Adaptive Signal Processing

Author: Jack Shih

First edit: May 6, 2025
Last edit: March 25, 2026

Contents

1 Z-Transform

1.1 Definition

The z-transform of a discrete-time signal $u[n]$ is defined as:

$$U(z) = \mathcal{Z}\{u[n]\} = \sum_{n=-\infty}^{\infty} u[n]z^{-n}, \ z \in \mathbb{C}$$

where z is a complex variable.

1.2 ROC

The Region of Convergence (ROC) is the set of all z such that $U(z)$ uniformly converges:

$$\text{ROC} = \{z \in \mathbb{C} \mid |U(z)| < \infty\}$$

1.3 Properties

Properties of the z-transform:

1. **Linearity:** $au_1(n) + bu_2(n) \leftrightarrow aU_1(z) + bU_2(z)$
2. **Time shift:** $u(n - n_0) \leftrightarrow z^{-n_0}U(z)$
3. **Convolution:** $u_1(n) * u_2(n) \leftrightarrow U_1(z)U_2(z)$

From property 2, we can use z^{-1} to denote unit delay operation.

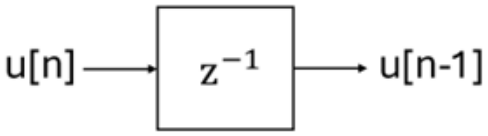


Figure 1: Unit delay operation using z^{-1}

2 LTI Filter

2.1 Definition

An LTI (Linear time-invariant) filter is completely described by its unit impulse response $h[n]$.

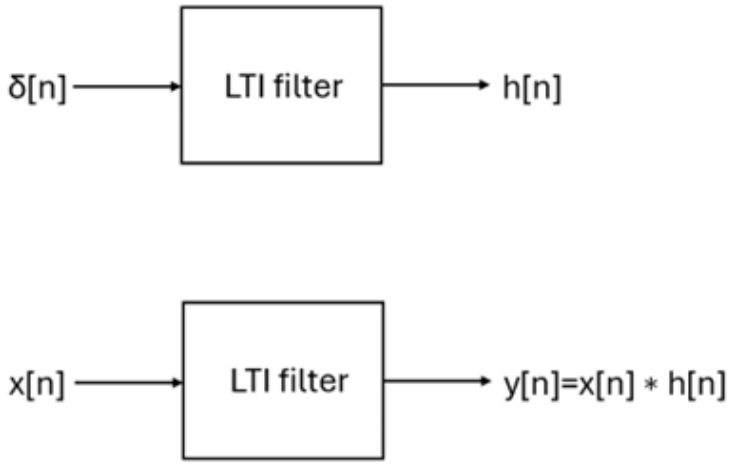


Figure 2: LTI system characterization by impulse response

2.2 In Z-Domain

In the z-domain, the output of an LTI filter is simply the input multiplied by the system's transfer function $H(z)$, which is the z-transform of the impulse response $h[n]$.

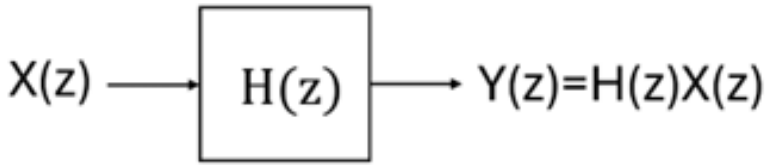


Figure 3: LTI filter in Z-domain with $Y(z) = H(z)X(z)$

When $z = e^{j\omega}$, $H(e^{j\omega})$ is called the frequency response of the system.

2.3 LTI Systems Described by LCCDE

A general LTI system can be described by LCCDE (Linear constant coefficient differential equations):

$$a_0y[n] + a_1y[n-1] + \ldots + a_Ny[n-N] = b_0x[n] + b_1x[n-1] + \ldots + b_Mx[n-M]$$

Or in more compact form:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

Taking the z-transform on both sides:

$$\sum_{k=0}^{N} a_k z^{-k} Y(z) = \sum_{k=0}^{M} b_k z^{-k} X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{B(z)}{A(z)}$$

The numerator $B(z)$ depends only on the input coefficients, and therefore represents the **forward path** of the system.

The denominator $A(z)$ depends only on the output coefficients, and thus represents the **feedback path**.

2.3.1 Zeros and Poles

The roots of $B(z)$ are the **zeros** of the filter, and the roots of $A(z)$ are the **poles** of the filter. The ROC of $H(z)$ must not include any poles of $H(z)$.

2.3.2 FIR Filter (All-zero)

The filter is FIR (Finite Impulse Response) if there is no pole except at $z = 0$. Hence, all the roots of $A(z)$ must occur at $z = 0$, and we can express it as:

$$\begin{aligned} A(z) &= az^m \\ \Rightarrow H(z) &= \frac{B(z)}{A(z)} = \frac{1}{a} B(z) z^{-m} \equiv B(z) \end{aligned}$$

This leads to:

$$\begin{aligned} Y(z) &= X(z)H(z) = X(z)B(z) = X(z)\sum_{k=0}^{M} b_k z^{-k} = \sum_{k=0}^{M} b_k z^{-k} X(z) \\ Y(z) &= b_0 X(z) + b_1 z^{-1} X(z) + \ldots + b_M z^{-M} X(z) \end{aligned}$$

$$\boxed{y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]}$$

$y[n]$ is the weighted sum of $x[n-k]$. That is, the output only depends on the current and past input. When $x[n] = \delta[n]$:

$$h[n] = b_0\delta[n] + b_1\delta[n-1] + \ldots + b_M\delta[n-M]$$

$$h[n] = \begin{cases} b_0, & n = 0 \\ b_1, & n = 1 \\ \vdots & \\ b_M, & n = M \\ 0, & n > M \end{cases}$$

The impulse response becomes zero when the time index exceeds M, which is why it is called a **finite** impulse response.

