$

$S^+_y(s)$ is often called the *canonical spectral factor* of $S_y(s)$. Since $S^+_y(s)$ has all of its poles and zeros in the LHP, its reciprocal, $W(s) = \frac{1}{S^+_y(s)}$, also has its poles and zeros in the LHP. Functions that have both poles and zeros in the left-half plane are said to be of *minimum phase*, and such functions may be viewed as transfer functions of causal systems that possess the property that their inverse is also causal. Thus, we may view $\{y_t\}$ as the output of a linear system with transfer function $S^+_y(s)$, driven by a white noise, $\{v_t\}$, as illustrated in figure 3.14. We take the spectral density of the white noise process $\{v_t\}$ to be unity ($S_v(s) = 1$), so the spectral density of $\{y_t\}$ is

$$S_y(s) = S^+_y(s)S_v(s)S^+_y(-s) = S^+_y(s)S^-_y(s),$$

which agrees with the canonical factorization (3.66).

Since $S^+_y(s)$ is causally invertible, we may also view $\{v_t\}$ as the output of a causal and causally invertible linear system with transfer function $\frac{1}{S^+_y(s)}$, driven by $\{y_t\}$, as illustrated in figure 3.15.

The relationship between $\{y_t\}$ and $\{v_t\}$ is very important. Since the transfer function $S^+_y(s)$ is causal, we can obtain $v_t$ from $\{y_t, \alpha < t\}$; and since the transfer function $\frac{1}{S^+_y(s)}$ is causal, we can obtain $y_t$ from $\{v_t, \alpha < t\}$. Thus, $\{y_t, \alpha < t\}$ and $\{v_t, \alpha < t\}$ contain exactly the same information—nothing is lost or destroyed as a result of the filtering operations.
We will say that two signals that enjoy this relationship are \textit{informationally equivalent}. The main difference between the two processes is that, while \( y_t \) may be dependent on \( \{ y_{\alpha} \mid \alpha < t \} \), \( v_t \) is \textit{not} dependent on \( \{ y_{\alpha} \mid \alpha < t \} \). In other words, \( \{ y_t \} \) is a correlated process, and \( \{ v_t \} \) is an uncorrelated process. The action of filtering by \( \frac{1}{S_y(s)} \) is to decorrelate \( \{ y_t \} \) by, essentially, removing all redundant information (that is, the part \( y_t \) that can be obtained as a function of \( y_{\alpha} \) for \( \alpha < t \)) from \( y_t \) at each time \( t \). The process \( \{ v_t \} \) is called the \textit{innovations} process, and contains only new information about \( y_t \) that cannot be predicted from past values. Since \( v(t) \) is a white-noise signal, we say that the filter \( W(s) = \frac{1}{S_y(s)} \) is a whitening filter. The process \( \{ v_t \} \) is a very special white-noise process, since it represents exactly the same information as is contained in the original signal.

\textbf{Additive decompositions}

Let \( f(t) \) be any function whose bilateral Laplace transform, \( \mathcal{L}\{ f(t) \} \), exists in a region containing the \( j\omega \) axis. The auto- and cross-correlation functions associated with lumped processes and transfer functions of lumped systems all satisfy this constraint. We may decompose \( f(t) \) into its left- and right-hand components

\[
f(t) = f(t)u(t) + f(t)u(-t) - f(0),
\]

where

\[
u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}
\]

is the unit step function. The bilateral Laplace transform of \( f(t) \), denoted

\[
F(s) = \mathcal{L}\{ f(t) \} = \int_{-\infty}^{\infty} f(t)e^{-st} dt.
\]

may be decomposed into

\[
F(s) = \{ F(s) \}_+ + \{ F(s) \}_-,
\]

where

\[
\{ F(s) \}_+ = \mathcal{L}\{ f(t)u(t) \} = \int_{0-}^{\infty} f(t)e^{-st} dt
\]

\textsuperscript{1}This notion of information is not the same as either Shannon information or Fisher information.
is the Laplace transform of the causal part of $f(t)$, and

$$\{F(s)\}_+ = \mathcal{L}\{f(t) u(-t)\} = \int_{-\infty}^{0-} f(t) e^{-st} dt$$

is the Laplace transform of the anticausal part of $f(t)$. Here $0-$ signifies taking a left-hand limit (this is necessary to account for impulsive autocorrelation functions).

Since $f(t) u(t)$ is a right-sided function, the region of convergence for its Laplace transform includes the RHP; that is, the transform $\{F(s)\}_+$ has no poles in the RHP. Similarly, since $f(t) u(-t)$ is an anticausal (left-sided) function, the region of convergence for its Laplace transform includes the LHP. Thus, for rational $F(s)$,

$$\{F(s)\}_+ = \mathcal{L}\{\text{impulsive functions}\} + \sum_{\text{LHP poles}} \{\text{partial fraction expansion of } F(s)\},$$

$$\{F(s)\}_- = \sum_{\text{RHP poles}} \{\text{partial fraction expansion of } F(s)\}.$$

Despite the confusing notation, the canonical factorization and the additive decomposition should not be confused: the canonical factorization is a multiplicative decomposition. For a function $F(s)$, we have the canonical factorization

$$F(s) = F^+(s) F^-(s),$$

and the additive decomposition

$$F(s) = \{F(s)\}_+ + \{F(s)\}_-,$$

and it is not the case that $F^+(s) = \{F(s)\}_+)$. What is true is that they both have poles only in the RHP, and causal inverse transforms. (Note that the canonical factorization places the $+$ and $-$ in the exponent, while the additive decomposition placed the $+$ and $-$ in the subscript.)

**Example 3.13.2** Let

$$f(t) = e^{-\alpha |t|},$$

where $\alpha > 0$. Then

$$F(s) = \frac{2\alpha}{\alpha^2 - s^2} = \frac{1}{s + \alpha} - \frac{1}{s - \alpha}.$$  

This has the canonical factorization

$$F(s) = \frac{\sqrt{2\alpha}}{s + \alpha} \frac{\sqrt{2\alpha}}{s - \alpha} = F^+(s) F^-(s).$$

The causal part of $f(t)$ is $f(t) u(t)$, which has Laplace transform

$$\{F(s)\}_+ = \frac{1}{s + \alpha}, \quad \text{Re}(s) > -\alpha,$$

leading to the additive decomposition

$$F(s) = \frac{1}{s + \alpha} - \frac{1}{s - \alpha} = \{F(s)\}_+ + \{F(s)\}_-.$$

**Example 3.13.3** Let $S^-(s)$ be the canonical factor with its poles and zeros in the RHP, of the form

$$S^-(s) = \frac{s^2 - 3s + 2}{s^2 - 7s + 12} = 1 + \frac{4s - 10}{s^2 - 7s + 2}.$$
We desire to find \( \{S_y^-(s)\}_+ \), the transform due to the causal part of the inverse Laplace transform of \( S_y^-(s) \). We first find

\[
\mathcal{L}^{-1}(S_y^-(s)) = \delta(t) - 6e^tu(-t) + 2e^u(-t),
\]
so that the "causal" part is simply \( \delta(t) \). Taking the transform of the causal part, we thus have

\[
\{S_y^-(s)\}_+ = 1.
\]

As this last example shows, \( \{S_y^-(s)\}_+ \) may have a nonzero part. In fact, if \( S_y^-(s) \) is rational, \( S_y^-(s) = \frac{\beta(s)}{\alpha(s)} \) for polynomials \( \beta(s) \) and \( \alpha(s) \) of equal degree, then with a little thought, we realize that

\[
\{S_y^-(s)\}_+ = 1. \quad (3.67)
\]

### 3.13.3 The Wiener–Hopf equation

Let \( x_t \) and \( y_t \) be zero-mean, stationary stochastic processes, and let \( \mathcal{Y}_t = \{y_\alpha, \alpha \leq t\} \) be observed. Suppose we wish to estimate \( x_{t+\lambda} \), given \( \mathcal{Y}_t \). If \( \lambda > 0 \), we wish to predict future values of \( x_t \) given past and present values of \( y_t \). This is called the prediction problem. If \( \lambda = 0 \), we wish to estimate \( x_t \) in real time; this is called the filtering problem. If \( \lambda < 0 \), we wish to estimate the signal \( \lambda \) time units in the past; this is called the smoothing problem. The prediction and filtering problems are causal, and can be implemented in real time, while the smoothing problem is noncausal, and cannot be implemented in real time.

We first formulate the integral

\[
\hat{x}_{t+\lambda} = \int_{-\infty}^{t} h(t, s) y_s \, ds, \quad (3.68)
\]

where \( h(t, s) \) is to be chosen such that

\[
E[(x_{t+\lambda} - \hat{x}_{t+\lambda})^2]
\]

is minimized and \( h(t, s) \) is causal (that is, \( h(t, s) = 0 \) for \( t < s \)). The integral in (3.68) is to be taken in the mean-square sense. We address this problem by appealing to the orthogonality principle, as we have done so many times, whereby we require

\[
E[(x_{t+\lambda} - \hat{x}_{t+\lambda})y_\sigma] = 0, \quad \forall \sigma \leq t; \quad (3.70)
\]

that is, the estimation error must be perpendicular to all data used to generate the estimate. This condition implies that

\[
E x_{t+\lambda} y_\sigma = E \hat{x}_{t+\lambda} y_\sigma = \int_{-\infty}^{t} h(t, \tau) E y_\tau y_\sigma \, d\tau, \quad \forall \sigma \leq t.
\]

or

\[
R_{xy}(t + \lambda - \sigma) = \int_{-\infty}^{t} h(t, \tau) R_y(\tau - \sigma) \, d\tau, \quad \forall \sigma \leq t.
\]

We can render this expression more simply by making some changes of variable. First, let \( \alpha = t - \tau \); then

\[
R_{xy}(t + \lambda - \sigma) = \int_{0}^{\infty} h(t, t - \alpha) R_y(t - \alpha - \sigma) \, d\alpha, \quad \forall \sigma \leq t.
\]

Next, let \( \xi = t - \sigma \), to obtain

\[
R_{xy}(\xi + \lambda) = \int_{0}^{\infty} h(\xi + \sigma, \xi + \sigma - \alpha) R_y(\xi - \alpha) \, d\alpha, \quad \forall \xi \geq 0.
\]
Since the left-hand side of this expression is independent of \( \sigma \), the right-hand side must also be independent of \( \sigma \), which in turn implies that \( h \) is not a function of \( \sigma \). The only way this can happen is if \( h \) is a function of the difference of its first and second arguments; that is, if \( h \) is a function of \( \alpha \) only. So, we introduce the (abuse of) notation \( h(z_1, z_2) = h(z_1 - z_2) \) and, reverting back to \( t \) as the independent variable, we obtain

\[
R_{xy}(t + \lambda) = \int_0^\infty h(\tau)R_y(\tau)\,d\tau, \quad \forall t \geq 0, \tag{3.71}
\]

the celebrated Wiener–Hopf equation.

Equation (3.71), describing the solution of an optimal filter in continuous time, should be contrasted with (3.56) of chapter 3. In chapter 3, a set of matrix equations is obtained, whereas in the present case an integral equation is obtained. However, the structure in both cases is equivalent: the optimal filter coefficients are operated on by the autocorrelation of the input function to obtain the cross-correlation between the input and output.

Once the Wiener–Hopf equation is solved for \( h \), then

\[
\hat{x}_{t+\lambda} = \int_{-\infty}^{t} h(t-\tau)y_\tau\,d\tau \tag{3.72}
\]

represents the minimum mean-square estimate of \( x_{t+\lambda} \). Solving (3.71), however, involves more than simply taking Fourier or even bilateral Laplace transforms. To see why this is so, take the Laplace transform of both sides of (3.71):

\[
\int_{0-}^{\infty} R_{xy}(t + \lambda)e^{-\sigma t}\,dt = \int_{0-}^{\infty} h(\tau)R_y(t - \tau)e^{-\sigma(t-\tau)}e^{-\sigma t}\,d\tau\,dt \\
= \int_{0-}^{\infty} e^{-\sigma t}h(\tau) \int_{0-}^{\infty} R_y(t - \tau)e^{-\sigma(t-\tau)}\,dt\,d\tau \\
= \int_{0-}^{\infty} e^{-\sigma t}h(\tau) \int_{-\tau}^{\infty} R_y(\sigma)e^{-\sigma \sigma}\,d\sigma\,d\tau, \tag{3.73}
\]

where we make the change of variable \( \sigma = t - \tau \) for the last integral. We observe that the right-hand side of (3.73) is not equal to the product of the Laplace transforms of \( h \) and \( R_y \), since the limits of the inner integral depend on \( \tau \). This condition arises from the requirement that \( t > 0 \) in (3.71). If we did not worry about physical realizability (that is, causality), we could relax the condition that \( h(t) = 0 \) for \( t < 0 \). In this case only, we may obtain, via Fourier analysis, the result that the optimal filter transfer function is given by

\[
H(\omega) = \frac{S_{xy}(\omega)}{S_y(\omega)}; \tag{3.74}
\]

the resulting impulse response function is noncausal unless \( x_t \) and \( y_t \) are white. For applications where causality is not a constraint, this result is perfectly valid. For example, let \( x_t \) be an image (here, \( t \) represents spatial coordinates), and suppose we observe

\[
y_t = x_t + v_t,
\]

where \( \{v_t, -\infty < t < \infty\} \) is a white-noise process with \( R_v(t) = \sigma^2\delta(t) \). It is easy to see that \( R_y(\tau) = R_y(\tau) + \sigma^2\delta(\tau) \) and \( R_y(\tau) = R_y(\tau) \), so

\[
H(\omega) = \frac{S_x(\omega)}{S_y(\omega) + \sigma^2}.
\]

This result admits a very intuitive interpretation. Over frequencies where the signal energy is high compared to the noise, the filter acts as an identity filter and passes the signal without change. Over frequencies where the noise power dominates, the signal filter attenuates the observation.
In some contexts, (3.74) is called a Wiener filter, but that is not quite accurate. More precisely, the Wiener filter is the solution to (3.71), and more sophistication is needed to obtain that solution. The solution comes via the celebrated Wiener–Hopf technique.

As we examine (3.71), we observe that we could solve this equation with transform techniques if \( R_y(\sigma) = 0 \) for \( \sigma < 0 \). Unfortunately, since \( R_y \) is a correlation function, this situation generally will not occur. One notable and important situation in which this does occur, however, is when \( \{y_n\} \) is a white-noise process, for then \( R_y(t) = \delta(t) \). In this case, the solution to (3.71) is trivial:

\[
R_{xy}(t + \lambda) = \int_0^\infty h(\tau)\delta(t - \tau)\,d\tau = h(t), \quad \forall t \geq 0,
\]

so

\[
h(t) = \begin{cases} R_{xy}(t + \lambda) & t \geq 0, \\ 0 & t < 0. \end{cases}
\]

### 3.13.4 Solution to the Wiener–Hopf equation

We will present two approaches to the solution of the Wiener-Hopf equation. The first is based upon careful consideration of the locations of poles. The second is based upon the innovations representation of a process. The second is easier, pointing out again the utility of placing signals in the proper coordinate frame (i.e., a set of orthogonal functions).

**Theorem 3.3** The solution to the Wiener–Hopf equation,

\[
R_{xy}(t + \lambda) = \int_0^\infty h(\tau)R_y(t - \tau)\,d\tau, \quad \forall t \geq 0,
\]

where

\[
h(t) = 0, \quad t < 0,
\]

is

\[
H(s) = \frac{1}{S_y(s)} \left\{ \frac{S_{xy}(s)e^{i\lambda}}{S_y(s)} \right\}^+.
\]

**Proof** We first observe that since \( h(t) \) is to be stable and causal, its bilateral Laplace transform will have no poles in the RHP. The transform of \( R_{xy}(t) \) is

\[
S_{xy}(s) = \int_{-\infty}^{\infty} R_{xy}(t)e^{-st}\,dt.
\]

Consequently,

\[
S_{xy}(s)e^{i\lambda} = e^{i\lambda}\int_{-\infty}^{\infty} R_{xy}(\tau)e^{-s(\tau-i\lambda)}\,d\tau
\]

\[
= \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-s(\tau-i\lambda)}\,d\tau
\]

\[
= \int_{-\infty}^{\infty} R_{xy}(t + \lambda)e^{-st}\,dt,
\]

where we have made the change of variable \( t = \tau - \lambda \) in the last integral.
Let

\[ g(t) = R_{xy}(t + \lambda) - \int_{-\infty}^{t} h(\tau) R_{y}(t - \tau) d\tau. \]

From (3.77), the right-hand side of this equation is zero for \( t \geq 0 \), so

\[ g(t) = \begin{cases} 0 & t \geq 0, \\ \text{unknown} & t < 0. \end{cases} \]

We will establish our result by examining the bilateral Laplace transform of \( g(t) \). Since \( g(t) \) is an anticausal (left-sided) function, its region of convergence is the LHP; consequently \( G(s) \) has no poles in the LHP. Taking the bilateral Laplace transform of \( g(t) \) and using (3.79),

\[ G(s) = S_{xy}(s)e^{\lambda s} - H(s)S_{y}(s). \]

Now, observing that \( S_{y}(s) = S_{y}^{+}(s)S_{y}^{-}(s) \) (canonical factorization) and dividing both sides of this equation by \( S_{y}^{-}(s) \), we obtain

\[ \frac{G(s)}{S_{y}^{-}(s)} = \frac{S_{xy}(s)e^{\lambda s}}{S_{y}^{-}(s)} - \frac{H(s)S_{y}^{+}(s)}{S_{y}^{-}(s)}. \]  

(3.80)

Since \( G(s) \) has no LHP poles and \( S_{y}^{-}(s) \) has no LHP zeros, \( \frac{G(s)}{S_{y}^{-}(s)} \) has no LHP poles. Furthermore, \( H(s) \) has no RHP poles, and neither does \( S_{y}^{+}(s) \), so the product \( H(s)S_{y}^{+}(s) \) has no RHP poles. The quantity \( \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \), however, may have poles in both the RHP and the LHP. The only way equality can obtain is for the LHP poles of \( \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \) to be equal to the poles of \( H(s)S_{y}^{+}(s) \). Let \( \phi(t) \) be the inverse Laplace transform of \( \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \); that is,

\[ \phi(t) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} e^{st} ds. \]

\( \phi(t) \) will, in general, be a two-sided function. The LHP poles of the bilateral Laplace transform of \( \phi(t) \), however, may be obtained by taking the Laplace transform of \( \phi(t)u(t) \). In other words, applying the \( \{\cdot\}_+ \) operation to both sides of (3.80) yields

\[ 0 = \left\{ \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \right\}_+ - H(s)S_{y}^{+}(s), \]

and, consequently,

\[ H(s) = \frac{S_{xy}(s)}{S_{y}^{+}(s)} \left\{ \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \right\}_+. \]

It may be useful to compare the causal Wiener filter,

\[ H(s) = \frac{1}{S_{y}^{+}(s)} \left\{ \frac{S_{xy}(s) e^{\lambda s}}{S_{y}^{-}(s)} \right\}_+, \]

with the noncausal "Wiener filter,"

\[ H(s) = \frac{S_{xy}(s)}{S_{y}(s)}. \]

We note that, except for the \( \{\cdot\}_+ \) operation, they are the same, so the structure is not so foreign as it might seem upon first exposure.
Let \( v \) be a white-noise process, and let us use this process to form the estimate \( \hat{x}_t + \lambda \). The second derivation of the Wiener filter is based upon two observations:

- The Wiener–Hopf equation is trivial to solve if the observed process is a white noise, since (repeating (3.75) and (3.76)),
  \[
  R_{xy}(t + \lambda) = \int_0^\infty h(\tau) \delta(t - \tau) \, d\tau = h(t), \quad \forall t \geq 0.
  \]
  so
  \[
  h(t) = \begin{cases} R_{xy}(t + \lambda) & t \geq 0 \\ 0 & t < 0 \end{cases}.
  \]

- A stationary lumped process may be transformed into a white noise without loss of information by means of a causal and causally invertible transform (see figure 3.15).

These two observations permit us to adopt a two-step procedure: (a) first, we "pre-whiten" the observed signal, \( \{y_t\} \), to create an innovations process, \( \{v_t\} \); (b) we then apply the trivial Wiener filter to the pre-whitened signal. In other words, the Wiener filter can be obtained by cascading the pre-whitening filter and the Wiener filter for white-noise observations, as illustrated in figure 3.16.

![Figure 3.16](image)

Figure 3.16: The optimal filter as the cascade of a pre-whitening filter and a Wiener filter with white-noise inputs

From our earlier development, the pre-whitening filter is simply \( W(s) = \frac{1}{S_y(s)} \), the canonical spectral factor of \( S_y(s) \), and the optimal filter based on white-noise observations is \( \{S_{xy}(s)e^{it}\}_+ \), so

\[
H(s) = \frac{1}{S_y^+(s)} \left\{ \frac{S_{xy}(s)e^{it}}{S_y^-(s)} \right\} _+ = \frac{1}{S_y^+(s)} \left\{ S_{xy}(s)e^{it} \right\} _+,
\]

since \( S_y^-(s) = 1 \).

The only thing left to compute is \( \{S_{xy}(s)e^{it}\}_+ \). But

\[
R_{xy}(t) = E x_{a+t} y_d = E \left[ x_{a+t} \int_{-\infty}^\infty w(\beta) y_{a-\beta} \, d\beta \right]
\]

\[
eq \int_{-\infty}^\infty w(\beta) E[x_{a+t} y_{a-\beta}] \, d\beta = \int_{-\infty}^\infty w(\beta) R_{xy}(t + \beta) \, d\beta = R_{xy}(t) * w(-t),
\]

where \( w(t) \) is the inverse transform of \( W(s) \). Consequently,

\[
S_{xy}(s) = S_{xy}(s)W(-s) = S_{xy}(s)\frac{1}{S_y^+(s)} = S_{xy}(s)\frac{1}{S_y^-(s)}.
\]

and therefore

\[
H(s) = \frac{1}{S_y^+(s)} \left\{ \frac{S_{xy}(s)e^{it}}{S_y^-(s)} \right\} _+,
\]

which is the same formula we obtained with our original derivation.
3.13.5 Examples of Wiener filtering

Example 3.13.4 Suppose \( x_t \equiv y_t \) and \( \lambda > 0 \) in (3.68). This is a problem of pure prediction: we wish to obtain an expression for \( x_{t+1} \), given \( \{y_t, \alpha \leq t\} \). Since \( S_y(s) = S_x(s) = S_{y\alpha}(s) \), (3.78) becomes

\[
H(s) = \frac{1}{S_x^+(s)} \left\{ S_y^+(s)e^{i\lambda} \right\}_+. 
\]

Now let \( y_t \) be an Ornstein-Uhlenbeck process, which is a process \( \{y_t, -\infty < t < \infty\} \) with zero-mean and with correlation function

\[
R_y(t) = e^{-\alpha t},
\]

with \( \alpha > 0 \). Then

\[
S_y(s) = \frac{\sqrt{2\alpha}}{\alpha - s^2} \quad S_y^+(s) = \frac{\sqrt{2\alpha}}{\alpha + s}. 
\]

Our task is to compute \( \{\sqrt{2\alpha} e^{i\lambda}\}_+ \). Since this transform is not a rational function of \( s \), we cannot use partial fractions directly, and must appeal to the definition by finding the inverse Laplace transform, then taking the causal part:

\[
F(s) = \frac{\sqrt{2\alpha} e^{i\lambda}}{\alpha + s},
\]

and specifying the region of convergence as \( \text{Re} \ s > -\alpha \). Taking the (stable) inverse Laplace transform, we find

\[
f(t) = \sqrt{2\alpha} e^{-\alpha t} e^{-\alpha t} u(t + \lambda),
\]

which is not causal. We find that the causal part of \( f(t) \) is

\[
f(t)u(t) = \sqrt{2\alpha} e^{-\alpha t} e^{-\alpha t} u(t).
\]

and so, taking transforms,

\[
\{F(s)\}_+ = \frac{\sqrt{2\alpha} e^{-\alpha t}}{s + \alpha}, \quad \text{Re} \ s > -\alpha.
\]

Therefore,

\[
H(s) = \frac{1}{S_x^+(s)} \left\{ S_y^+(s)e^{i\lambda} \right\}_+ 
= \frac{\alpha + s}{\sqrt{2\alpha}} \frac{\sqrt{2\alpha} e^{-\alpha t}}{\alpha + s} 
= e^{-\alpha t},
\]

so the impulse response function of the optimal filter is

\[
h(t) = e^{-\alpha t} \delta(t),
\]

and the optimal predictor is

\[
\hat{x}_{t+\lambda} = \int_{-\infty}^\infty e^{-\alpha \tau} \delta(t - \tau) x_t \, d\tau 
= e^{-\alpha \lambda} x_t.
\]

Thus, the predicted value of \( x_t \) decays from its last observed value exponentially at a rate governed by the correlation time-constant. \( \square \)

Example 3.13.5 Filtering in White Noise. Let

\[
y_t = x_t + \nu_t,
\]
where $b(s^2)$ and $a(s^2)$ are polynomials in $s^2$ with the degree of $b(s^2)$ strictly lower than the degree of $a(s^2)$. Furthermore, assume $R_{\nu}(t) \equiv 0$. Direct calculation yields

$$S_{\nu}(s) = S_s(s),$$

$$S_s(s) = S_s(s) + 1,$$

so

$$H(s) = \frac{1}{S_s^+(s)} \left( \frac{S_s(s)}{S_s^-(s)} - \frac{1}{S_s^-(s)} \right),$$

$$= \frac{1}{S_s^+(s)} \left( \frac{S_s(s)}{S_s^-(s)} - \frac{1}{S_s^-(s)} \right),$$

$$= \frac{1}{S_s^+(s)} \left( \frac{S_s(s)}{S_s^-(s)} - \frac{1}{S_s^-(s)} \right),$$

$$= \frac{1}{S_s^+(s)} \left( S_s(s) - \frac{1}{S_s^-(s)} \right).$$

Now, observe that

$$S_s(s) = \frac{b(s^2)}{a(s^2)} + 1 = \frac{a(s^2) + b(s^2)}{a(s^2)}.$$

Since the degree of $b(s^2)$ is lower than the degree of $a(s^2)$, the degrees of the numerator and denominator of $S_s(s)$ are the same, say of degree $2n$. The canonical factors of $S_s(s)$ will therefore be rational functions with numerators and denominators of degree $n$. Thus, $S_s^-(s)$ is of the form

$$S_s^-(s) = \frac{\beta(s)}{\alpha(s)} = \frac{1 + \gamma(s)}{\alpha(s)},$$

since the leading coefficients of both $\beta(s)$ and $\alpha(s)$ are the same. Since the rational function $\frac{\beta(s)}{\alpha(s)}$ has all of its poles in the RHP, we immediately obtain (see (3.67))

$$\left\{ \frac{1}{S_s^-(s)} \right\}_r = 1. \quad (3.81)$$

Thus,

$$H(s) = \frac{1}{S_s^+(s)} \left( S_s^+(s) - \frac{1}{S_s^-(s)} \right),$$

$$= 1 - \frac{1}{S_s^+(s)}. \quad (3.82)$$

**Example 3.13.6** As an application of the results of the previous problem, consider the case when

$$S_s(s) = \frac{s^2 - 9}{s^4 - 5s^2 + 4} = \frac{(s + 3)(s - 3)}{(s^2 + 3s + 2)(s^2 - 3s + 2)},$$

so that

$$S_s(s) = \frac{s^4 - 4s^2 - 5}{s^4 - 5s^2 + 4} = \frac{s^2 + (\sqrt{5} + 1)s + \sqrt{5}}{s^2 + 3s + 2} - \frac{(\sqrt{5} + 1)s + \sqrt{5}}{s^2 - 3s + 2} = S_s^+(s)S_s^-(s).$$
and, thus,

\[
\frac{1}{S^*_y(s)} = \frac{s^2 - 3s + 2}{s^2 - (\sqrt{5} + 1)s + \sqrt{5}} = 1 + \frac{\sqrt{5} - 2}{s^2 - (\sqrt{5} + 1)s + \sqrt{5}}.
\]

The inverse Laplace transform of \(1/S^*_y(s)\) is anticausal, except for the constant term 1, so that, as in (3.81), the portion due to the causal part is

\[
\left\{ \frac{1}{S^*_y(s)} \right\} = 1 + \left\{ \frac{y(s)}{\alpha(s)} \right\} = 1.
\]

Then

\[
H(s) = 1 - \frac{1}{S^*_y(s)} = \frac{(\sqrt{5} - 2)s + \sqrt{5} - 2}{s^2 + (\sqrt{5} + 1)s + \sqrt{5}}.
\]

### 3.13.6 Mean-square error

The error associated with the Wiener filtering problem is given by

\[
\hat{x}_{t+\lambda} = x_{t+\lambda} - \hat{x}_{t+\lambda},
\]

with

\[
\hat{x}_{t+\lambda} = \int_{-\infty}^{t} h(t - \tau) y_{\tau} \; d\tau,
\]

where

\[
h(t) = \mathcal{L}^{-1}\{H(s)\},
\]

which is the inverse Laplace transform of the optimal transfer function given by (3.78). The covariance of the estimation error is

\[
E \hat{x}^2_{t+\lambda} = E[x_{t+\lambda} - \hat{x}_{t+\lambda}]^2 = E \left[ x_{t+\lambda} - \int_{-\infty}^{t} h(t - \tau) y_{\tau} \; d\tau \right] \left[ x_{t+\lambda} - \int_{-\infty}^{t} h(t - \tau) y_{\tau} \; d\tau \right].
\]

(3.83)

Since \(\hat{x}_{t+\lambda}\) is a function of \(\{y_{\alpha}, \alpha \leq t\}\), the orthogonality condition (3.70) requires that the estimation error be orthogonal to the estimate, that is,

\[
E \left\{ \left[ x_{t+\lambda} - \int_{-\infty}^{t} h(t - \tau) y_{\tau} \; d\tau \right] \hat{x}_{t+\lambda} \right\} = 0,
\]

so (3.83) becomes

\[
E \hat{x}^2_{t+\lambda} = E \left\{ x_{t+\lambda} - \int_{-\infty}^{t} h(t - \tau) y_{\tau} \; d\tau \right\} x_{t+\lambda}
\]

\[
= R_y(0) - \int_{-\infty}^{t} h(t - \tau) R_{yx}(\tau - t - \lambda) \; d\tau
\]

\[
= R_y(0) - \int_{-\infty}^{t} h(t - \tau) R_{yx}(t - \tau + \lambda) \; d\tau
\]

\[
= R_y(0) - \int_{0}^{\infty} h(\alpha) R_{yx}(\alpha + \lambda) \; d\alpha.
\]

where the last equality holds by making the change of variable \(\alpha = t - \tau\).

### 3.13.7 Discrete-time Wiener filters

The Wiener filter theory also applies in discrete time. We have already seen, the Wiener filter results for FIR filters. We now apply the notion of spectral factorization to the Wiener–Hopf equations with causal IIR filters. We summarize the results for this development.
Canonical factorization

Let $S_y(z)$ be the power spectral density of a discrete-time random process. Then $S_y(z)$ has poles inside and outside the unit circle. The canonical factorization is

$$S_y(z) = S_y^+(z)S_y^-(z),$$

where $S_y^+(z)$ has all of its poles and zeros inside the unit circle.

Additive decomposition

Let $f[t], t = \ldots, -1, 0, 1, \ldots,$ be a discrete-time function. Then

$$f[t] = f[t]u[t] + f[t]u[-k] - f[0],$$

where $u[t]$ is the discrete-time unit-step function. The Z-transform of $f[t]$ is

$$F(z) = \sum_{n=-\infty}^{\infty} f[n]z^{-n}$$

$$= \sum_{n=-\infty}^{-1} f[n]z^{-n} + \sum_{n=0}^{\infty} f[n]z^{-n}.$$

Wiener–Hopf equation

Let $x_t$ and $y_t$ be zero-mean, jointly stationary discrete-time stochastic processes. We wish to estimate $x_{t+\lambda}$, given $\{y_j, j \leq t\}$, with an estimator of the form

$$\hat{x}_{t+\lambda} = \sum_{i=-\infty}^{\lambda} h[t - i]y_i, \quad \lambda \text{ an integer } \geq 0,$$

where $h(i)$ is the solution (from orthogonality) to

$$R_{xy}[t + p] = \sum_{i=0}^{\infty} h[i]R_y[t - i], \quad t \geq 0.$$

To solve this equation for $h$ we follow the Wiener–Hopf technique of defining the function

$$g[t] = R_{xy}[t + \lambda] = \sum_{i=0}^{\infty} h[i]R_y[t - i], \quad \text{all } t,$$

$$= \begin{cases} 0, & t \geq 0, \\ \text{unknown,} & t < 0. \end{cases}$$

Since $g[t]$ is an anticausal function, its region of convergence is the interior of the unit circle—it has no poles within the unit circle.
Taking the bilateral $Z$-transform of $g[t]$,

$$G(z) = \sum_{n=-\infty}^{\infty} R_{xy}[n+\lambda]z^{-n} + H(z)S_y(z)$$

$$= z^{\lambda} \sum_{j=\infty}^{\infty} R_{xy}(j)z^{-j} + H(z)S_y(z)$$

$$= z^{\lambda} S_{xy}(z) + H(z)S_y(z). \quad (3.84)$$

The canonical spectral factorization of $S_y(z)$ is of the form

$$S_y(z) = S_y^+(z)S_y^-(z),$$

where $S_y(z)$ has poles and zeros outside the unit circle and $S_y^+(z)$ has poles and zeros inside the unit circle. Dividing both sides of (3.84) by $S_y(z)$ and applying the $|.|_1$ operation to both sides yields

$$H(z) = \frac{1}{S_y^+(z)} \left( z^\lambda S_{xy}(z) \right)^{+} \frac{S_y^-(z)}{S_y(z)}.$$

the discrete-time Wiener filter.

**Example 3.13.7** Let $\{y_t, -\infty < t < \infty\}$ be a discrete-time, zero-mean, wide-sense stationary process with correlation function

$$R_y[t] = \frac{4}{3} \left( \frac{1}{2} \right)^{|t|.}$$

Let $x_t = y_t$, and predict $x_{t+i}$ for $\lambda \geq 0$.

We seek a predictor of the form

$$\tilde{x}_{t+i} = \sum_{n=-\infty}^{\infty} h[t-i]y_t, \quad \lambda \geq 0.$$ 

We have $R_y[t] = R_{xy}[t] = R_y[t]$, and

$$S_y(z) = \sum_{n=\infty}^{\infty} R_y[n]z^{-n} = \sum_{n=\infty}^{\infty} \frac{4}{3} \left( \frac{1}{2} \right)^n z^{-n} + \sum_{n=0}^{\infty} \frac{4}{3} \left( \frac{1}{2} \right)^n z^{-n}$$

$$= \sum_{j=0}^{\infty} \frac{4}{3} \left( \frac{z}{2} \right)^j - \frac{4}{3} + \sum_{n=0}^{\infty} \frac{4}{3} \left( \frac{1}{2} \right)^n z^{-n} = \frac{4}{3} \left[ \frac{1}{1 - \frac{1}{2}} - 1 + \frac{1}{1 - \frac{1}{2} z} \right]$$

$$= \frac{1}{S_y^+(z)} \frac{1}{S_y^-(z)}.$$ 

Next, we calculate

$$\frac{z^{\lambda} S_{xy}(z)}{S_y^+(z)} = \frac{z^{\lambda}}{(1 - \frac{1}{\tilde{z}})(1 - \frac{1}{z})} = \frac{2^{\lambda+1}}{2z - 1}. $$

By long division, we obtain

$$\frac{2^{\lambda+1}}{2z - 1} = z^\lambda + \frac{1}{2} z^{\lambda-1} + \frac{1}{4} z^{\lambda-2} + \cdots + \frac{1}{2^{\lambda+1}} z^0 + \frac{1}{2^{\lambda+2}} z^{-1} + \cdots$$

$$= \frac{1}{2^{\lambda+1}} \sum_{n=\infty}^{\infty} \frac{1}{2^n} z^{-n}. $$
We may obtain the inverse $Z$-transform as the coefficients of $z^{-n}$. The operation $\{\cdot\}_+$ is effected by discarding all samples before $n = 0$ and returning to the transform domain:

$$\left\{ \frac{z^n S_{\epsilon r}(z)}{S_r(z)} \right\}_+ = \frac{1}{2^k} \sum_{n=0}^\infty \left( \frac{1}{2} \right)^n z^{-n} = \frac{1}{2^k} \frac{1}{1 - \frac{1}{2^k}},$$

and

$$H(z) = \frac{1}{S_r(z)} \left\{ z^n S_{\epsilon r}(z) \right\}_+ = \frac{1}{2^k},$$

so

$$h[t] = \frac{1}{2^k} \delta_t,$$

and

$$\hat{x}_{\epsilon r} = \left( \frac{1}{2} \right)^k y_t.$$  

\[\Box\]

### 3.14 A dual approximation problem

The approximation problems we have seen up till now have selected a point from a finite-dimensional subspace of the Hilbert space of the problem. In each case, because the solution was in a finite-dimensional subspace, solving an $m \times m$ system of equations was sufficient. In some approximation problems, the subspace in which the solution lies is not finite dimensional, so a simple finite set of equations cannot be solved to obtain the solution. There are some problems, however, in which a finite set of constraints provides us with sufficient information to solve the problem from a finite set of equations.

We begin with a definition.

**Definition 3.2** Let $M$ be a subspace of a linear space $S$, and let $x_0 \in S$. The set $V = x_0 + M$ is said to be a **translation** of $M$ by $x_0$. This translation is called a **linear variety**.

A linear variety is not in general a subspace.

**Example 3.14.1** Let $M = \{(0, 0, 0), (0, 1, 0)\}$ in the vector space $(GF(2))^3$ introduced in example 2.12.1, and let $x_0 = (1, 1, 1) \in S$. Then

$$x + M = \{(1, 1, 1), (1, 0, 1)\}$$

is a linear variety.  

A version of the orthogonality theorem appropriate for linear varieties is illustrated in figure 3.17. Let $V = x_0 + M$ be a closed linear variety in a Hilbert space $H$. Then there is a **unique** vector $v_0 \in V$ of minimum norm. The minimizing vector $v_0$ is orthogonal to $M$. This result is an immediate consequence of the projection theorem for Hilbert spaces (simply translate the variety and the origin by $-x_0$).

![Figure 3.17: Minimum norm to a linear variety](image-url)

\[\Box\]
Let $S$ be a Hilbert space. Given a set of linearly independent vectors $y_1, y_2, \ldots, y_m \in S$, let $M = \text{span}(y_1, y_2, \ldots, y_m)$. The set of $x \in S$ such that

$$
\langle x, y_1 \rangle = 0 \\
\langle x, y_2 \rangle = 0 \\
\vdots \\
\langle x, y_m \rangle = 0
$$

is a subspace, which (because of these inner-product constraints) must be $M^\perp$. Suppose now we have a problem in which there are inner-product constraints of the form

$$
\langle x, y_1 \rangle = a_1 \\
\langle x, y_2 \rangle = a_2 \\
\vdots \\
\langle x, y_m \rangle = a_m.
$$

(3.85)

If we can find any point $x = x_0$ that satisfies the constraints in (3.85), then for any $v \in M^\perp$, $x_0 + v$ also satisfies the constraints. Hence the space of solutions of (3.85) is the linear variety $V = x_0 + M^\perp$. A linear variety $V$ satisfying the $m$ constraints in (3.85) is said to have codimension $m$, since the orthogonal complement of the subspace $M^\perp$ producing it has dimension $m$.

Example 3.14.2 In $\mathbb{R}^3$, let $y_1 = (1, 0, 0)$ and $y_2 = (0, 1, 0)$, and let $M = \text{span}(y_1, y_2)$. The set of points such that

$$
\langle x, y_1 \rangle = 0 \\
\langle x, y_2 \rangle = 0
$$

is

$$\text{span}(0, 0, 1) = M^\perp.$$

Now, for the constraints

$$
\langle x, y_1 \rangle = 3, \\
\langle x, y_2 \rangle = 4.
$$

observe that if $x = (3, 4, s)$ for any $s \in \mathbb{R}$ then the constraints are satisfied. The set $V = (3, 4, 0) + M^\perp$ is a linear variety of codimension 2.