Notation remark $H[z] \equiv B(z)$

While we may loosely write $H(z) \equiv B(z)$ to highlight the FIR nature (no feedback), this equivalence ignores the delay and scaling introduced by $A(z) = az^m$. Strictly speaking, the actual system includes a delay of m and scaling by $\frac{1}{a}$, so we only get $H(z) = B(z)$ when $a = 1$ and $m = 0$.

A more precise expression for the output is:

$$y[n] = \frac{1}{a}\left(b_0\, x[n-m] + b_1\, x[n-m-1)] + \ldots + b_M\, x[n-m-M)]\right)$$

2.3.3 IIR Filter (All-pole)

The filter is all-pole IIR (Infinite Impulse Response) if there is no zero except at $z = 0$. Hence, all the roots of $B(z)$ must occur at $z = 0$, and we can express it as:

$$B(z) = bz^m$$
$$\Rightarrow H(z) = \frac{B(z)}{A(z)} = \frac{bz^m}{A(z)} \equiv \frac{1}{A(z)}$$

This leads to:

$$X(z) = \frac{Y(z)}{H(z)} = Y(z)A(z) = Y(z)\sum_{k=0}^{N} a_k z^{-k} = \sum_{k=0}^{N} a_k z^{-k} Y(z)$$
$$X(z) = a_0 Y(z) + a_1 z^{-1} Y(z) + \ldots + a_N z^{-N} Y(z)$$
$$x[n] = a_0 y[n] + a_1 y[n-1] + \ldots + a_N y[n-N]$$

For $a_0 = 1$:

$$\boxed{y[n] = x[n] - a_1 y[n-1] - \ldots - a_N y[n-N]}$$

Since the output $y[n]$ depends on its own past values recursively, even if the input is a unit impulse, $y[n]$ may never become exactly zero. This is why it is called **infinite** impulse response.

2.3.4 IIR Filter (Zero-pole)

The filter is general IIR if there is at least one zero and one pole at $z \neq 0$.

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

Let:

$$\frac{Y(z)}{X(z)} = \sum_{k=0}^{M} b_k z^{-k} \text{ and } \frac{X(z)}{U(z)} = \frac{1}{\sum_{k=0}^{N} a_k z^{-k}}$$

$$\text{then } H(z) = \frac{Y(z)}{X(z)} \frac{X(z)}{U(z)}$$

Hence, a general IIR filter can be viewed as a cascade of an all-pole IIR filter followed by a FIR filter.