We are now in a position to state the minimization problem.

**Theorem 3.4** (Dual approximation) Let $\{y_1, y_2, \ldots, y_m\}$ be linearly independent in a Hilbert space $S$, and let $M = \text{span}(y_1, \ldots, y_m)$. The element $x \in S$ satisfying

$$
\langle x, y_1 \rangle = a_1 \\
\langle x, y_2 \rangle = a_2 \\
\vdots \\
\langle x, y_m \rangle = a_m.
$$

(3.86)

with minimum norm lies in $M$: specifically,

$$
x = \sum_{i=1}^{m} c_i y_i.
$$
where the coefficients in this linear combination satisfy

\[
\begin{bmatrix}
\langle y_1, y_1 \rangle & \langle y_2, y_1 \rangle & \cdots & \langle y_m, y_1 \rangle \\
\langle y_1, y_2 \rangle & \langle y_2, y_2 \rangle & \cdots & \langle y_m, y_2 \rangle \\
\vdots & & \ddots & \vdots \\
\langle y_1, y_m \rangle & \langle y_2, y_m \rangle & \cdots & \langle y_m, y_m \rangle
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_m
\end{bmatrix}
= \begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_m
\end{bmatrix}.
\]  

(3.87)

**Proof** By the discussion above, the solution lies in the linear variety \( V = x_0 + M \perp \) for some \( x_0 \). Furthermore, the optimal solution \( x \) is orthogonal to \( M \perp \), so that \( x \in M^{\perp \perp} = M \).

Thus, \( x_0 \) is of the form

\[ x = \sum_{i=1}^{m} c_i y_i. \]

Taking inner products of this equation with \( y_1, y_2, \ldots, y_m \), and recognizing that, for the solution, \( \langle x_0, y_i \rangle = a_i \), we obtain the set of equations in (3.87).

Example 3.14.3 For the linear variety of the previous problem, let us find the solution of minimum norm. Using (3.87), we find \( x = (3, 4, 0) \) to be the minimum norm solution satisfying the constraints.

Example 3.14.4 We examine here a problem in which the solution space is infinite dimensional. Suppose we have an LTI system with causal impulse response \( h(t) = e^{-2t} + 3e^{-6t} \), in which the initial conditions are \( y(0) = 0 \) and \( \dot{y}(0) = 0 \). We desire to determine an input signal \( x(t) \) so that the output \( y(t) = x(t) * h(t) \) satisfies the constraints

\[ y(1) = 1, \quad \int_{0}^{1} y(t) \, dt = 0 \]

in such a way that the input energy \( \int_{0}^{1} |x(t)|^2 \, dt \) is minimized. Writing the convolution integral for the first output, the first constraint can be written

\[ \int_{0}^{1} (e^{-2(1-t)} + 3e^{-6(1-t)}) x(\tau) \, d\tau = 1. \]

Using the inner product

\[ \langle f, g \rangle = \int_{0}^{1} f(\tau) g(\tau) \, d\tau, \]

the first constraint can be written as

\[ \langle x, y_1 \rangle = 1, \]

where

\[ y_1(\tau) = e^{-2(1-\tau)} + 3e^{-6(1-\tau)}. \]

The second constraint can be written using the integral of the impulse response (see exercise 3.14-27),

\[ k(t) = \int_{0}^{t} h(\tau) \, d\tau = \frac{5}{4} - \frac{3}{4} e^{-4t} - \frac{1}{2} e^{-2t}. \]

Then the second constraint is

\[ \langle x, y_2 \rangle = 0. \]
where
\[ y_2(t) = \frac{5}{4} - \frac{3}{4}e^{-4(t-\tau)} - \frac{1}{2}e^{-2(t-\tau)}. \]
The solution \( x_0(t) \) must lie in the space spanned by \( y_1 \) and \( y_2 \).
\[ x_0 = c_1 y_1(t) + c_2 y_2(t). \]
Then the equation (3.87) becomes
\[
\begin{bmatrix}
(y_1, y_1) & (y_1, y_2) \\
(y_1, y_2) & (y_2, y_2)
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix}
= \begin{bmatrix}
2.36756 & 0.682808 \\
0.682808 & 0.818254
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix}
= \begin{bmatrix}
-1 \\
0
\end{bmatrix},
\]
which has solution
\[ [c_1 \ c_2]^T = [0.5562 \ -0.4642]. \]

3.15 Minimum-norm solution of underdetermined equations

The solution to the dual approximation problem provides a method of finding a least-squares solution to an underdetermined set of equations.

Example 3.15.1 Suppose that we are to solve the set of equations
\[
\begin{bmatrix}
1 & 2 & -3 \\
-5 & 4 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= \begin{bmatrix}
-4 \\
6
\end{bmatrix},
\]
(3.88)

One solution is
\[ \mathbf{x} = \begin{bmatrix}
1 \\
2 \\
3
\end{bmatrix}. \]
However, observe that the vector \( \mathbf{v} = [1, 1, 1]^T \) is in the nullspace of \( A \), so that \( Av = 0 \); any vector of the form
\[ \begin{bmatrix}
1 \\
2 \\
3
\end{bmatrix} + t \begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix} \]
for \( t \in \mathbb{R} \) is also a solution to (3.88).

When solving \( m \) equations with \( n \) unknowns with \( m < n \), unless the equations are inconsistent, as in the example
\[
\begin{bmatrix}
1 & 2 & 3 \\
2 & 4 & 6
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= \begin{bmatrix}
4 \\
7
\end{bmatrix},
\]
there will be an infinite number of solutions.

Let \( \mathbf{x} \) be a solution of \( A\mathbf{x} = \mathbf{b} \), where \( A \) is an \( m \times n \) matrix with \( m < n \), and let \( N = \mathcal{N}(A) \). Then, if \( x_0 \) is a solution to \( A\mathbf{x} = \mathbf{b} \), so is any vector of the form \( x_0 + \mathbf{n} \), where \( \mathbf{n} \in \mathcal{N} \). If the nullspace is not trivial, a variety of solutions are possible. In order to have a well-determined algorithm for uniquely solving the problem, some criterion must be established regarding which solution is desired. A reasonable criterion is to find the solution \( \mathbf{x} \) of smallest norm. That is, we want to
\[
\text{minimize } \|\mathbf{x}\| \\
\text{subject to } A\mathbf{x} = \mathbf{b}.
\]
The minimum norm solution is appealing from a numeric standpoint, because representations of small numbers are usually easier than representations of large numbers. It also leads to a unique solution that can be computed using the formulation of the dual problem of the previous section.

Let us write $A$ in terms of its rows as

$$A = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_m^T \end{bmatrix}.$$ 

Then we observe that the equation $Ax = b$ is equivalent to

$$y_1^T x = b_1 \\
y_2^T x = b_2 \\
\vdots \\
y_m^T x = b_m.$$ 

Our constraint equation therefore corresponds to $m$ inner-product constraints of the sort shown in (3.85). By theorem 3.4, the minimum-norm solution must be of the form

$$x = \sum_{i=1}^{m} c_i y_i,$$ 

(3.89)

where the $c_i$ are the solution to (3.87). We can write (3.89) as

$$x = A^H c,$$ 

(3.90)

where

$$A^H = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}.$$ 

Furthermore, in matrix notation we can write (3.87) in the form

$$(AA^H)c = b.$$ 

Provided that the rows are linearly independent, the matrix $AA^H$ is invertible and we can solve for $c$ as

$$c = (AA^H)^{-1} b.$$ 

Substituting this into (3.90), we obtain the minimum-norm solution

$$x = A^H (AA^H)^{-1} b.$$ 

(3.91)

**Example 3.15.2** The minimum norm solution to (3.88) found using (3.91) is

$$x = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}.$$ 

The matrix $A^H (AA^H)^{-1}$ is a pseudoinverse of the matrix $A$.

### 3.16 Iterative reweighted LS (IRLS) for $L_p$ optimization

This chapter has focused largely on $L_2$ optimization, because the power of the orthogonality theorem allows analytical expressions to be determined in this case. In this section, we
examine an algorithm for determining solutions to $L_p$ optimization problems for $p \neq 2$. The method relies upon weighted least-squares techniques, but using a different weighting for each iteration.

We begin by examining a weighted least-squares problem. Suppose, as in section 3.1, we wish to determine a coefficient vector $c \in \mathbb{R}^m$ to minimize the weighted norm of the error $e$ in

$$x = Ac + e.$$  

Let $W = S^T S$ be a weighting matrix. Then, to find

$$\min_e e^T W e = \min_e e^T S^T S e,$$

we use (3.24) to obtain

$$c = (A^T S^T S A)^{-1} A^T S^T S x. \quad (3.92)$$

Now consider the $L_p$ optimization problem

$$\min_c \|x - Ac\|_p = \min_c \sum_{i=1}^m |x_i - (Ac)_i|^p. \quad (3.93)$$

Let $c^*$ be the solution to this optimization problem. The problem (3.93) can be written using a weighting as

$$\sum_{i=1}^m w_i |x_i - (Ac)_i|^2$$

where $w_i = |x_i - (Ac^*)_i|^{p-2}$, producing a weighted least-squares problem which has a tractable solution. However, the solution cannot be found in one step, because $c^*$ is needed to compute the appropriate weight. In iterative reweighted least-squares, the current solution is used to compute a weight which is used for the next iteration.

To this end, let $S^{[k]}$ be the weight matrix for the $k$th iteration, and let $c^{[k]}$ be the corresponding weighted least-squares solution obtained via (3.92). The error at the $k$th iteration is

$$e^{[k]} = x - Ac^{[k]}.$$  

Then a new weight matrix $S^{[k+1]}$ is created according to

$$S^{[k+1]} = \text{diag}[|e_1^{[k]}|^{p-2}/2, |e_2^{[k]}|^{p-2}/2, \ldots, |e_m^{[k]}|^{p-2}/2].$$

Using this weight, the weighted error measure at the $(k + 1)$st iteration is

$$e^{[k+1]}(S^{[k+1]})^T S^{[k+1]} e^{[k+1]} = \sum_{i=1}^m |x_i - (Ac^{[k+1]})_i|^p.$$  

If this algorithm converges, then the weighted least-squares solution provides a solution to the $L_p$ approximation problem.

However, it is known that the algorithm as described has slow convergence [45]. A more stable approach has been found; let

$$\hat{e}^{[k+1]} = (A^T (S^{[k+1]})^T S^{[k+1]} A)^{-1} A^T (S^{[k+1]})^T S^{[k+1]} x$$

and

$$e^{[k+1]} = \lambda \hat{e}^{[k+1]} + (1 - \lambda) e^{[k]}$$

for some $\lambda \in (0, 1]$. It has been found [89, 162] that choosing

$$\lambda = \frac{1}{p - 1}.$$
leads to convergence properties of the algorithm similar to Newton’s method (see section 14.4).

One final enhancement has been suggested [43]. A time-varying value of $p$ is chosen, such that near the beginning of the iterative process, $p$ is chosen to be small, then gradually increased until the desired $p$ is obtained. Thus

$$p^{[k]} = \min(p, \gamma p^{[k-1]})$$

is used for some small $\gamma > 1$ (a typical value is $\gamma = 1.5$). Algorithm 3.4 incorporates these ideas.

---

**Algorithm 3.4** Iterative reweighted least-squares

File: irwls.m

---

**Example 3.16.1** $L_p$ optimization methods have been used for filter design [43]. In this example we consider an odd tap-length filter

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n},$$

with $N$ even. The filter frequency response can be written (see section 6.8.2) as

$$H(e^{j\omega}) = e^{-jN\omega/2} H_K(\omega),$$

where

$$H_K(\omega) = \sum_{n=0}^{N/2} b_n \cos(\omega n) = b^T c(\omega).$$

Let $|H_d(\omega)|$ be the magnitude response of the desired filter. We desire to minimize

$$\int_0^\pi |H_f(\omega) - |H_d(\omega)||^p d\omega.$$ 

This can be closely approximated by sampling the frequency range at $L_f$ frequencies $\omega_0, \omega_1, \ldots, \omega_{L_f-1}$, and minimizing

$$\sum_{k=0}^{L_f-1} |H_f(\omega_k) - |H_d(\omega)||^p.$$ 

This is now expressed as a finite-dimensional $L_p$ optimization problem, and the methods of this section apply. Sample code that sets up the matrices, finds the solution, then plots the solution is shown in algorithm 3.5. Results of this for $p = 4$ and $p = 10$ are shown in figures 3.18(a) and (b), respectively. The $p = 10$ result shown closely approximates $L_\infty$ (equiripple) design.

---

**Algorithm 3.5** Filter design using IRLS

File: testirwls.m
Much of the transform theory employed in signal processing is encompassed by representations in an appropriate linear vector space. The set of basis functions for the transformation is chosen so that the coefficients convey desired information about the signal. By determining the basis functions appropriately, different information can be extracted from a signal by finding a representation of the signal in the basis.

In this section, we are largely (but not entirely) interested in approximating continuous-time functions. The metric space is $L_2$, and we deal with an infinite number of basis functions, so somewhat more care is needed than in the previous sections of this chapter.

Finding the best representation (in an $L_2$ norm sense) of a function $x(t)$ as

$$x(t) \approx \sum_{i=0}^{m} c_i p_i(t),$$

where $p_i(t)$ is a set of basis functions, is the approximation problem we have seen already many times. If the basis functions are orthonormal, the coefficients which minimize $\|x - \sum_{i=0}^{m} c_i p_i\|_2$ can be found as $c_i = \langle x, p_i \rangle$. The set of coefficients $\{c_i, i = 1, 2, \ldots, m\}$ provides the best representation (in the least-squares sense) of $x$. The minimum squared error of the series representation is

$$\left\| x - \sum_{i=1}^{m} c_i p_i \right\|^2 = \|x\|^2 - \sum_{i=1}^{m} \langle x, p_i \rangle^2.$$

Since the error is never negative, it follows that

$$\sum_{i=1}^{m} |c_i|^2 = \sum_{i=1}^{m} (x, p_i)^2 \leq \|x\|^2.$$  \hspace{1cm} (3.94)

This inequality is known as Bessel's inequality.

The function $\sum_{i=1}^{m} c_i p_i$, obtained as a best $L_2$ approximation of $x(t)$, is said to be the projection of $x(t)$ onto the space spanned by $\{p_1, p_2, \ldots, p_m\}$. This may be written as

$$x_{\text{proj}(p_1, p_2, \ldots, p_m)}(t).$$

Assume that $x$ and $\{p_i\}$ are in some Hilbert space $H$. If the set of basis functions $\{p_i\}$ is infinite, we can take the limit in (3.94) as $m \to \infty$. The representation of this limit is the
infinite series
\[ y(t) = \sum_{i=1}^{\infty} c_i p_i(t). \]

Since
\[ y_m(t) = \sum_{i=1}^{m} c_i p_i(t) \]
is a Cauchy sequence and the Hilbert space is complete, we conclude that \( y(t) \) is in the Hilbert space. For any orthonormal set \( \{p_i\} \), the best approximation of \( x \) (in the \( L_2 \) sense) is the function \( y \). We now want to address the question of when \( x = y \) for an arbitrary \( x \in H \).

We must first point out that by the "equality" \( x = y \), what we mean is that
\[ \| x - y \| = 0, \]
where the norm is the \( L_2 \) norm (since we are dealing with a Hilbert space). Functions that differ on a set of measure zero are "equal" in the sense of the \( L_2 \) norm. Thus "equal" does not necessarily mean "point-for-point equal," as discussed in section 2.1.3.

We now define a condition under which it is possible to represent every \( x \) using the basis set \( \{p_i\} \).

**Definition 3.3** An orthonormal set \( \{p_i, i = 1, 2, \ldots, \infty\} \) in a Hilbert space \( S \) is complete\(^2\) if
\[ x = \sum_{i=1}^{\infty} (x, p_i) p_i \]
for every \( x \in S \).

**Example 3.17.1** It is straightforward to show (by means of a simple counterexample) that simply having an infinite set of orthonormal functions is not sufficient to establish completeness. In \( L_2[0, 2\pi] \), consider the function \( x(t) = \cos t \). An infinite set of orthogonal functions is \( T = \{p_n(t) = \sin(nt), n = 1, 2, \ldots \} \). In the generalized Fourier series representation
\[ \hat{x}(t) = \sum_{i=1}^{\infty} c_i p_i(t), \]
we find that the coefficients are proportional to
\[ (\cos t, \sin nt) = \int_{0}^{2\pi} \cos(t) \sin(nt) \, dt = 0. \]
Hence \( \hat{x}(t) = 0 \), which is not a good representation. We conclude that the set is not complete.

Some results regarding completeness are expressed in the following theorem, which we state without proof.

**Theorem 3.5** \([177]\) A set of orthonormal functions \( \{p_i, i = 1, 2, \ldots\} \) is complete in an inner product space \( S \) with induced norm if any of the following equivalent statements holds:

1. For any \( x \in S \),
\[ x = \sum_{i=1}^{\infty} (x, p_i) p_i. \]

\(^2\)This concerns completeness of the set of functions, which refers to the representational ability of the functions, not the completeness of the space, which is used to describe the fact that all Cauchy sequences converge. Some authors use "total" in place of complete here.
2. For any \( \epsilon > 0 \), there is an \( N < \infty \) such that for all \( n \geq N \),
\[
\left\| x - \sum_{i=1}^{N} \langle x, p_i \rangle p_i \right\| < \epsilon.
\]
(In other words, we can approximate arbitrarily closely.)

3. Parseval’s equality holds:
\[
\|x\|^2 = \sum_{i=1}^{\infty} \langle x, p_i \rangle^2 \text{ for all } x \in S.
\]

4. If \( \langle x, p_i \rangle = 0 \) for all \( i \), then \( x = 0 \). (This was shown to fail in the last example.)

5. There is no nonzero function \( f \in S \) for which the set \( \{p_i, i = 1, 2, \ldots\} \cup f \) forms an orthogonal set.

For a finite-dimensional space \( S \) of dimension \( m \), to have \( m \) linearly independent functions \( p_k, k = 1, 2, \ldots, m \), is sufficient for completeness.

When \( \{p_i\} \) is a complete basis set, then the sequence \( \{c_1, c_2, \ldots\} \) completely describes \( x \); there is a one-to-one relationship between \( x \) and \( \{c_1, c_2, \ldots\} \). (Except that \( x \) is only unique “up to” a set of measure zero.) We sometimes say that the sequence \( \{c_1, c_2, \ldots\} \) is the transform or the generalized Fourier series of \( x \). Writing
\[
c = \{c_1, c_2, \ldots\}
\]
we can represent the transform relationship as
\[
x \leftrightarrow c.
\]

We can define different transformations depending upon the set of orthonormal basis functions we choose. Since each coefficient in the transform is a projection of \( x \) onto the basis function, the transform coefficient \( c_i \) determines how much of \( p_i \) is in \( x \). If we want to look for particular features of a signal, one way is to design a set of orthogonal basis functions that have those features and compute a transform using those signals.

If \( \{p_i, i = 1, 2, \ldots\} \) is a complete set, there is no error in the representation, so Bessel’s inequality (3.94) becomes an equality,
\[
\|x\|^2 = \sum_{i=1}^{\infty} |c_i|^2.
\]

This relationship is known as Parseval’s equality; it should be familiar in various special cases to signal processors. We can write this as
\[
\|x\| = \|c\|,
\]
where the norm on the left is the \( L_2 \) norm (if \( x \) is a function) and the norm on the right is the \( l_2 \) norm.

For transformations using orthonormal basis sets, the angles are also preserved:

**Lemma 3.1** If \( x \) and \( y \) have a generalized Fourier series representation using some orthonormal basis set \( \{p_i, i = 1, 2, \ldots\} \) in a Hilbert space \( S \), with
\[
x \leftrightarrow c \quad \text{and} \quad y \leftrightarrow b
\]
then
\[
\langle x, y \rangle = \langle c, b \rangle.
\]

**Proof** We can write
\[
x = \sum_{i=1}^{\infty} c_i p_i \quad \text{and} \quad y = \sum_{i=1}^{\infty} b_i p_i,
\]
Then
\[ \langle x, y \rangle = \left\langle \sum_{i=1}^{\infty} c_i p_i, \sum_{j=1}^{\infty} b_j p_j \right\rangle \]
\[ = \sum_{i=1}^{\infty} c_i b_i = \langle c, b \rangle, \quad (3.97) \]

where the cross products in the inner product in (3.97) are zero because of orthogonality.

Example 3.17.2 (Fourier series) The set of functions which are periodic on \([0, 2\pi]\) can be represented using the series
\[ f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2\pi} c_n e^{int}. \]

The basis functions \( p_n(t) = e^{int}/\sqrt{2\pi} \) are orthonormal, since
\[ \int_0^{2\pi} e^{int} e^{-imt} dt = \begin{cases} 0 & n \neq m, \\ 2\pi & n = m. \end{cases} \]

Then from (3.9),
\[ c_n = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(t) e^{-int} dt. \]

By Parseval’s relationship, we have
\[ \int_0^{2\pi} |f(t)|^2 dt = \sum_n |c_n|^2. \]

More commonly, we use the nonnormalized basis functions \( y_n(t) = e^{int} \), so the series is
\[ f(t) = \sum_n b_n e^{int}, \]

absorbing the normalizing constant into the coefficient as
\[ b_n = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-int} dt. \]

In this case, Parseval’s relationship must be normalized as
\[ \int_0^{2\pi} |f(t)|^2 dt = \frac{1}{2\pi} \sum_n |b_n|^2. \]

More generally, for a function periodic with period \( T_0 \), we have the familiar formulas
\[ f(t) = \sum_n b_n e^{-j\omega_0 nt}, \]

where \( \omega_0 = 2\pi/T_0 \), and
\[ b_n = \frac{1}{T_0} \int_0^{T_0} f(t) e^{-j\omega_0 nt} dt. \]

Example 3.17.3 (Discrete Fourier transform (DFT)) A discrete-time sequence \( x[t], t = 0, 1, \ldots, N - 1 \), is to be represented as a linear combination of the functions \( p_k[t] = (1/\sqrt{N})e^{j2\pi k/N} \), by
\[ x[t] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c_k e^{j2\pi k/N}. \]
The inner product in this case is

\[ \langle x[r], y[r] \rangle = \sum_{r=0}^{N-1} x[r]y[r]. \]

It can be shown (see exercise 3.17-32) that the set of basis functions \( \{ p_l[r] \} \) are orthogonal, with

\[ \langle p_k[l], p_l[l] \rangle = \begin{cases} 1 & k \mod l \pmod{N}, \\ 0 & \text{otherwise}. \end{cases} \]

The coefficients are therefore computed by

\[ c_k = \frac{1}{N} \sum_{r=0}^{N-1} x[r]e^{-j2\pi rk/N}. \]

More commonly we use the nonnormalized basis functions \( e^{j2\pi rk/N} \), and shift all of the normalization into the reconstruction formula. Then we have

\[ x[r] = \frac{1}{N} \sum_{k=0}^{N-1} d_k e^{j2\pi rk/N} \]

and

\[ d_k = \sum_{r=0}^{N-1} x[r]e^{-j2\pi rk/N}, \]

which is the usual Fourier transform pair. Parseval’s relationship (under this normalization) is

\[ \sum_{r=0}^{N-1} |x[r]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |d_k|^2. \]

### 3.18 Sets of complete orthogonal functions

There are several sets of complete orthogonal functions that are used in common applications. We will examine a few of the more commonly-used sets, mostly stating results without proofs.

#### 3.18.1 Trigonometric functions

As seen in example 3.17.2, the familiar trigonometric functions employed in Fourier series are orthogonal. They form a complete set of orthogonal functions.

#### 3.18.2 Orthogonal polynomials

As we have seen, one way to obtain orthogonal functions is by means of polynomials. Different sets of orthogonal polynomials are obtained by using different weighting functions, and the inner product is taken over some given interval. Some kinds of orthogonal polynomials arise commonly enough that they have been given names.

Let \( f(t) \) and \( g(t) \) be polynomials, and let \( I \) be a domain of interest, \( I = [a, b] \). The polynomials \( f(t) \) and \( g(t) \) are orthogonal with respect to the weighting function \( w(t) \) if

\[ \langle f, g \rangle_w = 0, \]

where

\[ \langle f, g \rangle_w = \int_a^b w(t)f(t)g(t) \, dt. \]

Using the Gram–Schmidt procedure it is possible to orthogonalize any set of polynomials with respect to any inner product; in particular, the set of polynomials \( 1, t, t^2, \ldots, t^n \) can
be orthogonalized with respect to the weighted inner product. We will denote a set of orthogonal polynomials by $p_0(t)$, $p_1(t)$, $p_2(t)$, and so forth, where the subscript denotes the degree of the polynomial.

It can be shown that the orthogonal polynomials (properly normalized) form a complete orthonormal basis for $L_2[a, b]$. The proof of this (which we do not present here) relies on the Weierstrass theorem, which states that any continuous function on an interval $[a, b]$ can be approximated arbitrarily closely by a polynomial. By this theorem we can establish a basis for $C[a, b]$. Extending this to $L_2[a, b]$ (which contains non-continuous functions) makes use of the fact that every discontinuous function over the interval $[a, b]$ is arbitrarily close to a continuous function.

An interesting fact about orthogonal polynomials is the following.

**Lemma 3.2** Orthogonal polynomials satisfy the recursion

$$tp_n(t) = a_n p_{n+1}(t) + b_n p_n(t) + c_n p_{n-1}(t)$$

for $n = 1, 2, \ldots$.

**Proof** Choose $a_n$ so that $tp_n(t) - a_n p_{n+1}(t)$ is of degree $n$.

$$tp_n(t) - a_n p_{n+1}(t) = g_n(t).$$

Then $g_n(t)$ can be written as a linear combination of $p_0, p_1, \ldots, p_n$:

$$g_n(t) = \sum_{i=0}^{n} d_i p_i(t),$$

where the coefficients are obtained by

$$d_i = \langle g_n(t), p_i(t) \rangle.$$\hspace{1cm}

But for $i < n - 1$, the coefficients are zero, since

$$\langle tp_n, p_i \rangle = \langle p_n, tp_i \rangle,$$

and that $p_n$ is orthogonal to all polynomials of lesser degree, including $tp_i$. When $i = n - 1$ and $i = n$, the coefficients are not zero,

$$b_n = \langle g_n, p_n \rangle \quad c_n = \langle g_n, p_{n-1} \rangle.$$\hspace{1cm}□

**Families of orthogonal polynomials**

A variety of types of orthogonal polynomials have been explored over the years. One of the general motivations for this is that orthogonal polynomials can be used to provide solutions to particular differential equations. Since these orthogonal polynomials form a complete orthogonal basis, they can be used to form series solutions for any boundary conditions and input function. Details of this kind of analysis are not discussed here, but may be found in applied mathematics or partial differential equations books, such as [259, 177]. However, the differential equations and several common orthogonal polynomials are presented in the exercises. Another important use of orthogonal polynomials is for Gaussian quadrature, which is an efficient method of numerical integration. This is also derived in the exercises; more details can be found in [265]. In this section we examine only two families of orthogonal polynomials, the Legendre and the Chebyshev polynomials.

**Legendre polynomials**

The Legendre polynomials are not the most commonly used orthogonal polynomials in signal processing, but occasional uses do arise. The Legendre polynomials use a weighting
function \( w(t) = 1 \) over the interval \([a, b] = [-1, 1]\). The first three are

\[
p_0(t) = 1 \quad p_1(t) = t \quad p_2(t) = t^2 - \frac{1}{3} \quad p_3(t) = t^3 - \frac{3}{5} t.
\]

Additional values can be obtained using the recurrence (3.98), which specializes for Legendre polynomials to

\[
(3.98) \quad tp_n(t) = \frac{n + 1}{2n + 1} p_{n+1}(t) + \frac{n}{2n + 1} p_{n-1}(t).
\]

Figure 3.19 shows \( p_0, p_1, p_2, p_3, p_4, \) and \( p_5 \). Observe that not all polynomials have the same amount of "ripple." This is to be contrasted with the Chebyshev polynomials, discussed next.

**Chebyshev polynomials**

Chebyshev polynomials are orthogonal with respect to the weighting function \( w(t) = \frac{1}{\sqrt{1 - t^2}} \) over the interval \( I = [-1, 1] \). In particular, if \( T_r(t) \) and \( T_s(t) \) are Chebyshev polynomials, then

\[
\int_{-1}^{1} \frac{1}{\sqrt{1 - t^2}} T_r(t) T_s(t) \, dt = \begin{cases} 
0 & r \neq s, \\
\pi & r = s = 0, \\
\frac{\pi}{2} & r = s \neq 0.
\end{cases}
\]

(See exercise 2.8-51) The recurrence relation for Chebyshev polynomials is

\[
T_{n+1}(t) = 2t T_n(t) - T_{n-1}(t), \quad T_0(t) = 1, \quad T_1(t) = t.
\]

The Chebyshev polynomials can be expressed as

\[
T_n(t) = \cos(n \cos^{-1} t).
\]

Using either (3.100) or (3.101), the next few Chebyshev polynomials can be found:

\[
T_2(t) = 2t^2 - 1 \quad T_3(t) = 4t^3 - 3t \quad T_4(t) = 8t^4 - 8t^2 + 1.
\]

The leading coefficient of the Chebyshev polynomial \( T_n(t) \) is \( 2^{n-1} \), so that \( \frac{1}{2^n} T_n(t) \) is a monic polynomial. From (3.101), it is clear that the zeros of \( T_n(t) \) are at

\[
t = \cos \left( \frac{2k + 1}{2n} \pi \right), \quad k = 0, 1, \ldots, n - 1.
\]
Over $[-1, 1]$ there are $n + 1$ extrema (counting the end points) of magnitude 1 at

$$t = \cos \frac{k\pi}{n}, \quad k = 0, 1, n.$$ 

Figure 2.11 illustrates the first six Chebyshev polynomials for $t \in [-1, 1]$. What is remarkable about these polynomials is that over this interval, each of the local extrema (maximum or minimum) takes on the value $\pm 1$. This is an important feature in the Chebyshev polynomials, and accounts for most of their applications. This is called the minimum maximum amplitude property: the maximum amplitude (deviation from zero) is minimized.

**Theorem 3.6** Of all monic polynomials of degree $n$, only the polynomial $Q_n(t) = \frac{1}{2^n} T_n(t)$ (the scaled Chebyshev polynomial) oscillates with the minimum maximum amplitude on the interval $[-1, 1]$.

**Proof** The proof is by contradiction. Suppose there exists a monic polynomial $q_n(t)$ of degree $n$ with smaller minimum maximum amplitude on $[-1, 1]$. Let

$$p_{n-1}(t) = Q_n(t) - q_n(t).$$

Since both $q_n$ and $Q_n$ are monic, $p_{n-1}$ must have degree not exceeding $n - 1$. The polynomial $Q_n$ has $n + 1$ extrema, each of magnitude $1/2^n$. By assumption, $q_n(t)$ has a smaller magnitude at each of these extrema, so that $p_{n-1}(t)$ has the same sign as $Q_n(t)$ at each of these extrema. Note that the $n + 1$ extrema of $T_n(t)$, and hence $Q_n(t)$, alternate in sign.

Thus $p_{n-1}(t)$ alternates in sign from one extremum of $Q_n(t)$ to the next. Since there are $n + 1$ extrema, there must be $n$ zeros of $p_{n-1}(t)$ in $[-1, 1]$. But $p_{n-1}(t)$ is a polynomial of degree $n - 1$, which has only $n - 1$ zeros, which is a contradiction.

Now suppose that $q_n(t)$ is another polynomial having the same minimum maximum amplitude as $Q_n(t)$. If $|q_n(t)| < |Q_n(t)|$ at an extremum, then we again arrive, as before, at a contradiction. On the other hand, if $q_n(t_0) = Q_n(t_0)$ at an extremum $t_0$, then $p_{n-1}(t_0) = 0$ and $p'_{n-1}(t_0) = 0$. Then $p_{n-1}(t)$ has (at least) a double zero at $t_0$. Counting the zeros of $p_{n-1}(t)$ again leads to a contradiction. \[\square\]

One application of Chebyshev polynomials is as basis functions in a series expansion, such as

$$f(t) = \sum_{j=0}^{\infty} c_j T_j(t).$$

This series converges uniformly whenever $f(t)$ is continuous and of bounded variation in $[-1, 1]$. Because of the minimum maximum property of Chebyshev polynomials, the approximate representation up to $m$th degree polynomials,

$$f(t) \approx \sum_{j=0}^{m} c_j T_j(t),$$

usually has less error than a corresponding representation using either the basis $1, t, \ldots, t^m$ or the Legendre polynomials.

### 3.18.3 Sinc functions

The function commonly known as a sinc function,

$$\text{sinc}(t) = \frac{\sin(\pi t)}{(\pi t)},$$

can be used to form a set of orthogonal functions

$$p_k(t) = \text{sinc}(2B(t - k/2B)). \quad (3.102)$$
It can be shown (see exercise 3.18-39) for the inner product
\[ (f, g) = \int_{-\infty}^{\infty} f(t)g(t) \, dt \]
that \( (p_k(t), p_l(t)) = \frac{1}{\sqrt{B}} \delta_{kl}. \) If \( f(t) \) is a bandlimited function such that its Fourier transform satisfies
\[ F(\omega) = 0 \quad \text{for} \ \omega \notin (-2\pi B, 2\pi B), \]
then, in the series representation
\[ f(t) = \sum_{k} c_k p_k(t), \]
the coefficients are found to be (see exercise 3.18-39)
\[ c_k = \frac{\langle f, p_k \rangle}{\langle p_k, p_k \rangle} = f(k/2B). \tag{3.103} \]
This gives rise to the familiar sampling theorem representation of a bandlimited function,
\[ f(t) = \sum_{k} f(k/2B) \frac{\sin(2\pi B(t - k/(2B))}{2\pi B(t - k/(2B))}. \]

### 3.18.4 Orthogonal wavelets

Recently, a set of functions known as wavelets has sparked considerable interest. Like the Fourier transform, the wavelet transform can provide information about the spectral content of a signal. However, unlike a sinusoidal signal with infinite support, wavelets are pulses which are well localized in the time domain so that they can provide different spectral information at different time locations of a signal. In doing this, they sacrifice some of their spectral resolution: by the uncertainty principle, we cannot localize perfectly well in both the time domain and the frequency domain. Wavelets have another property that make them practically useful. When used to analyze lower-frequency components, a wide wavelet signal is used; to analyze higher-frequency components, a narrow wavelet signal is used. Thus wavelets can (in principle) identify short bursts of high-frequency signals imposed on top of ongoing low-frequency signals. One of the major principles of wavelet analysis is that it takes place on several scales, using basis functions of different widths.

There are, in fact, several families of wavelets, each with its own properties and associated transforms. Not all families of wavelets form orthogonal waveforms. A particular family of wavelets that has perhaps attracted the most attention is known as the Daubechies wavelets. These wavelets, which form a complete set, have some very nice orthogonality properties that lead to fast computational algorithms. The Daubechies wavelets can be understood best in the context of a Hilbert space, using what is known as a multiresolution analysis. This involves projecting a function onto a whole series of spaces with different resolutions. We now present a brief introduction to the construction of these wavelets. Considerably more information is provided in the literature cited in the references, including generalization in a variety of useful ways of the concepts outlined here.

**Characterization of wavelets**

Throughout this section we will assume real functions for convenience. Most of these concepts can be generalized to functions of complex numbers. Suppose we have a set of closed subspaces of the Hilbert space \( L_2(\mathbb{R}) \), denoted by \( \ldots, V_{-1}, V_0, V_1, \ldots \), with the
following properties:

1. Nesting:

   \[ \cdots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \cdots . \]

2. Closure:

   \[ \text{closure} \left( \bigcup_{j \in \mathbb{Z}} V_j \right) = L_2(\mathbb{R}) ; \]

   that is, the closure of the set of spaces covers all of \( L_2(\mathbb{R}) \), so that every function in \( L_2 \) has a representation using elements in one of these nested spaces.

3. Shrinking:

   \[ \bigcap_{j \in \mathbb{Z}} V_j = \{0\} . \]

4. The “multiresolution” property is obtained by the requirement that if \( f(t) \in V_j \), then \( f(2^j t) \in V_0 \).

5. If \( f(t) \in V_0 \), then \( f(t - n) \in V_0 \) for all \( n \in \mathbb{Z} \).

6. Finally, there is some \( \phi \in V_0 \) such that the integer shifts of \( \phi \) form an orthonormal basis for \( V_0 \):

   \[ V_0 = \text{span}\{\phi(t - n), n \in \mathbb{Z}\} . \]

The function \( \phi(t) \) is said to be a **scaling function**. The property that \( \phi(t) \perp \phi(t - n) \) for \( n \in \mathbb{Z} \) is called the **shift orthogonality** property.

We will use the notation \( P_j f(t) \) to denote the projection of the function \( f(t) \) onto \( V_j \).

**Example 3.18.1** Let

\[ \phi(t) = u(t) - u(t - 1) \quad (3.104) \]

(a unit pulse), and form

\[ V_0 = \text{span}\{\phi(t - n), n \in \mathbb{Z}\} . \]

The set of functions \( \{\phi(t - n), n \in \mathbb{Z}\} \) forms an orthonormal set. Then functions in \( V_0 \) are functions that are **piecewise constant** on the integers. Figure 3.20 shows a function \( f(t) \), the projection \( P_0 f(t) \)—the nearest function to \( f(t) \) that is piecewise constant in the integers—and \( P_{-1} f(t) \)—which is piecewise constant on the half-integers.

As \( j \) decreases, the projection \( P_j f(t) \) represents \( f(t) \) with increasing fidelity.

Let us define the scaled and shifted version of the function \( \phi \) by

\[ \phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j} t - k) . \]

The index \( j \) controls the **scale** and the index \( k \) controls the location of the function \( \phi_{j,k} \). If \( \phi \) is normalized so that \( \|\phi(t)\| = 1 \), then so is \( \phi_{j,k}(t) \) for any \( j \) and \( k \). Since \( \phi(t) \in V_0 \subset V_{-1} \) and \( \phi_{-1,k}(t) \) form an orthonormal basis for \( V_{-1} \), it must be possible to express \( \phi(t) \) as a linear combination of \( \phi_{-1,k}(t) \):

\[ \phi(t) = \sum_k h_k \phi_{-1,k}(t) = \sqrt{2} \sum_k h_k \phi(2t - k) . \quad (3.105) \]

The set of coefficients in (3.105) determines the particular properties of the scaling function and the entire wavelet decomposition. Let \( N \) denote the total number of coefficients \( h_k \) in
Figure 3.20: A function $f(t)$ and its projection onto $V_0$ and $V_{-1}$

(3.105). In general, $N$ could be infinite, but in practice it is always a finite number. We also generally assume that the coefficients $h_k$ are indexed so that $h_k = 0$ for $k < 0$. Let us define $c_k = \sqrt{2}h_k$. Then we can write

$$
\phi(t) = \sum_k c_k \phi(2t - k);
$$

or, given our assumptions, we can write this more precisely as

$$
\phi(t) = \sum_{k=0}^{N-1} c_k \phi(2t - k).
$$

An equation of the form (3.107) is known as a two-scale equation.

Example 3.18.2 In (3.107) let us have two coefficients, $c_0 = 1$ and $c_1 = 1$. Then the two-scale equation becomes

$$
\phi(t) = \phi(2t) + \phi(2t - 1).
$$

It is straightforward to verify that the pulse in (3.104) satisfies this equation.

Lemma 3.3 If $\phi(t)$ satisfies a two-scale equation (3.106) and $\phi(t) \perp \phi(t - n)$ for all $n \in \mathbb{Z}$ with $n \neq 0$, then

$$
\sum_k c_k c_{-2n} = 2\delta_{0p}.
$$

Proof Using (3.106), we have

$$
\int \phi(t)\phi(t - n) \, dt = \int \sum_k c_k \phi(2t - k) \sum_j c_j \phi(2(t - n) - j) \, dt
$$

$$
= \frac{1}{2} \sum_j \left[ \sum_k c_k c_{k+2j} \right] \int \phi(t)\phi(t - j) \, dt.
$$
In order for this to be zero (because of the orthogonality), the bracketed term must be zero when \( j = 0 \) and \( 2n \neq 0 \). Then \( \sum_k c_k e_{k-2n} = 2\delta_{0n} \).