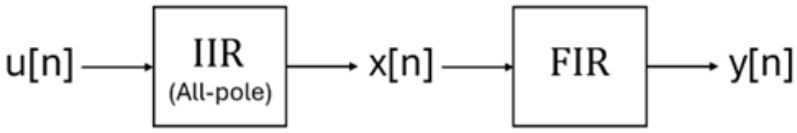


Figure 4: General IIR filter as cascade of all-pole IIR and FIR filters

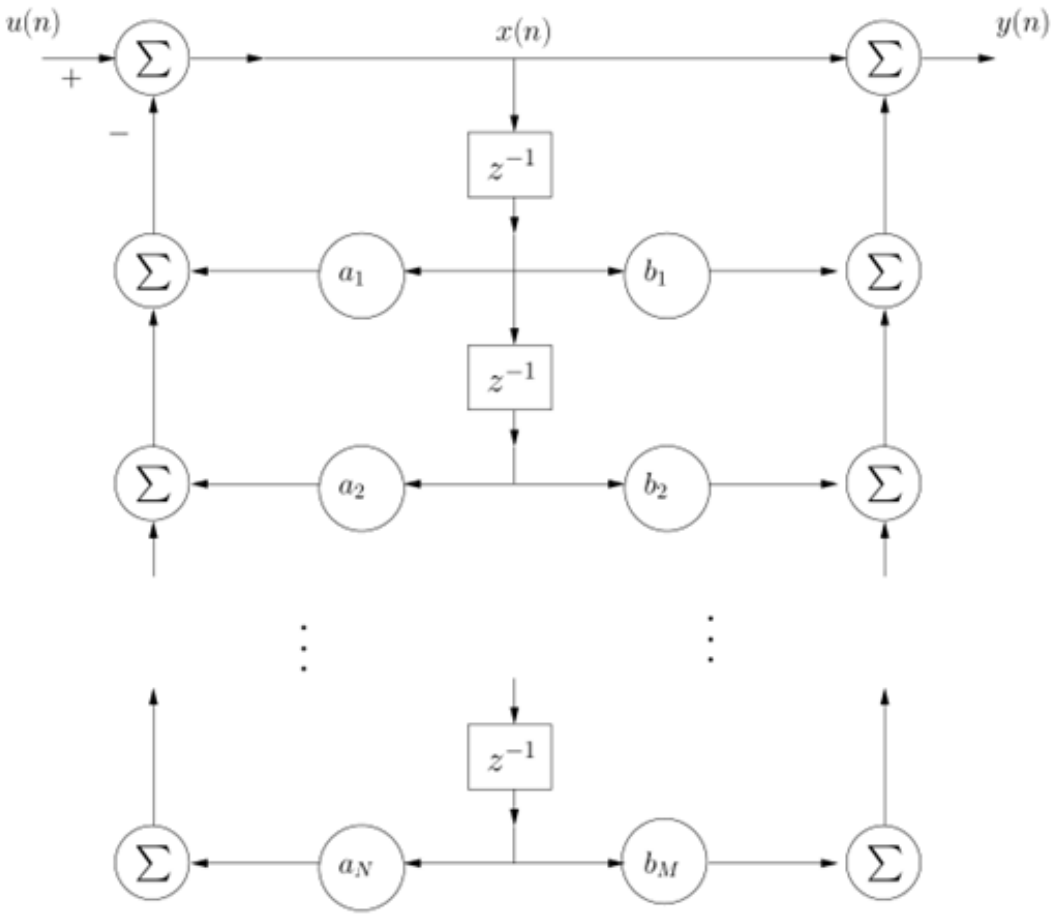


Figure 5: Block diagram of general IIR filter with feedback and forward paths

2.4 Causal and Stable

2.4.1 Causality

Causality: An LTI system is causal if its output at any time depends only on the current and past inputs, and has **no dependence on future inputs**.

In mathematical terms, a LTI filter is causal if and only if:

$$h[n] = 0, \text{ for } n < 0$$

This is because a present-time input $\delta[n]$ cannot produce a response in the past. Therefore, the z-transform of a causal system becomes:

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

This is a **one-sided z-transform**, its region of convergence is the exterior of the largest pole (see Appendix A):

$$\text{ROC} = \{z \in \mathbb{C} \mid |z| > \max(|p_i|)\}$$

where p_i are the poles of $H(z)$.

2.4.2 Stability

An LTI system is BIBO stable (bounded-input bounded-output) if every bounded input leads to a bounded output:

$$\begin{aligned}
|y[n]| &= |h[n] * x[n]| \\
&= \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right| \\
&\leq \sum_{k=-\infty}^{\infty} |h[k]| \, |x[n-k]| \\
&\leq \sum_{k=-\infty}^{\infty} |h[k]||x_{\max}| \\
&= |x_{\max}| \sum_{k=-\infty}^{\infty} |h[k]|
\end{aligned}$$

In other words:

$$\boxed{\text{BIBO stable} \Leftrightarrow \sum_{k=-\infty}^{\infty} |h[k]| < \infty}$$

Consider the z-transform of the system:

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

The ROC of the z-transform:

$$\begin{aligned}
\text{ROC} &= \{z \in \mathbb{C} \mid |H(z)| < \infty\} \\
&= \left\{ z \in \mathbb{C} \mid \sum_{n=-\infty}^{\infty} |h[n]z^{-n}| < \infty \right\} \\
&= \left\{ z \in \mathbb{C} \mid \sum_{n=-\infty}^{\infty} |h[n]| \, |z^{-n}| < \infty \right\}
\end{aligned}$$

We can see that when $|z| = 1$, $H(z)$ must converge, therefore, when $z = e^{j\omega}$, $H(z)$ must converge. In summary:

Theorem: BIBO Stability Equivalence

BIBO stable $\Leftrightarrow$ Impulse response $h[n]$ is absolutely summable
$\Leftrightarrow$ System response $H(z)$ converges on $|z| = 1$
$\Leftrightarrow$ ROC includes unit circle

2.4.3 Causality and Stability

Key System Conditions:
Causal: ROC $= \{z \in \mathbb{C} \mid |z| > \max(|\text{poles of } H(z)|)\}$
Stable: $z = e^{j\omega} \in$ ROC
Causal and Stable: All poles of $H(z)$ lie inside the unit circle

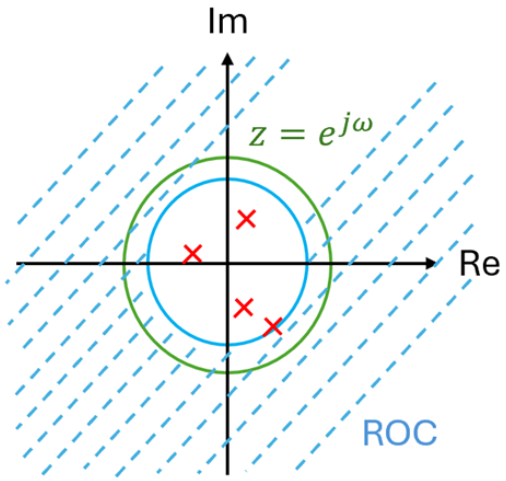


Figure 6: ROC for a Causal and Stable System in Complex Plane

2.4.4 Z-Transform to DTFT

If the ROC of $H(z)$ includes the unit circle $|z| = 1$, then we can evaluate the z-transform on the unit circle by substituting $z = e^{j\omega}$:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$$

This is the Discrete-Time Fourier Transform (DTFT) of $h[n]$. Note that DTFT exists if and only if the Z-transform converges on the unit circle, which happens **only when the system is BIBO stable**.

2.4.5 DTFT to DFT

If we sample the DTFT $H(e^{j\omega})$ at N equally spaced points around the unit circle, we obtain the N-point discrete Fourier transform (DFT).

This sampling is performed at frequencies $\omega_k = \frac{2\pi k}{N}$, where $k = 0, 1, 2, \ldots, N-1$, i.e., uniformly spaced samples on the unit circle over $[0, 2\pi)$:

$$H(e^{j\omega})|_{\omega=2\pi k/N} = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}|_{\omega=2\pi k/N}$$

$$H[k] = \sum_{n=0}^{N-1} h[n] e^{-j2\pi \frac{k}{N} n}$$

Definition

If the signal $h[n]$ is of length N (or assumed to be periodic with period N), then the DFT is:

$$H[k] = \sum_{n=0}^{N-1} h[n] e^{-j2\pi \frac{kn}{N}}, \quad k = 0, 1, 2, \ldots, N-1$$

The inverse DFT is:

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} H[k] e^{j2\pi \frac{kn}{N}}, \quad n = 0, 1, 2, \ldots, N-1$$

Table 1: Comparison of Z-Transform, DTFT, and DFT

	Z Transform	**DTFT**	**DFT**
Domain	Complex plane	Unit circle	Equally spaced points on unit circle
Input	Infinite-length discrete signals	Stable signals (length can be infinite or finite)	Finite-length signals
Output	Complex function $H(z)$	Continuous function $H(e^{j\omega})$	Length-N sequence $H[k]$
Purpose	Analyze ROC and pole-zero distribution	Analyze frequency response	Digital computation (e.g., FFT)

3 Stochastic Process

3.1 Random Variable

A **probability space** is a mathematical framework used to model random experiments. It is defined as a triple $(\Omega, \mathcal{F}, P)$, where:

- Ω is the sample space, which is the set of all possible outcomes of a random experiment.
- $\mathcal{F}$ is the event space i.e., a set of all subsets of Ω. It satisfies:
  - $\Omega \in \mathcal{F}$ (contains the whole space),
  - If $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$ (closed under complementation),
  - If $A_1, A_2, \cdots \in \mathcal{F}$ are countable disjoint events, then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ (closed under countable unions).

  Note: $|\mathcal{F}| = 2^{|\Omega|}$ *(e.g.* $\Omega = \{A, B, C\}$*, then the number of events (subsets) of* Ω *is* $2^3 = 8$*)*
- P is a **probability measure**, assigning a probability to each event, and satisfies:
  - $P : \mathcal{F} \to [0, 1]$
  - $P(\Omega) = 1$
  - For disjoint events $A_i \in \mathcal{F}$,

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

A **random variable** X on a probability space is a real-valued function:

$$X : \Omega \to \mathbb{R}$$

Intuitively, a random variable maps an outcome $\omega \in \Omega$ to a real number $X(\omega)$:

$$X = \{X(\omega_1), X(\omega_2), \ldots\}$$

3.2 Random Process

A **random process** $X[n, \omega]$ on a probability space is a real-valued function:

$$X[n, \omega] : \mathbb{Z} \times \Omega \mapsto \mathbb{R}$$

If we fixed n, we obtain a random variable:

$$X[n_0, \omega] = \{X[n_0, \omega_1], X([n_0, \omega_2], \ldots\}$$

If we fixed ω , we obtain a deterministic signal (function of time):

$$X[n, \omega_k] = X[n]$$

Hence, a random process $X[n, \omega]$ represents a **collection of signals**:

$$X[n, \omega] = \{X[n, \omega_1], X[n, \omega_2], \ldots\} = \{X_1[n], X_2[n], \ldots\}$$

For example, speech signals or thermal noise can be modeled as random processes. Each time the process "occurs" (i.e., a specific ω is realized), we obtain a deterministic signal — a single **realization** of the process.

Notation remark $X[n]$

Although the full form of a random process is $X[n, \omega]$, it is common in practice to write it simply as $X[n]$, especially when the dependence on ω is understood implicitly. This notation is context-dependent and corresponds to two commonly used terms:

- When ω is fixed, $X[n]$ denotes a single realization, also known as a **deterministic signal**.
- When ω is not fixed, $X[n]$ refers to the full random process, also known as a **random signal** — i.e., the collection of all possible realizations $\{X[n, \omega]\}_{\omega \in \Omega}$.

3.3 Wide Sense Stationary (WSS)

A random process is said to be WSS if its mean is time-invariant and its autocovariance depends only on the time difference (i.e., the lag).

3.3.1 Definition

The first and second-order moments of a complex-valued random process $u[n]$ are defined as:

- **Mean:**
$$\mu[n] = \mathbb{E}[u[n]]$$
- **Autocorrelation:**
$$r[m, n] = \mathbb{E}[u[m]u^*[n]]$$
- **Autocovariance:**
$$c[m, n] = \mathbb{E}\left[(u[m] - \mu[m])\left(u^*[n] - \mu^*[n]\right)\right] = r[m, n] - \mu[m]\mu^*[n]$$

Theorem

If $u[n]$ is WSS, then:

- The mean is constant:
$$\mu[n] = \mu$$

- The autocorrelation depends only on the time lag:

$$r[m,n] = r[m-n] \quad \text{or written as} \quad \boxed{r[l] = \mathbb{E}[u[n]u^*[n-l]]}$$

- The autocovariance becomes:

$$c[m,n] = r[m,n] - \mu[m]\mu^*[n] = r[m-n] - \mu\mu^* = c[m-n]$$

- Conjugate symmetry:

$$\begin{aligned} r^*[l] &= \mathbb{E}^*[u[n]u^*[n-l]] \\ &= \mathbb{E}[u^*[n]u[n-l]] \\ &= \mathbb{E}[u[n-l]u^*[n]] \\ &= r[-l] \end{aligned}$$

3.3.2 Estimating Moments

For a WSS process, time averages can be used to approximate ensemble moments:

$$\hat{\mu}_N = \frac{1}{N}\sum_{n=0}^{N-1} u[n], \quad \hat{r}_N[k] = \frac{1}{N}\sum_{n=0}^{N-1} u[n]u^*[n-k]$$