In going from a projection \( P_{j-1} f(t) \) to a lower-resolution projection \( P_j f(t) \), there is some detail information that is lost in the orthogonal complement of \( V_j \). We can represent this detail by saying that

\[
V_{j-1} = V_j \oplus W_j,
\]

where \( W_j = V_j^\perp \) in \( V_{j-1} \). (The direct sum is interpreted in the isomorphic sense.) Thus, \( W_j \) contains the detail lost in going from \( V_{j-1} \) to \( V_j \). Also (as we shall see), the \( W_j \) spaces are orthogonal, so \( W_j \perp W_j' \) if \( j \neq j' \).

Now we introduce the set of functions \( \psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \) as an orthonormal basis set for \( W_j \), with \( \psi(t) \in W_0 \). The function \( \psi(t) \) is known as a wavelet function, or sometimes as the mother wavelet, since the functions \( \psi_{j,k}(t) \) are derived from it. Since \( V_{-1} = V_0 \oplus W_0 \) and \( \psi(t) \in V_{-1} \), we have

\[
\psi(t) = \sum_k g_k \phi_{-1,k}(t) = \sqrt{2} \sum_k g_k \phi(2t - k).
\]

(3.109)

We desire to choose the \( g_k \) coefficients to enforce the orthogonality of the spaces. It will be convenient to write

\[
d_k = \sqrt{2} g_k.
\]

**Theorem 3.7** If \( \{\phi(t - n), n \in \mathbb{Z}\} \) forms an orthogonal set and

\[
d_k = (-1)^k c_{2M+1-k}
\]

for any \( M \in \mathbb{Z} \), then \( \{\psi_{j,k}(t)\} \) forms an orthogonal set for all \( j, k \in \mathbb{Z} \). Furthermore, \( \psi_{j,k}(t) \perp \phi_{l,m}(t) \) for \( l \geq j \).

**Proof** We begin by showing that \( \{\psi_{j,k}(t)\} \) forms an orthogonal set for fixed \( j \).

\[
\int 2^{-j} \psi(2^{-j}t) \psi(2^{-j}t - k) \, dt = \int \sum_i d_i \phi(2u - l) \sum_m d_m \phi(2(u - k) - m) \, du
\]

(where \( u = 2^{-j}t \))

\[
= \int \left( \frac{1}{2} \sum_i (-1)^i c_{2M+1-i} \phi(x) \right) \times \left( \sum_m (-1)^m c_{2M+1-m} \phi(x + l - 2k - m) \right) \, dx
\]

(where \( x = 2u - l \))

\[
= \frac{1}{2} \sum_j c_j c_{j+2k} \int \phi^2(x) \, dx
\]

(by orthogonality, with \( j = 2M + 1 - l \))

\[
= \delta_{0k}. \quad \text{(using (3.108))}
\]

Now we show that \( \phi_{j,k}(t) \perp \psi_{j,m}(t) \) for all \( k, m \in \mathbb{Z} \), for fixed \( j \). We have

\[
\int \psi_{j,k}(t) \phi_{j,m}(t) \, dt = \int 2^{-j} \psi(2^{-j}t - k) \phi(2^{-j}t - m) \, dt
\]

\[
= \int \psi(u - k) \phi(u - m) \, dt \quad \text{(where } u = 2^{-j}t \text{)}
\]
\[ \int \sum (-1)^j c_{2M+1-j} \phi (2(u - k) - l) \sum_j c_j \phi (2(u - m) - j) \, du \]

\[ = \frac{1}{2} \int \sum (-1)^j c_{2M+1-j} \phi (x) \sum_j c_j \phi (x + l + 2k - 2m - j) \, dx \]

(3.111)

\[ = \frac{1}{2} \sum_i (-1)^i c_{2M+1-1} c_{l-2m} \int \phi^2 (t) \, dt \quad \text{(by orthogonality)} \]

In the summation in (3.111), let \( p = m - k \), so the summation is

\[ S = \sum_i (-1)^i c_{2M+1-j} c_{l-2p}. \]

Now, letting \( j = 2M + 1 - l + 2p \), we can write

\[ S = \sum_j (-1)^j c_{j-2p} c_{2M+1-j} = - \sum_j (-1)^j c_{2M+1-j} c_{j-2p} = -S. \]

Since \( S = -S \), we must have

\[ 0 = S = \sum_i (-1)^i c_{2M+1-j} c_{l-2p}, \]

(3.112)

establishing the desired orthogonality.

Finally, we show that \( \psi_{j,k} \perp \psi_{l,m} \) for all \( j, k, l, m \in \mathbb{Z} \) if \( j \neq l \) and \( k \neq m \). We have already established this for \( j = l \). By the multiscale relationship, \( \psi_{j,k} (t) \in W_j \). Let \( j' < j \), so that \( W_j \subset V_{j'} \). But \( V_{j'} \perp W_{j'} \), so that \( \psi_{j',k} (t) \), which is in \( W_{j'} \), must be orthogonal to \( \psi_{j,k} (t) \).

Example 3.18.3 We have seen that a scaling function \( \phi (t) \) can be formed when \( c_0 = c_1 = 1 \). The wavelet \( \psi (t) \) corresponding to this scaling function is

\[ \psi (t) = \phi (2t) - \phi (2t - 1). \]

A plot of \( \phi (t) \) and \( \psi (t) \) is shown in figure 3.21. The function \( \psi (t) \) is also known as the Haar basis function.

\[ \begin{array}{c}
\text{Figure 3.21: The simplest scaling and wavelet functions}
\end{array} \]

There are several families of orthonormal compactly supported wavelets. Algorithm 3.6 provides coefficients for several Daubechies wavelets (there exist wavelets in this family with coefficients of every positive even length). The transform for these coefficients is called the \( D_N \), where there are \( N \) coefficients. Plots of some of the corresponding scaling and wavelet functions are shown in figure 3.22. We observe that the functions become smoother as the number of coefficients increases.
Wavelet transforms

In the wavelet transform, a function $f(t)$ is expressed as a linear combination of scaling and wavelet functions. Both the scaling functions and the wavelet functions are complete sets. However, it is common to employ both wavelet and scaling functions in the transform representation.

Suppose that we have a projection of $f(t)$ onto some space $V_j$ of sufficient resolution that it provides an adequate representation of the data. Then we have

$$f(t) \approx P_j f(t) = \sum_k (f(t), \phi_{j,k}(t)) \phi_{j,k}(t).$$
Commonly we assume that the data has been scaled so that the initial scale is $j = 0$, so that our starting point is $P_0 f(t)$. Let us call this starting function $f_0(t)$, so that

$$f_0(t) = \sum_n \langle f(t), \phi_{0,n}(t) \rangle \phi_{0,n}(t).$$

For the purposes of the transform, we regard the coefficients of this representation as the representation of $f(t)$. In practice, the set of initial coefficients are simply samples of $f(t)$ obtained by sampling every $T$ seconds. That is, we assume that $(f(t), \phi_{0,n}(t)) \approx f(nT)$ for some sampling interval $T$. Under this approximation, the wavelet transform deals with discrete-time sequences. (Further discussion of this point is provided in [63, page 166].) For convenience of notation, let us denote the sequence $\{(f_0, \phi_{0,n}(t))\}$ as $[c^0_n]$, and let us denote the vector of these values as $c^0$:

$$c^0 = [c^0_0 \ c^0_1 \ c^0_2 \ \ldots]^T.$$

In the wavelet transform, we express $f_0(t)$ in terms of wavelets on longer scales. For example, using (3.109) we have $V_0 = V_1 \oplus W_1$, so that $f_0(t) \in V_0$ can be represented as

$$f_0(t) = \sum_n \langle f_0(t), \psi_1,n(t) \rangle \psi_1,n(t) + \sum_n \langle f_0(t), \phi_1,n(t) \rangle \phi_1,n(t).$$

Let $c^1_n = \langle f_0(t), \phi_{1,n}(t) \rangle$ and $d^1_n = \langle f_0(t), \psi_{1,n}(t) \rangle$, and let us denote

$$f_1(t) = \sum_n \langle f_0(t), \phi_1,n(t) \rangle \phi_1,n(t) = \sum_n c^1_n \phi_1,n(t)$$

and

$$\delta_1(t) = \sum_n \langle f_0(t), \psi_1,n(t) \rangle \psi_1,n(t) = \sum_n d^1_n \psi_1,n(t),$$

where $f_1 \in V_1$ and $\delta_1 \in W_1$. Then

$$f_0(t) = f_1(t) + \delta_1(t). \quad (3.113)$$

Since $f_1 \in V_1$ and $V_1 = V_2 \oplus W_2$, we can split $f_1$ into its projection onto $V_2$ and $W_2$ as

$$f_1(t) = \sum_n \langle f_1(t), \phi_2,n(t) \rangle \phi_2,n(t) + \sum_n \langle f_1(t), \psi_2,n(t) \rangle \psi_2,n(t)$$

$$= \sum_n c^2_n \phi_2,n + \sum_n d^2_n \psi_2,n$$

$$= f_2(t) + \delta_2(t). \quad (3.114)$$

where $f_2(t) \in V_2$, and $\delta_2(t) \in W_2$, and $c^2_n = \langle f_1(t), \phi_2,n(t) \rangle$ and $d^2_n = \langle f_1(t), \psi_2,n(t) \rangle$. Substituting (3.114) into (3.113), we have

$$f_0(t) = \delta_1(t) + \delta_2(t) + f_2(t).$$

We will use the notation $c^j$ and $d^j$ to represent the coefficients $c^j_n$ and $d^j_n$, respectively. We can repeat this decomposition for up to $J$ scales, writing $f_j(t) \in V_j$ on each scale $j = 1, 2, \ldots, J$ as

$$f_j(t) = f_{j+1}(t) + \delta_{j+1}(t). \quad (3.115)$$

so

$$f_0(t) = \sum_{j=1}^J \delta_j(t) + f_J(t).$$

The set of coefficients $[d^1, d^2, \ldots, d^J, c^J]$ collectively are the wavelet transform of the function $f_0(t)$. 
Figure 3.23: Illustration of a wavelet transform

The computations just described are outlined in figure 3.23. Starting from the initial set of coefficients $e^0$, the algorithm successively produces $e^{j+1}$ and $d^{j+1}$ until the $J$th level is reached. The set of coefficients $\{d^1, d^2, \ldots, d^J, e^J\}$ is the wavelet transform of the original data. The coefficients at scale $d^j$ represent the signal on longer scales (lower-frequency band) than the coefficients at scale $d^{j-1}$. The coefficients $e^j$ represents an average of the original data.

While it is conceivable to compute the transform by directly evaluating the indicated inner products, a significantly faster algorithm exists. We note that by (3.107)

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) = 2^{-j/2}\sum_n g_n\phi(2^{-j}t - k - n)$$

$$= \sum_n g_n\phi_{j-1,2k+n}(t)$$

$$= \sum_n g_{n-2k}\phi_{j-1,n}(t). \tag{3.116}$$

When we compute the wavelet transform coefficient $(f_0(t), \psi_{1,k}(t))$, we get

$$\langle f_0(t), \psi_{1,k}(t) \rangle = \sum_n g_{n-2k}(f_0(t), \phi_{0,n}(t)) = \sum_n g_{n-2k}e^0_n. \tag{3.117}$$

To understand this sum better, let us write

$$x_n = g_{-n},$$

and form the vector $x = [x_0, x_1, \ldots, x_{N-1}]$. Let $y = x * e^0$ (convolution); then

$$y_j = \sum_n x_{j-n}e^0_n = \sum_n g_{n-j}e^0_n.$$  

From this we observe that the summation in (3.117) is the convolution of the sequence \{g_{-n}\} with the sequence \{e^0_n\}, in which we retain only the even-numbered outputs.

At a general scale $j$, we compute the wavelet coefficients as

$$\langle f_0, \psi_{j,k}(t) \rangle = \sum_n g_{n-2k}(f_0, \phi_{j-1,n}), \tag{3.118}$$

which is a convolution of the sequence \{g_{-n}\} with the sequence \{(f_0, \phi_{j-1,n})\}, retaining even samples. To compute the coefficients in (3.118), we need to know $(f_0, \phi_{j-1,n})$. However, these can also be obtained efficiently, since

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k)$$

$$= \sum_n h_{n-2k}\phi_{j-1,n}(t). \tag{3.119}$$

so that

$$\langle f_0, \phi_{j,k} \rangle = \sum_n h_{n-2k}(f_0, \phi_{j-1,n}),$$

which is again a convolution followed by decimation by 2.
Putting all the pieces together, the wavelet transform is outlined as follows:

1. Let \( c^0_k = (f_0, \phi_{0,k}) \) be the given initial data. (Normally a sequence of samples of \( f(t) \).)

2. Compute the set of wavelet coefficients on scale 1, \( d^1_k = (f_0, \psi_{1,k}) \), using

\[
d^1_k = \sum_n g_{n-2k} c^0_n. \tag{3.120}
\]

Also compute the scaling coefficients on this scale, \( c^1_k = (f_0, \phi_{1,k}) \), using

\[
c^1_k = \sum_n h_{n-2k} c^0_n. \tag{3.121}
\]

3. Now, proceed up through level \( J \) similarly,

\[
d^j_k = \sum_n g_{n-2^j k} c^{j-1}_n, \tag{3.122}
\]

\[
c^j_k = \sum_n h_{n-2^j k} c^{j-1}_n, \quad j = 1, 2, \ldots, J. \tag{3.123}
\]

The wavelet transform computations can be represented in matrix notation. The operation (3.123) can be represented as a matrix \( L \), where \( L_{ij} = h_{j-2i} \) for \( i \) and \( j \) in some suitable range. The operation (3.122) can be represented as a matrix \( H \), where \( H_{ij} = g_{j-2i} \).

**Example 3.18.4** We will demonstrate this matrix notation for a wavelet with four coefficients, \( h_0, h_1, h_2, h_3 \). We choose \( M \) so that \( \{g_0, g_1, g_2, g_3\} = \{h_3, -h_2, h_1, h_0\} \). Also, for the sake of a specific representation, we assume that \( \{c^0_n\} \) has six elements in it. From (3.121),

\[
e^1 = \begin{bmatrix} c^1_0 \\ c^1_1 \\ c^1_2 \\ c^1_3 \end{bmatrix} = \begin{bmatrix} h_2 & h_3 \\ h_0 & h_1 & h_2 & h_3 \\ h_0 & h_1 \\ h_0 & h_1 \end{bmatrix} \begin{bmatrix} c^0_0 \\ c^0_1 \\ c^0_2 \\ c^0_3 \end{bmatrix} = Le^0.
\]

(The truncation evident in the first and last rows of the matrix corresponds to an assumption that data outside the samples are equal to zero. As discussed below, there is another assumption that can be made.)

From (3.120),

\[
d^1 = \begin{bmatrix} d^1_0 \\ d^1_1 \\ d^1_2 \\ d^1_3 \\ d^2_0 \\ d^2_1 \end{bmatrix} = \begin{bmatrix} g_2 & g_3 \\ g_0 & g_1 & g_2 & g_3 \\ g_0 & g_1 \\ g_0 & g_1 \end{bmatrix} \begin{bmatrix} c^0_0 \\ c^0_1 \\ c^0_2 \\ c^0_3 \\ c^1_0 \\ c^1_1 \end{bmatrix} = He^0.
\]
The transform data at the next resolution $d^2$, and the data $e^2$, can be obtained (using the same indexing convention as before) as

$$e^2 = \begin{bmatrix} h_3 \\ h_1 \\ h_0 \\ h_1 \\ h_2 \end{bmatrix} c^1, \quad d^2 = \begin{bmatrix} g_3 \\ g_1 \\ g_2 \\ g_0 \\ g_1 \\ g_2 \end{bmatrix} c^1.$$

It is perhaps worthwhile to point out that the indexing convention on $c^1$ could be changed (with a corresponding change in (3.123)), so that we interpret $c^1$ as the vector

$$c^1 = \begin{bmatrix} c_0^1 \\ c_1^1 \\ c_2^1 \\ c_3^1 \end{bmatrix}.$$

Making this change, the matrix for the second stage transformation would be written as

$$e^2 = \begin{bmatrix} h_3 \\ h_1 \\ h_0 \\ h_1 \\ h_2 \end{bmatrix} \begin{bmatrix} c_0^1 \\ c_1^1 \\ c_2^1 \\ c_3^1 \end{bmatrix},$$

with similar changes for $d^2$ and its associated transformation matrix. Provided that the same indexing convention is used for the forward transformation as the inverse transformation, the transform is still fully reversible.

The notation $L$ and $H$ for the matrix operators is deliberately suggestive. The $L$ matrix is a lowpass operator, and the data sequence $c^1$ is a lowpass sequence. It corresponds to a “blurring” of the original data $c^0$. The $H$ matrix is a highpass operator, and the data $d^1$ is highpass (or bandpass) data.

The filtering/subsampling operation represented by these matrices can continue through several stages. The transform coefficients at the end of the process are the collection of data $d^1, d^2, \ldots, d^J$ and $e^J$, where $e^J$ is a final course approximation of the original starting data $c^0$. The wavelet transform computations can also be represented as a filtering/decimation operation, as shown in figure 3.24. The signal $c^0$ passes through a lowpass and highpass filter, whose outputs are decimated, as indicated by $\downarrow 2$, taking every other sample.

**Inverse wavelet transform**

The inverse wavelet transform can be obtained by working backwards. Given $d^j$ and $c^j$, we wish to find $c^{j-1}$. We note from (3.115) that

$$f_{j-1} = f_j + \delta_j = \sum_k c_k^j \phi_{j,k} + \sum_k d_k^j \psi_{j,k}. \quad (3.124)$$

Then, using the fact that $c^{j-1}_n = \langle f^{j-1}, \phi_{j-1,n} \rangle$ and taking inner-products of both sides of (3.124), we have

$$c^{j-1}_n = \langle f_{j-1}, \phi_{j-1,n} \rangle = \sum_k c_k^j \langle \phi_{j,k}, \phi_{j-1,k} \rangle + \sum_k d_k^j \langle \psi_{j,k} \phi_{j-1,k} \rangle. \quad (3.125)$$
Taking inner products on both sides of (3.119) with \( \phi_{j-1,m} \) we observe that
\[
\langle \phi_{j,k}, \phi_{j-1,m} \rangle = \sum_n h_{n-2k} \langle \phi_{j-1,n}, \phi_{j-1,m} \rangle = h_{m-2k}
\]
by the orthogonality of the \( \phi \) function. Similarly, from (3.116),
\[
\langle \psi_{j,k}, \phi_{j-1,m} \rangle = g_{m-2k}.
\]
Substituting these into (3.125), we find that
\[
c_{j-1}^{l-1} = \sum_k c_{j,k} h_{n-2k} + \sum_k d_{j,k} g_{n-2k}. \quad (3.126)
\]
This tells us how to go upstream from \( c^j \) and \( d^j \) to \( c^{j-1} \). The process is outlined in figure 3.25. As before, the reconstruction can be expressed in matrix form,
\[
c^{j-1} = L^*c^j + H^*d^j,
\]
where \( L^* \) is the adjoint (conjugate transpose) of \( L \) and \( H^* \) is the adjoint of \( H \) (see section 4.3).

**Example 3.18.5** Let us consider a specific numeric example. Using the wavelet with four coefficients, the code in algorithm 3.7 finds the two-scale wavelet transform data \( d^1, d^2, c^2 \) for the data set \( c = [1, 2, 3, 4, 5, 6]^T \). Also, the inverse transform is found. The pertinent variables of the execution are

\[
c_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \quad d_1 = \begin{bmatrix} -0.482963 \\ 7.62188e-09 \\ 2.28656e-08 \\ 3.38074 \\ -0.776457 \end{bmatrix}, \quad d_2 = \begin{bmatrix} -0.541266 \\ -0.670753 \\ 0.270032 \\ -0.375 \end{bmatrix}, \quad c_2 = \begin{bmatrix} -0.145032 \\ 0.557132 \\ 8.68838 \\ 1.39952 \end{bmatrix}.
\]

Observe that there are six points in the original data, and thirteen points in this transform. The reconstructed signal \( c_0 \) is equal to the original signal \( c_0 \).