The process $u[n]$ is called a **mean-ergodic process**, if $\hat{\mu}_N$ converges to μ (in the mean-squared error sense):

$$\lim_{N\to\infty} \mathbb{E}\left[|\hat{\mu}_N - \mu|^2\right] = 0$$

The process $u[n]$ is called a **correlation-ergodic process**, if $\hat{r}_N[k]$ converges to $r[k]$:

$$\lim_{N\to\infty} \mathbb{E}\left[|\hat{r}_N[k] - r[k]|^2\right] = 0$$

3.3.3 Power Spectral Density

For a WSS process $u[n]$ the second-order statistic in the time domain is its autocorrelation $r[l]$. By taking the DTFT of $r[l]$, we obtain the second-order statistic in the frequency domain, known as the power spectral density (PSD):

$$S(\omega) = \text{DTFT}\{r[l]\} = \sum_{l=-\infty}^{\infty} r[l]e^{-j\omega l}$$

Conversely, the autocorrelation function can be recovered via the inverse DTFT:

$$r[l] = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)e^{j\omega l}d\omega$$

Properties of PSD $S(\omega)$:

1. **DTFT pair:** $r[l] \leftrightarrow S(\omega)$
2. **Periodicity:** $S(\omega + 2k\pi) = S(\omega)$
3. **Conjugate symmetry:** $S^*(\omega) = S(-\omega)$
4. **Total power of $u[n]$:** $\mathbb{E}[|u[n]|^2] = r[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)d\omega$
5. **Nonnegativity:** $S(\omega) \geq 0$

The fourth property shows that $S(\omega)$ is like a power density in the frequency domain. Since physical power cannot be negative, it also explains why $S(\omega) \geq 0$.

3.3.4 WSS Process into LTI filter

If a WSS process $u[n]$ with mean μ_u and autocorrelation $r_u[l]$ pass through a LTI filter with impulse response $h[n]$, then the output is:

$$y[n] = h[n] * u[n]$$

The mean of the output is:

$$\mu_y = \mu_u H(e^{j0})$$

The autocorrelation of the output is:

$$r_y[l] = h[l] * h^*[-l] * r_u[l]$$

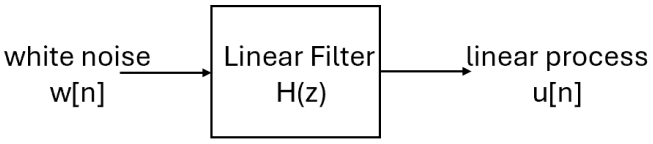
Then we take DTFT and z-transform both side, we obtain:

$$S_y(e^{j\omega}) = |H(e^{j\omega})|^2 S_u(e^{j\omega}), \quad S_y(z) = H(z)H^*(\frac{1}{z^*})S_u(z)$$

3.4 Linear Stochastic Models

Linear stochastic models are used to simulate or approximate real-world random processes. These models treat the process as the output of a LTI system driven by white noise $w[n]$, which is zero-mean and uncorrelated. The system is characterized by a transfer function $H(z)$, and the output is the desired random process $u[n]$:

$$\mathcal{Z}\{u[n]\} = H(z)\mathcal{Z}\{w[n]\}$$



Since white noise is WSS, this setup is exactly a special case of the WSS process passed through an LTI filter. Therefore, all results from Section 3.3.4 can be directly applied to analyze linear stochastic models.

3.4.1 MA Model

A moving average (MA) model describes a process generated by passing white noise through a **FIR filter**. It is suitable for simulating processes with short-term fluctuations and without long-term memory. The LCCDE is:

$$u[n] = w[n] + b_1 w[n-1] + \ldots + b_k w[n-k]$$

3.4.2 AR Model

An autoregressive (AR) model represents a system where the output depends on its own past values and white noise. It corresponds to an **all-pole IIR filter**. This model is suitable for processes with long-term memory and autocorrelation. The LCCDE is:

$$u[n] = w[n] + a_1 u[n-1] + \ldots + a_k u[n-k]$$

3.4.3 ARMA Model

An autoregressive moving average (ARMA) model combines both AR and MA characteristics, and is suitable for processes that exhibit both short-term fluctuations and long-term dependencies. The system is modeled as a **pole-zero IIR filter**, and the corresponding LCCDE is:

$$u[n] + a_1 u[n-1] + \ldots + a_k u[n-k] = w[n] + b_1 w[n-1] + \ldots + b_m w[n-m]$$

3.5 Correlation matrix

For M random variables sampled from a random process u[n] at times $n, n-1, \ldots, n-M+1$, it can be written as an $M \times 1$ **random vector** (i.e., a vector of random variables):

$$\mathbf{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix}$$

Definition

A statistic of the random vector $\mathbf{u}[n]$ is the **correlation matrix**, defined as

$$\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$$

where ^{H} denotes the Hermitian transpose. For any complex matrix $\mathbf{M}$, this is defined as

$$\mathbf{M}^H = \left(\mathbf{M}^T\right)^* = \left(\mathbf{M}^*\right)^T$$

Expand the correlation matrix:

$$\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$$

$$= \mathbb{E}\left(\begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix} \begin{bmatrix} u^*[n] & u^*[n-1] & \cdots & u^*[n-M+1] \end{bmatrix} \right)$$

$$= \begin{bmatrix} \mathbb{E}[u[n]u^*[n]] & \mathbb{E}[u[n]u^*[n-1]] & \cdots & \mathbb{E}[u[n]u^*[n-M+1]] \\ \mathbb{E}[u[n-1]u^*[n]] & \mathbb{E}[u[n-1]u^*[n-1]] & \cdots & \mathbb{E}[u[n-1]u^*[n-M+1]] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[u[n-M+1]u^*[n]] & \mathbb{E}[u[n-M+1]u^*[n-1]] & \cdots & \mathbb{E}[u[n-M+1]u^*[n-M+1]] \end{bmatrix}$$

$$= \begin{bmatrix} r[n,n] & r[n,n-1] & \cdots & r[n,n-M+1] \\ r[n-1,n] & r[n-1,n-1] & \cdots & r[n-1,n-M+1] \\ \vdots & \vdots & \ddots & \vdots \\ r[n-M+1,n] & r[n-M+1,n-1] & \cdots & r[n-M+1,n-M+1] \end{bmatrix}$$

Hence, $\mathbf{R}_{kl} = r[n-k+1, n-l+1]$.

Theorem

For a WSS process $u[n]$, the autocorrelation matrix is:

$$\mathbf{R} = \begin{bmatrix} r[0] & r[1] & \cdots & r[M-1] \\ r[-1] & r[0] & \cdots & r[M-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[-M+1] & r[-M+2] & \cdots & r[0] \end{bmatrix}$$

That is, the element at row k and column l is:

$$\mathbf{R}_{kl} = r[l-k]$$

Notice that $l-k = (n-k+1) - (n-l+1)$ is the time difference.

Furthermore, using the conjugate symmetry property $r(-l) = r^*(l)$, another equivalent representation is:

$$\mathbf{R} = \begin{bmatrix} r[0] & r[1] & \cdots & r[M-1] \\ r^*[1] & r[0] & \cdots & r[M-2] \\ \vdots & \vdots & \ddots & \vdots \\ r^*[M-1] & r^*[M-2] & \cdots & r[0] \end{bmatrix}$$

Properties of R for a WSS u[n]:

1. $\mathbf{R}$ is Hermitian i.e., $\mathbf{R} = \mathbf{R}^H$
2. $\mathbf{R}$ is Toeplitz, i.e., every diagonal from top-left to bottom-right has the same elements
3. $\mathbf{R}$ is nonnegative definite, i.e., $\mathbf{x^H R x} \geq 0$ for any $\mathbf{x} \neq 0$
4. The correlation matrix of $\mathbf{u}^B[n]$ (reversal permutation of $\mathbf{u}[n]$) is $\mathbf{R}^* = \mathbf{R}^T$
5. $\mathbf{R}_{M+1} = \begin{bmatrix} r[0] & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_M \end{bmatrix} = \begin{bmatrix} \mathbf{R}_M & (\mathbf{r}^B)^* \\ (\mathbf{r}^B)^T & r[0] \end{bmatrix}$

   where $\mathbf{r} = \begin{bmatrix} r[-1] & r[-2] & \cdots & r[-M] \end{bmatrix}^T$, and $\mathbf{r}^B$ is its reversal permutation.
6. In the presence of AWGN, i.e., $\mathbf{u}(n) = \mathbf{s}(n) + \mathbf{v}(n)$ leads to a correlation matrix $\mathbf{R} = \mathbf{R}_s + \sigma_v^2 \mathbf{I}$. Then, $\mathbf{R}$ is positive definite and nonsingular. (see Appendix B)

Note: For the autocorrelation of a $(M+1) \times 1$ $\mathbf{u}[n]$*:*

$$\mathbf{R}_{M+1} = \begin{bmatrix} r[0] & r[1] & \cdots & r[M-1] \\ r[-1] & r[0] & \cdots & r[M-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[-M] & r[-M+1] & \cdots & r[0] \end{bmatrix}$$

4 Wiener Filters

4.1 Motivation

In many signal processing applications, we wish to estimate or recover a desired signal $d[n]$ from a noisy or distorted observation $u[n]$. This kind of problem shows up in situations like:

- Denoising speech or audio corrupted by background noise
- Channel equalization in digital communication systems
- Predicting future values in time series

To estimate desired signal $d[n]$, we pass the observed signal $u[n]$ through a filter to produce an output $y[n]$, which serve as a guess of the original signal:

$$y[n] = \text{output of filter applied to } u[n] \approx d[n]$$

The filter is designed such that the estimation error:

$$e[n] = d[n] - y[n]$$

is as small as possible in some well-defined sense.

4.2 Model

To estimate the desired signal $d[n]$ from an observed signal $u[n]$, we propose to pass $u[n]$ through a linear filter whose output $y[n]$ will serve as our estimate of $d[n]$. The model is established in four key steps.

Firstly, we assume the filter is a linear time-invariant (LTI) finite impulse response (FIR) filter of length M. That is, the output at time n is a weighted sum of the current and past inputs:

$$y[n] = \sum_{k=0}^{M-1} h[k]\, u[n-k]$$

This is consistent with the FIR model introduced in Section 2.3.2, where the coefficients $h[k]$ are typically fixed based on system specifications or impulse response design.

Secondly, unlike in system analysis where filter coefficients are assumed known, we treat them as tunable parameters to be optimized. Let these coefficients be $w_k \in \mathbb{C}$, and we express them with complex conjugation w_k^* to simplify derivations in the context of mean-square error minimization (a common practice in signal estimation):

$$y[n] = \sum_{k=0}^{M-1} w_k^*\, u[n-k]$$

Thirdly, to simplify notation and enable matrix-based derivation, we adopt a compact vector representation. Define the weight (coefficient) vector and input vector as:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}, \quad \mathbf{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix}$$

Then, the filter output becomes a Hermitian inner product:

$$y[n] = \mathbf{w}^H \mathbf{u}[n]$$

This representation clearly identifies $\mathbf{w}$ as the optimization variable and $\mathbf{u}[n]$ as the observed input data vector at time n.