**Algorithm 3.7** Demonstration of wavelet decomposition

File: wavetest.m

---

For comparison, algorithm 3.8 shows a decomposition and reconstruction with a different indexing convention. In this case, the transform data is

\[
d_1 = \begin{bmatrix} -0.12941 \\ 0 \\ 0 \\ -1.99191 \end{bmatrix}, \quad d_2 = \begin{bmatrix} -1.145037 \\ 0.195272 \\ 2.33133 \end{bmatrix}, \quad c_2 = \begin{bmatrix} -0.30681 \\ 2.10617 \\ 8.70064 \end{bmatrix}.
\]

The reconstructed signal \( c_0 \) is equal to the original signal. This transform has ten points in it.
Algorithm 3.8 Demonstration of wavelet decomposition (alternative indexing)

File: wavetest0.m

The $L$ and $H$ matrices have some interesting properties. In the following theorem, the $L$ and $H$ matrices are assumed to be infinite, so that partial sequences of coefficients do not appear on any rows.

**Theorem 3.8** The $L$ and $H$ operators defined by the operations

$$Lc = \sum_n h_{n-2k} c_n \quad Hc = \sum_n g_{n-2k} c_n$$

have the following properties:

1. $HL^H = 0$,
2. $LL^H = I$ and $HH^H = I$, and
3. $L^H L$ and $H^H H$ are mutually orthogonal projections.

**Proof** Let $h_{n-2k}$ denote the $k$th column of $L^H$, and let $g_{n-2l}$ denote the $l$th row of $H$. The inner product of these can be written

$$\sum_n h_{n-2k} g_{n-2l} = \sum_n h_{n-2k} (-1)^n h_{2(M+l)+1-n},$$

which is zero by (3.112). Since this is true for any $l$ and $k$ it follows that $HL^H = 0$.

The fact that $L^H L = I$ and $H^H H = I$ is shown by multiplication, using (3.108).

Then we note that $(L^H L + H^H H) L = L^H (L^H L + H^H H) L = L^H L$, so $L^H L$ is a projection, and similarly for $H^H H$. By the fact that $HL^H = 0$ it follows that $L^H L$ and $H^H H$ are orthogonal. Now note that

$$H(L^H L + H^H H) = H(H \ast H) = H$$

and

$$L(L^H L + H^H H) = L.$$

Thus $L^H L + H^H H$ acts as an identity on the ranges of both $H$ and $L$, so it is an identity.

The filtering interpretation for the reconstruction is shown in figure 3.26: the samples are expanded by inserting a zero between every sample, then filtering. When the forward operation and the backward operation are placed together, as shown in figure 3.27, an

![Figure 3.26: Filtering interpretation of an inverse wavelet transform](attachment://image.png)
identity operation from end to end results. One family of such filtering configurations is known as a quadrature mirror filter; it is an example of a perfect reconstruction filter. This multirate configuration is used in data compression, in which the lowpass and highpass signals are quantized using quantizers specialized for the frequency range of the signals.

**Periodic wavelet transform**

The wavelet transform produces more output coefficients than input coefficients, due to the convolution. If there are \( n \) input points and the filters are \( m \) points long, then the convolution/decimation operation produces \( \lfloor (n + m)/2 \rfloor \) points (or one less, depending on how the indexing is interpreted), so each stage of the transform produces more than half the number of points from the previous stage. Having more transform data than original data is troubling in many circumstances, such as data compression. It is common to assume that the data is periodic and to perform a periodized transform. Suppose that there are \( L \) points in \( c^0 \),

\[
    c^0 = [c_0^0, c_1^0, \ldots, c_{L-1}^0]^T.
\]

Then periodized \( c^0 \) is formed (conceptually) by stacking \( c^0 \),

\[
    c^0 = [\ldots, (c^0)^T, (c^0)^T, (c^0)^T, \ldots]^T.
\]

Then an \( L \)-point wavelet transform is computed on the periodized data. The effect is that the wavelet transform coefficients appear cyclically shifted around the \( L \) and \( H \) matrices. For example, with four coefficients and eight data points, the \( L \) and \( H \) matrices would look like the following:

\[
    L = \begin{bmatrix}
        h_0 & h_1 & h_2 & h_3 \\
        h_0 & h_1 & h_2 & h_3 \\
        h_2 & h_3 & h_0 & h_1
    \end{bmatrix},
\]

\[
    H = \begin{bmatrix}
        g_0 & g_1 & g_2 & g_3 \\
        g_0 & g_1 & g_2 & g_3 \\
        g_2 & g_3 & g_0 & g_1
    \end{bmatrix}.
\]

The same equations used to represent the nonperiodized transforms (3.122) and (3.123), and the inverse transform (3.126), also apply for the periodized transform and its inverse, provided that the indices are taken modulo the appropriate data size.

**Wavelet transform implementations**

Algorithm 3.9 performs a nonperiodic wavelet transform. The first function, \texttt{wavetrans}, sets up some data that is used by the recursively-called function \texttt{wave}. Implementation of \texttt{wave} is straightforward, with some caution needed to get the indexing started correctly. Since different levels have different lengths of coefficients, an array is also returned indexing the transform coefficients for each level.
Algorithm 3.9  Non-periodic wavelet transform  
File: wavetrans.m

Algorithm 3.10  Non-periodic inverse wavelet transform  
File: invwavetrans.m

Example 3.18.6  The two-level nonperiodic wavelet transform $c = [1, 2, 3, 4, 5]^T$ using the $D_4$ coefficients is computed using $[C, ap] = \text{wavetrans}(c, d4coeff, 2)$, which gives:

$$C = [-0.1294, 2.8978, -0.647, -1.145, 3.2688, -1.3068, -0.3068, 2.9297, 4.8771],$$

$$ap = [5, 1, 5, 8].$$

from which we interpret:

$$d_1 = [-0.1294, 0.0000, 2.8978, -0.6470],$$

$$d_2 = [-1.1450, 3.2688, -1.3068],$$

$$c_2 = [-0.3068, 2.9297, 4.8771].$$

The inverse transform computed by $\text{invwave}(C, ap, d4coeff)$ returns the original data vector.

Applications of wavelets

Wavelets have been used in a variety of applications, of which we mention only a few.

Data compression.  One of the most common applications of wavelets is to data compression. A set of data $f$ is transformed using a wavelet transform. The wavelet transform coefficients smaller than some prescribed threshold are set to zero, and the remaining coefficients are quantized using some uniform quantizer. It is a matter of empirical fact that in
most data sets, a large proportion of the coefficients are zeroed out. The truncated/quantized coefficients are then passed through a run-length encoder (and perhaps other lossless encoding techniques), which represents runs of zeros by a single digit indicating how many zeros are in the run.

A more sophisticated version of this algorithm is employed for image compression, in which a two-dimensional wavelet transform is employed. In this case, the hierarchical structure of the wavelet transform is exploited, so that if coefficients on one stage are small, there is a high probability that coefficients underneath are also small. Details of an algorithm of this sort are given in [305].

**Time/frequency analysis.** Wavelets are naturally employed in the analysis of signals which have a time-varying frequency content, such as speech or geophysical signals.

### 3.19 Signals as points; digital communications

The vector space viewpoint allows us to view signals, either in discrete or continuous time, as points in a vector space. This signals-as-points interpretation is especially useful in digital communications. In digital communications, a small set of basis functions is chosen—not a complete set—to have certain desired spectral properties. Signals that are transmitted are represented as linear combinations of these points.

As a particular example, let \( \phi_1(t) \) and \( \phi_2(t) \) be two orthonormal functions as illustrated in figure 3.28(a). (Note: the use of the notation \( \phi(t) \) as a basis function in this section is distinct from the notation for \( \phi(t) \) as a scaling function in section 3.18.4.) Then a variety of functions, such as those shown in figure 3.28(b), can be formed as linear combinations of \( \phi_1(t) \) and \( \phi_2(t) \):

\[
\begin{align*}
f_1(t) &= 2\sqrt{2}\phi_1(t) + 4\sqrt{2}\phi_2(t) \quad \Rightarrow \quad f_1 = (2\sqrt{2}, 4\sqrt{2}), \\
f_2(t) &= 3\sqrt{2}\phi_1(t) - 3\sqrt{2}\phi_2(t) \quad \Rightarrow \quad f_2 = (3\sqrt{2}, -3\sqrt{2}), \\
f_3(t) &= 3\sqrt{2}\phi_1(t) + 3\sqrt{2}\phi_2(t) \quad \Rightarrow \quad f_3 = (3\sqrt{2}, 3\sqrt{2}).
\end{align*}
\]

Figure 3.28(c) shows the points in \( \mathbb{R}^2 \) corresponding to the coordinates of the functions.

A function represented by a generalized Fourier series of \( m \) orthonormal functions

\[
f(t) = \sum_{i=1}^{m} c_i \phi_i(t)
\]

may be equivalently represented by the set of coordinates

\[
f(t) \longleftrightarrow (c_1, c_2, \ldots, c_m) = \mathbf{c}.
\]

and be conceptualized as a point in \( \mathbb{R}^m \). As shown in (3.96), the inner-product relationship between the functions is the same as the inner product between the vectors: if \( f_1(t) \) has the coordinate representation \( \mathbf{c}_1 \) and \( f_2(t) \) has the coordinate representation \( \mathbf{c}_2 \), then

\[
\langle f_1, f_2 \rangle = \langle \mathbf{c}_1, \mathbf{c}_2 \rangle,
\]

where the inner product on the left is defined for functions and the inner product on the right is defined for vectors. This means that

\[
\| f_1 \| = \| \mathbf{c}_1 \| \quad \text{and} \quad \| f_1 - f_2 \| = \| \mathbf{c}_1 - \mathbf{c}_2 \|.
\]
So the distance can be computed for either the function or the vector. Note that if the basis functions are not normalized, then $\|e_i\|$ in (3.127) must be normalized according to the norm of the basis functions.

Suppose we have $m$ orthogonal basis functions $\phi_i(t)$, $i = 1, 2, \ldots, m$, and assume that they have support over $[0, T)$. (It is not strictly necessary to deal with orthogonal basis functions, but it makes several of the computations easier. Of course, by the Gram–Schmidt orthogonalization procedure, we can always determine an orthonormal set spanning the same space as a set of nonorthogonal functions, so assuming orthonormality does not represent any loss of generality.)

In the $m$-dimensional space $S$ spanned by these functions, a set of $M = 2^k$ signal points, known as the signal constellation, is selected. Let $s_1, s_2, \ldots, s_M$ denote the signal constellation points, where the points are

$$s_i = [s_{i1}, s_{i2}, \ldots, s_{im}]^T.$$

These points in the signal constellation represent the signals that can be sent, $s_i(t)$, $i = 1, 2, \ldots, M$, where

$$s_i(t) = \sum_{i=1}^m s_{il} \phi_i(t) \in S.$$
The vector $s_i$ is sometimes referred to as the symbol, while the corresponding $s_i(t)$ is referred to as the signal. Normally (though not always), the basis functions $\phi_i(t)$ are designed to last $T$ seconds, and such that $\phi_i(t)$ has support over $[0, T]$. The time $T$ is called the symbol time.

Every $T$ seconds, $k$ bits are accepted into the transmitter. These $k$ bits are used to select one of $M = 2^k$ signal points, with its corresponding signal. The transmitted signal $s(t)$ is obtained by concatenating these signals together in time, which we can write as

$$s(t) = \sum_n s_n(t - nT),$$

where $s_n(t - nT)$ is the signal that starts at time $nT$ and has support over $t \in [nT, (n+1)T)$, and $l_n$ is the index of the signal selected at the $n$th symbol time. We will denote the signal that is transmitted at the $n$th symbol time as $s^n(t)$,

$$s^n(t) = s_{l_n}(t - nT).$$

In a practical system, it is customary to produce the signal $s(t)$ at baseband, then mix it up to some appropriate carrier frequency. In this presentation, we will focus only on the baseband signal $s(t)$. For additional simplicity, we will assume that all signals are real.

### 3.19.1 The detection problem

In a channel model that is commonly assumed, the signal $s(t)$ is delayed by some delay $\tau$ as it passes through the channel, and corrupted by additive noise $v(t)$. The received signal is modeled as

$$r(t) = s(t - \tau) + v(t).$$

Most of the intuitive discussion that follows in this section is accurate only in the case that the noise is Gaussian. We assume that the delay $\tau$ is known.

The signal $r(t)$ for $t \in [nT + \tau, (n+1)T + \tau)$ does not, in general, lie in $S$ because of the additive noise. The problem of reliable reception (the detection problem) is to determine the best estimate of the transmitted signal $s^n(t)$, given $r(t)$. A more formal exploration of this problem is conducted in chapter 11. However, for the purposes of this section we can employ our intuition about how the detection problem should work.

The first step in detection is to project the received signal over one symbol time onto $S$. The component of the $n$th received signal in the $l$th direction (assuming that $\tau$ is known) is

$$r_l = \langle r(t), \phi_l(t - \tau - nT) \rangle = \int_{\tau+nT}^{\tau+(n+1)T} r(t)\phi_l(t - \tau - nT) \, dt.$$

The processing accomplished by (3.128) is termed a correlator, illustrated in figure 3.29(a). It is also possible to implement the correlator by using a filter with impulse response $h_l(t) = \phi_l(T - t)$. In this case, the filter is termed a matched filter. The output of the filter is

$$y(t) = \int r(u)h_l(t-u) \, du = \int r(u)\phi_l(u-t+T) \, dt.$$

Sampling the output at the instant $t = \tau + (n+1)T$ produces the output value $r_l$ (see exercise 3.18-47). The coordinates $r' = [r_1, r_2, \ldots, r_m]^T$ represent the projection of the received signal onto $S$ for the $n$th symbol interval. The detector determines which of the signal points $s_1, s_2, \ldots, s_M$ is closest to $r$. The closest point corresponds to $s^n(t)$, from which it can be determined which bits were sent.
The projection onto the signal space is illustrated geometrically in Figure 3.30 for $m = 2$. The signal $r(t)$ is projected onto the signal-space point $r$. The nearest point in $S$ to $r$ is then determined as the estimate of the transmitted signal. The overall processing (using a matched filter implementation) is shown in Figure 3.31.

**Example 3.19.1** Let $\phi_1(t) = \sqrt{2/T} \cos(2\pi t)$ and $\phi_2(t) = \sqrt{2/T} \sin(2\pi t)$ for $t \in [0, T)$. These are orthonormal signals. Let the signal constellation be as in Figure 3.32. This type of constellation, in which every signal has the same amplitude but different components of phase (due to the combinations of the basis functions), is known as phase-shift keying (PSK).

In this signal constellation, suppose that the symbol $s_0$ is sent, and the projected received signal $\tilde{r}$ is as shown. The vector $r$ falls in the decision region of $s_0$, shown shaded in the figure. Thus $r$ is detected as the signal $s_0$.

Of course, it is possible that the noise is severe enough that a received signal is incorrectly detected, and so there is still a nonzero probability of error. Nevertheless, the
operation of projection and finding nearest neighbor is (for white Gaussian noise) the optimal decision rule. Some examination of the computation of probability of error takes place in chapter 11.

Another way of looking at the detection problem is to find the signal point \( s_i(t - \tau - nT) \) that is closest to \( r(t) \) for \( t \in [nT + \tau, (n + 1)T + \tau) \). That is, we wish to find \( s_i \) to minimize

\[
\int_{nT}^{T+(n+1)T} (r(t) - s_i(t - \tau - nT))^2 dt.
\]

Expanding this, we want to minimize

\[
\int_{nT}^{T+(n+1)T} r^2(t) dt - 2 \int_{nT}^{T+(n+1)T} r(t)s_i(t - \tau - nT) dt + \int_{nT}^{T+(n+1)T} s_i^2(t) dt.
\]

The first term does not depend upon \( s_i \), and the last term represents signal energy \( \|s_i\|^2 \), which can be precomputed. The decision statistic that we use is

\[
Z_i = \int_{nT}^{T+(n+1)T} r(t)s_i(t - \tau - nT) dt
\]

(3.129)

The processing in (3.129) can be done either by a correlator or a matched filter with impulse response \( h_i(t) = s_i(T - t) \), sampling the output of the filter at \( t = \tau + (n + 1)T \). The decision rule, in terms of this nearest signal interpretation, becomes: select the point \( s_i \) such that

\[-2Z_i + \|s_i\|^2 \]

is minimized.

One of the particularly interesting aspects about the vector space viewpoint for digital communications is that it allows different aspects of the problem to be addressed separately. The probability of error for a signal constellation depends ultimately on the geometry of the points in the signal constellation, and the average energy required to send the signals in comparison to the strength of the noise signal. The probability of error is thus completely unaffected by the particular waveforms underlying the signal constellation, provided only that orthonormal waveforms are selected. In contrast, the power spectral density of the transmitted signal depends very strongly on the waveform shapes of the signals transmitted. This separation of probability of error performance from spectral performance leads to better designs.

### 3.19.2 Examples of basis functions used in digital communications

A variety of waveforms can be used in digital communications. We met in example 3.19.1 the basis functions used for phase-shift keying. Here is a brief survey of some other simple signaling waveforms.

**On–off keying**, or OOK. When a single basis function \( \phi_1(t) \) is used (regardless of its waveform), with one point in the signal constellation at the origin and the other somewhere along the \( \phi_1(t) \) axis, a signaling technique known as on–off keying is produced. (See figure 3.33(a).)
Signals as Points; Digital Communications

Binary phase-shift keying, or BPSK. When a single basis function is used with two points in the signal constellation \( s_1 = -s_0 \), the resulting signaling is known as BPSK. (See figure 3.33(b).)

Pulse-position modulation, or PPM is obtained by using a set of \( N \) short orthogonal pulses, as shown in figure 3.33(c), which shows a four-dimensional set of basis functions. Usually, a single amplitude is employed along each orthogonal axis.

Frequency-shift keying, or FSK is obtained by using \( M \) sinusoidal signals of different frequencies which are spaced so that they are orthogonal over the interval \([0, T]\).

Quadrature-amplitude modulation, or QAM is obtained by using two orthogonal basis functions, as for PSK, but by employing both amplitude and phase modulation.

3.19.3 Detection in nonwhite noise

In the last section, the channel noise was assumed to be white and the optimal detector was obtained by simply projecting the received signal \( r(t) \) onto the signal space \( S \) with an orthogonal projection. When the noise is not white, however, the noise may tend to pull the received signal predilectably toward different spectral components. In this case, a more sophisticated filter must be used to obtain a projection onto the signal space to compensate for any bias introduced by the noise. The design of the filter provides yet another application of the Cauchy–Schwarz inequality.