Finally, we define the instantaneous estimation error $e[n]$ as the difference between the desired signal and the filter output:

$$e[n] = d[n] - y[n] = d[n] - \mathbf{w}^H \mathbf{u}[n]$$

This error measures the deviation of our estimate from the true desired signal. Since $e[n]$ is generally complex and not directly comparable in magnitude, we consider its squared magnitude $|e[n]|^2$, which is real. To capture its typical behavior over time, we take the expected value. The resulting quantity is the **cost function** we aim to minimize, also known as the **Minimum Mean-Square Error (MMSE)**:

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

4.3 Solution of Wiener Filter

With the cost function $J(\mathbf{w}) = \mathbb{E}[|e[n]|^2]$ defined, we now seek to determine the optimal weight vector that minimize cost function:

$$\mathbf{w}_o = \arg\min_{\mathbf{w}} J(\mathbf{w}) = \arg\min_{\mathbf{w}} \mathbb{E}\left[\left|d[n] - \mathbf{w}^H \mathbf{u}[n]\right|^2\right]$$

4.3.1 Quadratic Nature of Cost Function

We now expand $J(\mathbf{w})$ to reveal its quadratic nature. Starting from the expanded product:

$$\begin{aligned} J(\mathbf{w}) &= \mathbb{E}\left[\left(d[n] - \mathbf{w}^H \mathbf{u}[n]\right)^H \left(d[n] - \mathbf{w}^H \mathbf{u}[n]\right)\right] \\ &= \mathbb{E}\left[\left(d^*[n] - \mathbf{u}^H[n]\mathbf{w}\right)\left(d[n] - \mathbf{w}^H \mathbf{u}[n]\right)\right] \\ &= \mathbb{E}\left[|d[n]|^2 - d^*[n]\mathbf{w}^H \mathbf{u}[n] - \mathbf{u}^H[n]\mathbf{w}d[n] + \mathbf{u}^H[n]\mathbf{w}\mathbf{w}^H \mathbf{u}[n]\right] \end{aligned}$$

By the linearity of expectation and the fact that $\mathbf{w}$ is deterministic, we can take expectations term by term:

$$\begin{aligned} J(\mathbf{w}) &= \mathbb{E}[|d[n]|^2] - \mathbf{w}^H \mathbb{E}[d^*[n]\mathbf{u}[n]] - \mathbb{E}[d[n]\mathbf{u}^H[n]]\mathbf{w} + \mathbf{w}^H \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]\mathbf{w} \\ &= \mathbf{w}^H \mathbf{R}\mathbf{w} - \mathbf{w}^H \mathbf{p} - \mathbf{p}^H \mathbf{w} + \sigma_d^2 \end{aligned}$$

where $\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$ is the autocorrelation matrix of the input vector, $\mathbf{p} = \mathbb{E}[\mathbf{u}[n]d^*[n]]$ is the cross-correlation vector between input and desired signal, $\sigma_d^2 = \mathbb{E}[|d[n]|^2]$ is the power of the desired signal.

This expression shows that $J(\mathbf{w})$ is a general quadratic form, and will attain a unique minimum if $\mathbf{R}$ is positive definite.

Math Background: Quadratic Form

A **quadratic form (quadratic homogeneous polynomial)** is a polynomial

where all terms are of degree two. For example,

$$f(x, y) = 3x^2 + 2xy + 4y^2$$

is a quadratic form in two variables.

Such functions are homogeneous of degree two, meaning that for any scalar $k \in \mathbb{R}$,

$$f(kx, ky) = k^2 f(x, y)$$

The example above can be written compactly in matrix form:

$$f(x, y) = 3x^2 + 2xy + 4y^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Hence, a quadratic form is

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symmetric matrix.

The definite of $\mathbf{A}$ determines the convexity of the function $f(\mathbf{x})$:

- Positive definite ($\forall \mathbf{x} \neq 0, \mathbf{x}^T \mathbf{A} \mathbf{x} > 0$): $f(\mathbf{0}) = \mathbf{0}$ is the unique global minimum
- Positive semidefinite ($\forall \mathbf{x} \neq 0, \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$): $f(\mathbf{0}) = \mathbf{0}$ is a global minimum
- Indefinite (neither definite nor semidefinite): Global minimum not exists

A **general quadratic form (nonhomogeneous)** is

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a Symmetric matrix, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$.

If $\mathbf{A}$ is positive definite, then $f(\mathbf{x})$ has a unique minimum and $\mathbf{A}^{-1}$ exists.

To find the minimum, we set the gradient of function to zero:

$$\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{0}$$

$$\mathbf{x} = -\frac{1}{2} \mathbf{A}^{-1} \mathbf{b}$$

which is the optimal point.

For one variable $\mathbf{x} = x$, the optimal point becomes

$$x = \frac{-b}{2a}$$

which is the vertex of a parabola.

In the **complex domain** $\mathbf{x} \in \mathbb{C}^n$, the general quadratic form becomes

$$f(\mathbf{x}) = \mathbf{x}^H \mathbf{A}\mathbf{x} + \mathbf{b}^H \mathbf{x} + c$$

where $\mathbf{A} \in \mathbb{C}^{n\times n}$ is a Hermitian matrix, $b \in \mathbb{C}^n$ and $c \in \mathbb{C}$.

To find the minimum, we have to consider the gradient (Wirtinger derivatives) of function to zero with respect to both $\mathbf{x}$ and $\mathbf{x}^*$.

However, if $f : \mathbb{C}^n \to \mathbb{R}$, then

$$\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)^* = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}^*}$$

which shows that if one gradient is zero, then the other is also zero.