We desire to find a filter with impulse response \( h(t) \), so that when \( r(t) \) is passed through the filter, the ratio of the signal power to the noise power is maximized at some particular sample time \( t_0 \), as shown in figure 3.34. When

\[
r(t) = s(t) + v(t),
\]

then the output of the filter is

\[
r(t) * h(t) = s(t) * h(t) + v(t) * h(t) = s_0(t) + n_0(t).
\]

Figure 3.34: Block diagram for detection processing
Signal power

Assuming a causal signal $s(t)$, the portion of the output due to the input signal is

$$s_o(t) = \int_{-\infty}^{t} s(\tau)h(t-\tau)\,d\tau.$$  

At the time instant $t_0$, we have

$$s_o(t_0) = \int_{0}^{t_0} s(\tau)h(t_0-\tau)\,d\tau.$$  

If we assume in addition that $s(t)$ is supported only over $[0, t_0)$ (so that we are using the entire signal $s$ to make our decision), then we can write

$$s_o(t_0) = \int_{-\infty}^{\infty} s(\tau)h(t_0-\tau)\,d\tau.$$  

Let $w(\tau) = \overline{h}(t_0-\tau)$. Then

$$s_o(t_0) = \int_{-\infty}^{\infty} s(\tau)\overline{w}(\tau)\,d\tau = \int_{-\infty}^{\infty} S(f)\overline{W}(f)\,df,$$  

where $S(f)$ and $W(f)$ are the Fourier transforms, respectively, of $s_o(t)$ and $w(t)$, and where the equality follows by Parseval’s theorem. Using the definition of $w$, we have

$$s_o(t_0) = \int_{-\infty}^{\infty} S(f)H(f)e^{j2\pi f t_0}\,df.$$  

The signal power $S$ at some time instant $t_0$ is $|s_o(t_0)|^2$, or

$$S = \left| \int_{-\infty}^{\infty} S(f)H(f)e^{j2\pi f t_0}\,df \right|^2. \quad (3.130)$$

Noise power

Let the PSD of $v(t)$ be $S_v(f)$. Then the PSD of the noise component at the output of the filter is

$$S_v(f)|H(f)|^2,$$  

and the total noise power is

$$N = \int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df. \quad (3.131)$$

The ratio of the signal to noise (SNR) power is, from (3.130) and (3.131),

$$\frac{S}{N} = \frac{\left| \int_{0}^{\infty} S(f)H(f)e^{j2\pi f t_0}\,df \right|^2}{\int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df}. \quad (3.132)$$

The problem can now be stated as: determine a filter with transfer function $H(f)$ that maximizes $S/N$ in (3.132). There is a tradeoff here: the wider the bandwidth of $H(f)$, the more signal gets through, but the more noise also gets through. This is a maximization problem that looks difficult, since the approach to maximization usually involves taking a derivative, and at this stage of development it is difficult to see what it would mean to take a derivative with respect to a transfer function. As we shall see, we will not have to take a derivative at all.
Maximizing $S/N$

The key to maximizing (3.132) is to use the Cauchy–Schwarz inequality in its integral form,

$$\left| \int x(f)\bar{y}(f)\,df \right|^2 \leq \int |x(f)|^2\,df \int |y(f)|^2\,df.$$  \hfill (3.133)

We can write

$$\frac{\int_{-\infty}^{\infty} S(f)H(f)e^{j2\pi f_0}\,df}{\int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df} = \frac{\int_{-\infty}^{\infty} (H(f)\sqrt{S_v(f)}) \left( \frac{S(f)}{\sqrt{S_v(f)}} \right) e^{j2\pi f_0}\,df}{\int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df} \leq \frac{\int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df \int_{-\infty}^{\infty} |S(f)|^2\,df}{\int_{-\infty}^{\infty} S_v(f)|H(f)|^2\,df} = \int_{-\infty}^{\infty} \frac{|S(f)|^2}{S_v(f)}\,df,$$  \hfill (3.134)

where (3.134) comes from the Cauchy–Schwarz inequality, using (by comparison with (3.133))

$$x(f) = \sqrt{S_v(f)}H(f) \quad \text{and} \quad \bar{y}(f) = \frac{S(f)}{\sqrt{S_v(f)}}e^{j2\pi f_0}.$$  

By this inequality, an upper bound on the SNR has been obtained which is independent of any filter and which, therefore, must be the largest possible regardless of the filter employed. The filter that can be used to achieve this upper bound with equality is found by employing the conditions under which the Cauchy–Schwarz inequality is satisfied with equality; in this case, that means that we must have $x(f) = Cy(f)$ for some nonzero complex constant $C$, or

$$H(f) = C\frac{S(f)}{S_v(f)}e^{-j2\pi f_0}$$

for any nonzero complex constant $C$.

If $v(t)$ is white, so that $S_v(f) = \frac{1}{2}N_0$, then we have

$$H(f) = C\frac{S(f)}{N_0/2}.$$  

Assume for ease of notation that $C = 2/N_0$. Then, taking the inverse transform, we have

$$h(t) = \bar{s}(t_0 - t).$$

The output of the filter with this impulse response when $t_0 = T$ is

$$\int_0^T r(t)\bar{s}(t)\,dt.$$  

3.20 Exercises

3.1-1 There is a connection between Grammians and linear independence, as demonstrated in theorem 3.1. We explore this connection further in this problem.

Let $\{p_1, p_2, \ldots, p_k\}$ be a set of vectors, and let us suppose that the first $k - 1$ vectors of this set have passed a test for linear independence. We form

$$e_k = c_{k-1}^p p_1 + c_{k-2}^p p_2 + \cdots + c_1^p p_{k-1} + p_k$$

and want to know if $e_k$ is equal to zero for any set of coefficients

$$c = [c_{k-1}, c_{k-2}, \ldots, c_1, 1].$$
If so, then \( p_k \) is linearly dependent. Let 
\[ A_k = [p_1, p_2, \ldots, p_k] \]
be a data matrix, and let \( R_k = A_k^H A_k \) be the corresponding Grammian.

(a) Show that the squared error can be written as
\[ e_k^H e_k = \sigma_k^2 = e_k^H \begin{bmatrix} R_{k-1} & h_k \\ h_k^H & r_{kk} \end{bmatrix} c \]
for some \( h_k \) and \( r_{kk} \). Identify \( h_k \) and \( r_{kk} \).

(b) Determine the minimum value of \( \sigma_k^2 \) by minimizing (3.135) with respect to \( c_k \), subject to the constraint that the last element of \( c_k \) is equal to 1. Hint: take the gradient of
\[ e_k^H \begin{bmatrix} R_{k-1} & h_k \\ h_k^H & r_{kk} \end{bmatrix} c - \lambda (c_k^H d - 1), \]
where \( \lambda \) is a Lagrange multiplier and \( d = [0, 0, \ldots, 0, 1]^T \). Show that we can write the corresponding equations as
\[ \begin{bmatrix} R_{k-1} & h_k \\ h_k^H & r_{kk} \end{bmatrix} c = \sigma_k^2 d. \]  (3.136)

(c) Show that (3.136) can be manipulated to become
\[ \sigma_k^2 = r_{kk} - h_k^H R^{-1}_{k-1} h_k. \]
The quantity \( \sigma_k^2 \) is called the Schur complement of \( R_k \). If \( \sigma_k^2 = 0 \), then \( p_k \) is linearly dependent.

3.5-2 Referring to (3.30), show that
\[ (I - A(A^H A)^{-1} A^H) \]
is positive semidefinite, and hence that the minimum error \( e_{\text{min}} \) has smaller norm than the original vector \( x \). Hint: consider \( 0 \leq \|Bx\|^2 \), where \( B = I - A(A^H A)^{-1} A^H \).

3.8-3 Consider the set of data
\[ x = \{2, 2.5, 3, 5, 9\} \quad y = \{-4.2, -5, 2, 1, 24.3\}. \]

(a) Make a plot of the data.

(b) Determine the best least-squares line that fits this data and plot the line.

(c) Assuming that the first and last points are believed to be the most accurate, formulate a weighting matrix and compute a weighted least-squares line that fits the data. Plot this line.

3.8-4 Formulate the regression problem (3.34) in a linear form as in (3.37).

3.8-5 Formulate the regression problem (3.35) in a linear form as in (3.37).

3.8-6 Formulate the regression \( y \approx c e^{ax} \) as a linear regression problem, with regression parameters \( c \) and \( a \).

3.8-7 Formulate the regression \( y \approx ax^k \) as a linear regression problem.

3.8-8 Perform the computations to verify the slope and intercept of the linear regression in (3.39).
3.8-9 As a measure of fit in a correlation problem, the correlation coefficient, analogous to (1.49), can be obtained as

$$\rho = \frac{(\mathbf{x}, \mathbf{y}) - (\mathbf{x}, 1)(\mathbf{y}, 1)}{(\|\mathbf{x}\| - (\mathbf{x}, 1))(\|\mathbf{y}\| - (\mathbf{y}, 1))}. $$

The correlation coefficient $\rho = \pm 1$ if $x$ and $y$ are exactly functionally related, and $\rho = 0$ if they are independent. For the linear regression in (3.38), determine an explicit expression for $\rho$.

3.8-10 Define an inner product between matrices $X$ and $Y$ as

$$\langle X, Y \rangle = \text{tr}(XY^H),$$

where $\text{tr}(\cdot)$ is the sum of the diagonal elements (see section C.3). We want to approximate the matrix $Y$ by the scalar linear combination of matrices $X_1, X_2, \ldots, X_m$, as

$$Y = c_1X_1 + c_2X_2 + \cdots + c_mX_m + E.$$

Using the orthogonality principle, determine a set of normal equations that can be used to find $c_1, c_2, \ldots, c_m$ that minimize the induced norm of $E$.

3.8-11 For the ARMA input/output relationship of (1.2), determine a set of linear equations for determining the ARMA model parameters $\{a_1, a_2, \ldots, a_p, b_0, b_1, \ldots, b_q\}$, assuming that the model or $(p, q)$ is known, and that the input is known.

3.9-12 For the data sequence $\{1, 1, 2, 3, 5, 8, 13\}$:

(a) Write down the data matrix $A$ and the Grammian $A^H A$ using (i) the covariance, and (ii) the autocorrelation methods. Assume $m = 2$.

(b) We desire to use this sequence to train a simple linear predictor. The “desired signal” $d[t]$ is the value of $x[t-1]$, and the data used are the two prior samples. That is,

$$x[t] = a_1x[t-1] + a_2x[t-2] + e[t],$$

where $e[t]$ is the prediction error. Determine the least-squares coefficients for the predictor using the covariance and autocorrelation methods.

(c) Determine the minimum least-squares error for both methods.

3.10-13 Consider a data sequence $[x[t]]$, the correlation matrix $R$ is

$$R = \begin{bmatrix} 5 & .3 \\ .3 & .5 \end{bmatrix}$$

and the cross-correlation vector $p$ with a desired signal is

$$p = \begin{bmatrix} .2 \\ .5 \end{bmatrix}.$$ 

Determine the optimal weight vector.

3.10-14 Consider a zero-mean random vector $\mathbf{x} = [x_1, x_2, x_3]$ with covariance

$$\text{cov}(\mathbf{x}) = E[\mathbf{x}\mathbf{x}^T] = \begin{bmatrix} 1 & .7 & .5 \\ .7 & 4 & 2 \\ .5 & 2 & 3 \end{bmatrix}.$$ 

(a) Determine the optimal coefficients of the predictor of $x_1$ in terms of $x_2$ and $x_3$.

$$\hat{x}_1 = c_1x_2 + c_2x_3.$$
(b) Determine the minimum mean-squared error.
(c) How is this estimator modified if the mean of \( x \) is \( E[x] = [1, 2, 3]^T \)?

3.10-15 [132] A discrete-time radar signal is transmitted as
\[
s(t) = A_0 e^{-j\omega_0 t}\]
The sampled noisy received signals are represented as
\[
x(t) = A_1 e^{-j\omega t} + v(t),
\]
where \( \omega_0 \) is the received signal frequency, in general different from \( \omega_0 \) because of Doppler shift, and \( v(t) \) is a white-noise signal with variance \( \sigma_v^2 \). Let
\[
x(t) = [x[0], x[1], \ldots, x[m-1]]^T
\]
be a vector of received signal samples.
(a) Show that
\[
R = E[x(t)x^H(t)] = \sigma_v^2 I + \sigma_x s(\omega_0) s^H(\omega_0),
\]
where
\[
s(\omega_0) = [1, e^{-j\omega_0}, e^{-j2\omega_0}, \ldots, e^{-j(m-1)\omega_0}]^T \quad \text{and} \quad \sigma_v^2 = E[|A_1|^2].
\]
(b) The time series \( x(t) \) is applied to an FIR Wiener filter with \( m \) coefficients, in which the cross-correlation between \( x(t) \) and the desired signal \( d[t] \) is preset to
\[
p = s(\omega_0).
\]
Determine an expression for the tap-weight vector of the Wiener filter.

3.11-16 A channel with transfer function
\[
H_c(z) = \frac{1}{1 - .2 z^{-1}}
\]
and output \( u(t) \) is driven by an AR(1) signal \( d[t] \) generated by
\[
d(t) - .4d(t-1) = v(t),
\]
where \( v(t) \) is a zero-mean white-noise signal with \( \sigma_v^2 = 2 \). The channel output is corrupted by noise \( n[t] \) with variance \( \sigma_n^2 = 1.5 \), to produce the signal
\[
f(t) = u(t) + n(t).
\]
Design a second-order Wiener equalizer to minimize the average squared error between \( f[t] \) and \( d[t] \). What is the MSE?

3.11-17 Linear prediction A common application of Wiener filtering is in the context of linear prediction. Let \( d[t] = x[t] \) be the desired value, and let
\[
x(t) = \sum_{i=1}^{m} w_{i,j} x(t-i)
\]
be the predicted value of \( x[t] \) using an \( m \)th order predictor based upon the measurements \( \{x[t-1], x[t-2], \ldots, x[t-m]\} \), and let
\[
f_n[t] = x[t] - \hat{x}[t]
\]
be the forward prediction error. Then
\[
f_n[t] = \sum_{i=0}^{m} a_{i,j} x(t-i),
\]
where \( a_{f,0} = 1 \) and \( a_{f,i} = -w_{f,i}, i = 1, 2, \ldots, M \).

Assume that \( x[t] \) is a zero-mean random sequence. We desire to determine the optimal set of coefficients \( \{w_{j,i}, i = 1, 2, \ldots, M\} \) to minimize \( E[f_M[t]^2] \).

(a) Using the orthogonality principle, write down the normal equations corresponding to this minimization problem. Use the notation \( r[j-l] = E[x[t-l]x[t-j]] \) to obtain the Wiener–Hopf equation

\[
Rw_f = r,
\]

where \( R = E[x[t-1]x^H[t-1]], r = E[x[t-1]x[t]], \) and \( x[t-1] = [x[t-1], x[t-2], \ldots, x[t-m]]^T \).

(b) Determine an expression for the minimum mean-squared error, \( P_m = \min E[f_M[t]^2] \).

(c) Show that the equations for the optimal weights and the minimum mean-squared error can be combined into augmented Wiener–Hopf equations, as

\[
\begin{bmatrix} r[0] & r^H \\ r & R \end{bmatrix} \begin{bmatrix} 1 \\ -w_f \end{bmatrix} = \begin{bmatrix} P_m \\ 0 \end{bmatrix}
\]

(d) Suppose that \( x[t] \) happens to be an AR\((m)\) process driven by white noise \( v[t] \), such that it is the output of a system with transfer function

\[
H(z) = \frac{1}{1 + \sum_{k=1}^{m} a_k z^{-k}}.
\]

Show that the prediction coefficients are \( w_{f,k} = -a_k \), and hence the coefficients of the prediction error filter \( f_n[t] \) are

\[
a_{f,i} = a_i.
\]

(Hint: see section 1.4.2; write down the Yule-Walker equations.) Hence, conclude that in this case the forward prediction error \( f_n[t] \) is a white-noise sequence. The prediction-error filter can thus be viewed as a whitening filter for the signal \( x[t] \).

(e) Now let

\[
\hat{x}[t-m] = \sum_{i=1}^{m} w_{b,i} x[t-i+1]
\]

be the backward predictor of \( x[t-m] \) using the data \( x[t-m+1], x[t-m+2], \ldots, x[t], \) and let

\[
b_n[t] = x[t-m] - \hat{x}[t-m]
\]

be the backward prediction error. A backward predictor seems strange—after all, why predict what we should have already seen—but the concept will have useful applications in fast algorithms for inverting the autocorrelation matrix. Show that the Wiener–Hopf equations for the optimal backward predictor can be written as

\[
Rw_b = r^B,
\]

where \( r^B \) is the backward ordering of \( r \) defined above.

(f) From (3.137), show that

\[
R^H \overline{w_b^B} = r,
\]

where \( w_b^B \) is the backward ordering of \( w_b \). Hence, conclude that

\[
\overline{w_b^B} = w_f;
\]

that is, the optimal backward prediction coefficients are the reversed conjugated optimal forward prediction coefficients.
3.11-18 Let

\[ x[r] = 0.8x[r-1] + v[r], \]

where \( v[r] \) is a white-noise zero-mean, unit-variance noise process. We want to determine an optimal predictor.

(a) If the order of the predictor is 2, determine the optimal predictor \( \hat{x}[r] \).
(b) If the order of the predictor is 1, determine the optimal predictor \( \hat{x}[r] \).

3.11-19 Random vectors The mean-squared methods to this point have been for random scalars. Suppose we have the random vector approximation problem

\[ y = c_1p_1 + c_2p_2 + \cdots + c_mp_m + e, \]

in which we desire to find an approximation \( y \) in such a way that the norm of \( e \) is minimized. Let us define an inner product between random vectors as

\[ \langle x, y \rangle = \text{tr}(E[x'y'^H]). \]

(a) Based upon this inner product and its induced norm, determine a set of normal equations for finding \( c_1, c_2, \ldots, c_m \).
(b) As an exercise in computing gradients, use the formula for the gradient of the trace (see appendix E) to arrive at the same set of normal equations.

3.11-20 Multiple gain-scaled vector quantization Let \( X \) and \( Y \) be vector spaces of the same dimensionality. Suppose that there are two sets of vectors \( X_1, X_2 \subseteq X \). Let \( Y \) be the set of vectors pooled from \( X_1 \) and \( X_2 \) by the invertible matrices \( T_1 \) and \( T_2 \), respectively. That is, if \( x \in X_i \), then \( y = T_i x \) is a vector in \( Y \). Indicate that a vector \( y \in Y \) came from a vector in \( X_i \) by a superscript \( i \), so \( y' \in Y \) means that there is a vector \( x \in X_i \) such that \( y' = T_i x \). Distances relative to a vector \( y' \in Y \) are based upon the \( l_2 \) norm of the vectors obtained by mapping back to \( X_i \), so that

\[ d(y', y) = \|y' - y\| = \|T_i^{-1}(y' - y)\| = (y' - y)^T W_i (y' - y), \]

where \( W_i = T_i^{-T} T_i^{-1} \). This is a weighted norm, with the weighting dependent upon the vector in \( Y \). (Note: in this problem \( \| \cdot \|_1 \) and \( \| \cdot \|_2 \) refer to the weighted norm for each data set, not the \( l_1 \) and \( l_2 \) norms, respectively.)

We desire to find a single vector \( y_0 \in Y \) that is the best representation of the data pooled from both data sets, in the sense that

\[ \sum_{y \in Y} \| y - y_0 \| = \sum_{y' \in Y} \| y' - y_0 \| + \sum_{y' \in Y} \| y' - y_0 \|_2 \]

is minimized. Show that

\[ y_0 = Z^{-1}r, \]

where

\[ Z = \sum_{y' \in Y} W_1 + \sum_{y' \in Y} W_2 \quad \text{and} \quad r = \sum_{y' \in Y} W_1 y'_1 + \sum_{y' \in Y} W_2 y'_2. \]

Hint: this is probably easier using gradients than trying to identify the appropriate inner product.
3.12-21 Assume the estimated autocorrelation

\[ \hat{r}[n] = \frac{1}{N} \sum_{k=1}^{N} x(k)\overline{x}(k - n). \]

(a) Take the expectation \( E[\hat{r}[n]] \) and show that it is not equal to \( r[n] \), the true value of the autocorrelation.
(b) Determine a scaling factor to make the \( \hat{r}[n] \) an unbiased estimate.
(c) Write a MATLAB function that computes \( \hat{r}[n] \) from (3.65).

3.13-22 Let \( x, y, \text{ and } u \) be continuous-time random processes, with \( y = x + u \), and \( S_y(s) = 1 \). Determine an optimal causal filter \( h(t) \) to determine \( x(t) \) when:
(a) The PSD of \( x(t) \) is

\[ S_x(s) = \frac{s^2 - 16}{s^4 - 53s^2 + 196}. \]

(b) The PSD of \( x(t) \) is

\[ S_x(s) = \frac{s^4 - 10s^2 + 9}{s^4 - 53s^2 + 196}. \]

3.13-23 (Spectral factorization; the Fejér–Riesz theorem) Because of the importance of the canonical factorization in signal processing, it is of interest to determine when a “square root” of a function exists. In this problem you will prove the following: If \( W(z) = \sum_{n=0}^{\infty} w[n]z^{-n} \) is real and \( W(e^{j\omega}) \geq 0 \) for all \( \omega \), then there is a function

\[ Y(z) = \sum_{n=0}^{m} y[n]z^{-n} \]

such that \( W(e^{j\omega}) = |Y(e^{j\omega})|^2 \).
(a) Show that \( w[n] = \overline{w}[n] \).
(b) Show that \( W(z) = W(1/\overline{z}) \).
(c) Show that if \( z_i \) is a root of \( W(z) \), then \( 1/\overline{z}_i \) is a root of \( W(z) \).
(d) Argue that if \( z_i = e^{j\phi_i} \) is a root on the unit circle, then it must have even multiplicity.
(e) Let \( Z = \{z_i : W(z_i) = 0; |z_i| \leq 1, (\text{only half the roots on } |z| = 1) \} \) be the set of roots inside, and half those on, the unit circle. Then \( Z \) has \( m \) elements and

\[ W(z) = Az^{-m} \prod_{i=1}^{m} (z - z_i) \prod_{i=1}^{m} (z\overline{z}_i - 1). \]

From this form, find \( Y(z) \).

3.13-24 Filtering in White Noise Let \( x, y \text{ and } u \) be discrete-time random processes with
\( y = x + u \)
and
\( S_x(z) = 1 \).
\( S_u(z) = \frac{b(z)}{a(z)} \),

where \( b(z) \text{ and } a(z) \) are polynomials in \( z \) with the degree of \( b(z) \) strictly lower than the degree of \( a(z) \). Furthermore, assume \( R_{xx}(t) \equiv 0 \). Show that (3.82) holds in the discrete-time...
3.13-25 Let
\[ y_i = x_i + v_i \]
where
\[ R_x(t) = \begin{cases} \frac{2}{3} \delta_t, & t = 0 \\ \frac{10}{27} \left( \frac{1}{2} \right)^{|t|}, & t \neq 0 \end{cases} \]
with \( E x_i = E v_i = E x_i v_i = 0 \). Show that
(a) \( S_y(z) = \frac{(1 - z^{-1})}{(1 - e^{-\delta} z^{-1})^2} \), and thus obtain \( S_y^+(z) \) and \( S_y^-(z) \).
(b) \[ \left\{ \frac{S_y^+(z)}{S_y(z)} \right\} = \frac{1}{z^2} \]
and, thus, that the Wiener filter is
\[ H(z) = \frac{1}{1 - \frac{1}{2} z} \].

3.13-26 Let \( x_i, y_i, \) and \( v_i \) be discrete-time random processes with \( y_i = x_i + v_i \), \( S_x(z) = 1 \), and
\[ S_y(z) = \frac{z^4 - 9.0067z^3 + 28.04z^2 - 9.0067z + 1}{z^4 - 2.0111z^3 + 3.0446z^2 - 2.0111z + 1} \]
Determine the filter \( h[z] \) to optimally predict \( x_{i+2} \).

3.14-27 Let \( h(t) \) be the impulse response of a system, and let \( y(t) = x(t) * h(t) \). Show that
\[ \int_0^T y(t) \, dt = x(t) * k(t) \bigg|_{t=T}, \]
where \( k(t) \) is the integral of the impulse response,
\[ k(t) = \int_0^t h(\tau) \, d\tau. \]

3.14-28 A system is known to have impulse response \( h(t) = 3e^{-2t} + 4e^{-3t} \), and is initially relaxed (initial conditions are zero). Determine an input \( x(t) \) so that the output satisfies the conditions
\[ y(2) = 2 \quad \text{and} \quad \int_0^2 y(t) \, dt = 3, \]
in such a way that the input energy \( \| x(t) \|^2 \) is minimized. Plot \( x(t) \).

3.14-29 Let \( h[t] = (0.2)^t + 3(0.4)^t \) for \( t \geq 0 \) be the impulse response of a discrete-time system with zero initial conditions. It is desired to determine a causal input sequence \( x[t] \), such that the output \( y[t] = h[t] * f[t] \) satisfies the constraints
\[ y[10] = 5, \quad \sum_{j=0}^{10} y[j] = 2, \]
and such that the input energy \( \sum_{t=0}^{10} |x[t]|^2 \) is minimized. Formulate this as a dual approximation problem and find the minimizing sequence \( x[t] \).
3.15-30 [209] Using the projection theorem, solve the finite dimensional problem

\[
\text{minimize } x^\top Q x \\
\text{subject to } Ax = b,
\]

where \( x \in \mathbb{C}^n \), \( Q \) is a positive-definite symmetric matrix, and \( A \) is an \( m \times n \) matrix with \( m < n \).

3.15-31 [209] Let \( x \) be a vector in a Hilbert space \( S \) and let \( \{x_1, x_2, \ldots, x_m\} \) and \( \{y_1, y_2, \ldots, y_n\} \) be sets of linearly independent vectors in \( S \). We desire to minimize \( \|x - \hat{x}\| \), while satisfying

\( \hat{x} \in M = \text{span}(\{x_1, x_2, \ldots, x_m\}) \)

and \( (\hat{x}, y_i) = c_i, i = 1, 2, \ldots, n \). Find equations for the solution which are similar to the normal equations.

3.17-32 Show that the functions defined by

\[ p_k(t) = \frac{1}{\sqrt{N}} e^{i2\pi k t/N} \]

are orthonormal with respect to the inner product

\[ (x[t], y[t]) = \sum_{r=0}^{N-1} x[r] y[r]. \]

3.17-33 Let \( g(t) = e^{-t^2/2} \) for \( 0 \leq t \leq \pi \), and let \( f(t) \) be the \( \pi \)-periodic extension of \( g(t) \).

\[ f(t) = \sum_k g(t - k\pi). \]

(a) Find the Fourier series coefficients of \( f(t) \).

(b) Find the sum of the series

\[ \sum_n \left( \frac{2^\alpha}{\pi^2} \frac{1}{1 + 16n^2} \right). \]

Hint: Use Parseval’s theorem.

3.18-34 Show that the definition of Chebyshev polynomials (3.101) satisfies the recurrence in (3.100) for \( |t| < 1 \). Show for \( |t| > 1 \) that \( T_n(t) = \cosh(n \cosh^{-1} t) \) satisfies the recursion (3.100).

3.18-35 The Christoffel-Darboux formula

(a) Using (3.98), show that the polynomials \( p_k(t) \), orthonormal with respect to the inner product \( (f, g)_w = \int_a^b f(t) g(t) w(t) \, dt \), satisfy

\[ \int_a^b t p_n(t) p_{n+1}(t) w(t) \, dt = a_n. \]

Also show that

\[ c_n = a_{n-1}. \]

(b) Consider the partial sum

\[ S_n(t) = \sum_{k=0}^n (f, p_k)_w p_k(t). \]

Show that the sum can be written as

\[ S_n(t) = \int_a^b f(y) K_n(t, y) w(y) \, dy. \]
where

\[ K_n(x, y) = \frac{a_n(p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y))}{x - y} \]

and where \( a_n \) comes from (3.98). This formula for \( K_n(x, y) \) is known as the Christoffel-Darboux formula, and is analogous to the Dirichlet kernel of Fourier series. Hint: form \((x - y)K_n(x, y)\) and use the results from part (a).

3.18-36 Show that each of the polynomials produced by orthogonalizing \([1, x, x^2, \ldots]\) using the Gram-Schmidt procedure over the interval \([a, b]\) has zeros which are real, simple, and located in \((a, b)\).

3.18-37 In this exercise we introduce the idea of Gaussian quadrature, a fast and important method of numerical integration. The idea is to approximate the integral as a summation:

\[ \int_a^b f(t) \, dt \approx \sum_{i=1}^m a_i f(t_i). \]

Unlike many conventional numerical integration formulas, in Gaussian quadrature the abscissas are not evenly spaced. The problem is to find the \([t_i]\) (abscissas) and \([a_i]\) (weights) so that the integral is as accurate as possible. In the Gaussian quadrature method of numerical integration, for polynomials up to degree \(2m - 1\) the result of the integration is exact. For sufficiently smooth non-polynomial functions the method is often very accurate. The solution makes significant use of orthogonal polynomials. For the purposes of this exercise, we will assume the inner product \((f, g) = \int_{-1}^1 f(t)g(t) \, dt\).

(a) As this first part shows, without loss of generality, we may restrict attention to the interval \(a = -1, b = 1\). Show that for the integral

\[ \int_{-1}^1 g(x) \, dx \]

the substitution

\[ t = \frac{1}{b - a} (2x - a - b) \]

leads to an integral of the form

\[ \int_{-1}^1 f(t) \, dt. \]

(Hence the limits of \(a\) and \(b\) can be converted to limits of \(-1\) to \(1\).)

(b) If \([p_n(t), n = 0, 1, \ldots, m]\) is a set of polynomials orthogonal over \([-1, 1]\), where \(p_n(t)\) is a polynomial of degree \(n\), show that

\[ (p(t), p_n(t)) = 0 \]

for all polynomials \(p(t)\) of degree \(\leq m - 1\).

(c) Let \(f(t)\) be a polynomial of degree \(2m - 1\). Show that \(f(t)\) can be written as

\[ f(t) = q(t)p_m(t) + r(t), \]

where \(q(t)\) and \(r(t)\) are of degree \(\leq m - 1\). Hint: divide.

(d) Show that there are series expansions

\[ q(t) = \sum_{k=0}^{m-1} \alpha_k p_k(t) \quad \text{and} \quad r(t) = \sum_{k=m}^{m-1} \beta_k p_k(t). \]
(e) Show that
\[ \int_{-1}^{1} f(t) \, dt = \beta_0 \int_{-1}^{1} p_0(t) \, dt. \] (3.138)

(f) Let \( t_1, t_2, \ldots, t_m \) be the roots of \( p_m(t) \). Show that
\[ \sum_{i=1}^{m} a_i f(t_i) = \sum_{k=0}^{m-1} \beta_k \sum_{i=1}^{m} a_i p_k(t_i). \] (3.139)

(g) Show that if the weights \( a_i \) are chosen so that
\[ \sum_{i=1}^{m} a_i p_k(x_i) = \begin{cases} \int_{-1}^{1} p_0(t) \, dt & k = 0, \\ 0 & k = 1, 2, \ldots, n - 1, \end{cases} \]
then (3.139) can be written as
\[ \sum_{i=1}^{m} a_i f(t_i) = \beta_0 \int_{-1}^{1} p_0(t) \, dt. \] (3.140)

(h) Write a matrix equation for the weights \( \{a_i\} \), chosen in part (g).

(i) Hence, equating (3.138) and (3.140), write down the formula for Gaussian quadrature.

(j) Generalize this to finding \( \int_{-1}^{1} w(t) f(t) \, dt \), where the polynomials \( p_k(t) \) are orthogonal with respect to the inner product \( (f, g) = \int_{-1}^{1} f(t) g(t) w(t) \, dt \).

3.18-38 Prove Parseval's theorem for Fourier transforms: If \( y_1(t) \leftrightarrow Y_1(\omega) \) and \( y_2(t) \leftrightarrow Y_2(\omega) \), then
\[ \int_{-\infty}^{\infty} y_1(t) \overline{y_2(t)} \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y_1(\omega) \overline{Y_2}(\omega) \, d\omega. \]

3.18-39 Sampling theorem representations.

(a) Show for \( p_k(t) \) defined as in (3.102) that \( (p_k, p_l) = \frac{1}{2B} \delta_{k,l} \). Along the way, show that
\[ \int_{-\infty}^{\infty} \sin t \sin(t - z) \, dt = \frac{\pi \sin z}{z}. \]

Hint: use Parseval's theorem and Fourier transforms.

(b) Show that (3.103) is correct for a bandlimited function \( f(t) \).

(c) Show that if \( f(t) \) is bandlimited to \( B \) Hz,
\[ f(z) = 2B \int_{-\infty}^{\infty} f(t) p_0(t - z) \, dt, \]

Thus, for bandlimited functions, \( p_0(t) \) behaves like a \( \delta \) function.

3.18-40 Show that if \( \phi(t) \) is normalized then \( 2^{-1/2} \phi(2^{-1/2} t) \) is normalized.

3.18-41 In (3.106), show that the coefficients \( c_n \) must satisfy
\[ \sum_{n} c_n = 2. \]

3.18-42 Show that there is no orthogonal scaling function defined by a two-scale equation (3.106) with exactly three nonzero coefficients \( c_0, c_1, \) and \( c_2 \).

3.18-43 For the multiresolution analysis:

(a) Show that \( W_j \perp W_{j'} \).
(b) Show that for \( j < J \),
\[
V_j = V_j \oplus \bigoplus_{k=0}^{J-j-1} W_{j-k}.
\]

3.18-44 Show that if \( \phi(t) \) obeys the two-scale relationship in (3.105), and if \( \hat{\phi}(\omega) \) represents the Fourier transform of \( \phi(t) \), then
\[
\hat{\phi}(\omega) = m_0(\omega/2)\hat{\phi}(\omega/2).
\]
where
\[
m_0(\omega) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-j\omega n} \tag{3.141}
\]
is the scaled discrete-time Fourier transform of the coefficient sequence.

3.18-45 **Decimation** Because of the connection of wavelet transforms to multirate signaling, it is worthwhile to examine the transform of decimated signals. You will show that if \( y[n] \) is a decimation of \( x[n] \),
\[
y[n] = x[nD],
\]
then
\[
Y(z) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{-j2\pi{k}/D} z^{1/D}). \tag{3.142}
\]

(a) Let \( p[n] \) be the periodic sampling sequence
\[
p[n] = \begin{cases} 1 & n = 0, \pm D, \pm 2D, \ldots \\ 0 & \text{otherwise} \end{cases}
\]
Show that
\[
p[n] = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi{k}n/D}.
\]

(b) Let \( z[n] = x[n]p[n] \). Then \( y[n] = z[nD] \). Show that
\[
Y(z) = \sum_m y[m]z^{-m} = \sum_m z[nD]z^{-m} = \sum_n x[n]p[n]z^{-n/D}.
\]

(c) Finally, show that (3.142) is true.

3.18-46 Show that the orthogonality condition (3.108) is equivalent to
\[
|m_0(\omega/2)|^2 + |m_0(\omega/2 + \pi)|^2 = 1.
\]

Hint: recognize that (3.108) is a decimated convolution, and use the fact that if the Fourier transform of a sequence \( z_n \) is \( Z(\omega) \), then the Fourier transform of \( z_{2n} \) is
\[
\frac{1}{2} [Z(\omega/2) + Z(\omega/2 + \pi)].
\]

3.19-47 Let \( \phi(t) \) be a one-dimensional basis function for digital transmission, of the form
\[
\phi(t) = u(t) - u(t-1)
\]
1.20 Exercises 227

(a unit pulse). Assume that \( s(t) = \phi(t) \) is transmitted. Let \( r(t) = s(t) \) (noise-free reception).
Show the output of the correlator
\[
y_1(t) = \int_{0}^{t} r(s) \phi(s) \, ds
\]
and the output of the matched filter with impulse response \( h(t) = \phi(T - t) \).
\[
y_2(t) = r(t) * h(t).
\]
Show that at the sample instant \( t = T \), \( y_1(t) = y_2(t) \).

3.19-48 Let
\[
\phi_m(t) = \begin{cases} 
\cos\left(2\pi f_c + 2\pi m \Delta f \cdot t\right) & 0 \leq t \leq T, \\
0 & \text{otherwise}
\end{cases}
\]
for \( m = 0, 1, \ldots, M - 1 \) be a set of basis functions. Determine the minimum frequency separation \( \Delta f \) such that
\[
\int_{0}^{T} \phi_m(t) \phi_k(t) \, dt = 0
\]
for \( k \neq m \). Assume that \( f_c T = n \) for some integer \( n \). (Digital transmission with such signals is called frequency-shift keying.)

3.19-49 (Spread-spectrum multiple access) In this exercise, we examine matched filters for a more complicated scenario: spread spectrum multiple access. In this model, \( K \) users are transmitting simultaneously, with the \( k \)-th user transmitting a signal
\[
s_k(t) = \sum_{n} b_k(n) \sqrt{2} w_k \phi_k(t - nT).
\]
where \( \phi_k(t) \) is the \( k \)-th user's unique waveform, a signal with support over \([0, T]\). The received signal consists of the sum of each user's delayed signal, appearing in additive noise:
\[
r(t) = \sum_{k=1}^{K} \sum_{n} b_k(n) w_k \phi_k(t - nT - \tau_k) + z(t).
\]
The users' basis functions are not orthogonal. Assume that the users are ordered so that \( \tau_1 \leq \tau_2 \leq \cdots \leq \tau_K < T \). A matched-filter (or correlator) output is obtained for each user over the \( n \)-th bit interval, as
\[
y_k(n) = \int_{-\infty}^{\infty} r(t) \phi_k(t - nT - \tau_k) \, dt.
\]
Let \( y(n) = [y_1(n), y_2(n), \ldots, y_K(n)]^T \) be the vector of matched filter outputs for all users at interval \( n \).

(a) Show that
\[
y(n) = [H(1)B(n-1) + H(0)B(n) + H(-1)B(n+1)]w + z(n)
\]
where \( H(m) \) is a correlation matrix with elements
\[
H_{ij}(m) = \int_{-\infty}^{\infty} \phi_i(t - \tau_i) \phi_j(t - mT - \tau_j) \, dt.
\]
\( B \) is a diagonal matrix of bits, \( B(n) = \text{diag}(b_1(n), b_2(n), \ldots, b_K(n)) \), \( w = [w_1, w_2, \ldots, w_K]^T \), and \( z(n) = [z_1(n), z_2(n), \ldots, z_K(bn)]^T \), where
\[
z_k(n) = \int_{-\infty}^{\infty} z(t) \phi_k(t - nT - \tau_k) \, dt.
(b) If \( z(t) \) is white with \( E[z(t)z(t-s)] = \sigma^2 \delta(t-s) \), show that \( z(n) \) satisfies

\[
E[z(n)z^T(m)] = \begin{cases} 
\sigma^2 H(0) & n = m, \\
\sigma^2 H(1) & n = m + 1, \\
\sigma^2 H(-1) & n = m - 1, \\
0 & \text{otherwise.}
\end{cases}
\]

3.21 References

The Hilbert approximation theory presented here is summarized from [209] and [177]. Some of the discussion about the Grammian matrix was drawn from [291].

The various windowing methods are described in [132, chapter 11]. A discussion of least-squares and minimum mean-squares filtering is in [132, 263, 291]. Our discussion of Wiener filtering is drawn from [165] and [316]. A thorough discussion of the spectral factorization problem appears in [248].

The Gram–Schmidt is discussed in most books on linear algebra. Specific results on numeric accuracy of the method can be found in [114].

Several variants on least-squares and constrained least-squares, including pseudocode for several useful algorithms, are in [197].

Orthogonal functions are widely discussed in [2], including an extensive table of polynomials orthogonal with respect to many weighting functions, and their properties. In addition to orthogonal polynomials in continuous time, there are also orthogonal polynomials in discrete variables. These are summarized in [2] and examined more thoroughly in [79] and [337]. A recent book describing a variety of orthogonal functions and their smoothness properties is [358].

The use of the function \( \sin(x)/x \) (the sinc function) as an orthogonal basis is introduced in [177]. An extensive discussion occurs in [323] and [322].

There has been an explosion of literature on wavelets and wavelet transforms. The definitive reference is probably [62]; see also [63]. Another book with a broad base of coverage is [53]. Among the generalizations discussed in these books are biorthogonal wavelets (in which different filters are used to reconstruct the signal than to analyze it), wavelet packets (choosing different trees of coefficients), and several other families of wavelets. A recent tutorial is [44]. A thorough discussion of implementation of wavelet transforms (and a variety of other useful transforms as well) is provided in [362]. A definitive reference on multirate signal processing is [342]; for a solid introduction to this area see [341].

IRLS is discussed in [43] and references therein, where the number of iterations required to design a filter is closely examined. An alternative viewpoint on estimation, using the \( L_1 \) norm for spectral estimation, is investigated in [294, 295, 73, 359]. A more thorough treatment is presented in [37].

The vector space viewpoint, signal constellations, and matched filters are presented in every text on digital communications. See, for example, [373], [261] or [35]. A historical treatment of orthogonal functions used in signaling is given in [377], which also presents some useful orthogonal functions other than those presented here.

There is a tremendous literature on orthogonal polynomials. A recent book is [358]. A classic reference is [337]. Additional information is found in [2].