To ensure f is real-valued, we can let f as:

$$f(\mathbf{x}) = \mathbf{x}^H \mathbf{A}\mathbf{x} + \mathbf{b}^H \mathbf{x} + \mathbf{x}^H \mathbf{b} + c$$

this is because $\mathbf{x}^H \mathbf{A}\mathbf{x}$ is real for Hermitian $\mathbf{A}$ and $\mathbf{b}^H \mathbf{x} + \mathbf{x}^H \mathbf{b} = 2\mathrm{Re}(\mathbf{b}^H \mathbf{x})$.

So the optimal point in this case (f is real-valued function and $\mathbf{A}$ is positive definite) is

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}^*} = \mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{0}$$
$$\mathbf{x} = -\mathbf{A}^{-1}\mathbf{b}$$

4.3.2 Optimal Solution

Compare the general quadratic form and the cost function:

$$f(\mathbf{x}) = \mathbf{x}^H \mathbf{A}\mathbf{x} + \mathbf{b}^H \mathbf{x} + \mathbf{x}^H \mathbf{b} + c$$
$$J(\mathbf{w}) = \mathbf{w}^H \mathbf{R}\mathbf{w} - \mathbf{w}^H \mathbf{p} - \mathbf{p}^H \mathbf{w} + \sigma_d^2$$

We get the optimal weight vector to minimize the cost function:

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$$

The minimum of cost function is

$$\begin{aligned} J_{\min} &= J(\mathbf{w}_o) \\ &= \mathbf{w}_o^H \mathbf{R}\mathbf{w}_o - \mathbf{w}_o^H \mathbf{p} - \mathbf{p}^H \mathbf{w}_o + \sigma_d^2 \\ &= \mathbf{p}^H \mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{p} - \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} - \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} + \sigma_d^2 \\ &= \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} - 2\mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} + \sigma_d^2 \\ &= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} \\ &= \sigma_d^2 - \mathbf{w}_o^H \mathbf{R}^{-1}\mathbf{w}_o \end{aligned}$$

Wiener–Hopf Equation

The optimal solution satisfies

$$\mathbf{R}\mathbf{w}_o = \mathbf{p},$$

which is called the **Wiener–Hopf equation**.

The filter whose coefficient vector $\mathbf{w}_o$ satisfy the Wiener-Hopf equation is called the **Wiener filter** for the MMSE problem.

4.4 Eigen-Domain Representation of Cost Function

From the property of $\mathbf{p} = \mathbf{R}\mathbf{w}_o$ and $\mathbf{R}$ is positive definite, the general cost function or the MSE of arbitrary $\mathbf{w}$ is

$$\begin{aligned} J(\mathbf{w}) &= \mathbf{w}^H\mathbf{R}\mathbf{w} - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \sigma_d^2 \\ &= \mathbf{w}^H\mathbf{R}\mathbf{w} - \mathbf{w}^H\mathbf{R}\mathbf{w}_o - \mathbf{w}_o^H\mathbf{R}\mathbf{w} + \sigma_d^2 \\ &= \mathbf{w}^H\mathbf{R}\mathbf{w} - \mathbf{w}^H\mathbf{R}\mathbf{w}_o - \mathbf{w}_o^H\mathbf{R}\mathbf{w} + \mathbf{w}_o^H\mathbf{R}\mathbf{w}_o - \mathbf{w}_o^H\mathbf{R}\mathbf{w}_o + \sigma_d^2 \\ &= (\mathbf{w} - \mathbf{w}_o)^H\mathbf{R}(\mathbf{w} - \mathbf{w}_o) + J_{\min} \\ &\geq J_{\min} \end{aligned}$$

The cost function is a quadratic surface centered at the optimal point $(\mathbf{w}_o, J_{\min})$, with curvature determined by $\mathbf{R}$.

Due to the Hermitian property of $\mathbf{R}$, the eigen-decomposition of $\mathbf{R}$ exists:

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$$

where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_M \end{bmatrix}, \quad \mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M)$$

with $\mathbf{q}_i$ being the eigenvectors and λ_i the corresponding eigenvalues of $\mathbf{R}$. Since $\mathbf{R}$ is Hermitian, $\mathbf{Q}$ is a unitary matrix, i.e., $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$.

Substituting into the cost function:

$$\begin{aligned} J(\mathbf{w}) &= (\mathbf{w} - \mathbf{w}_o)^H\mathbf{R}(\mathbf{w} - \mathbf{w}_o) + J_{\min} \\ &= (\mathbf{w} - \mathbf{w}_o)^H\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H(\mathbf{w} - \mathbf{w}_o) + J_{\min} \\ &= \mathbf{v}^H\mathbf{\Lambda}\mathbf{v} + J_{\min} \end{aligned}$$

where

$$\mathbf{v} = \mathbf{Q}^H(\mathbf{w} - \mathbf{w}_o) = (v_1, v_2, \ldots, v_M)^T$$

is a rotated and shifted version of $\mathbf{w}$.

Since $\mathbf{\Lambda}$ is diagonal, there are no cross-coupling terms:

$$J(\mathbf{w}) = \sum_{k=1}^{M} \lambda_k |v_k|^2 + J_{\min} \geq J_{\min}$$

Each term $\lambda_k|v_k|^2$ represents the contribution of the error along the eigenvector direction $\mathbf{q}_k$, weighted by the corresponding eigenvalue λ_k.

4.5 Perspective of Orthogonality

Another way to interpret the Wiener solution is through the principle of orthogonality. Starting from the optimal condition, that is, Wiener–Hopf equation:

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}$$

where $\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$ and $\mathbf{p} = \mathbb{E}[\mathbf{u}[n]d^*[n]]$.

Substituting these definitions into the Wiener–Hopf equation:

$$\mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]\,\mathbf{w}_o = \mathbb{E}[\mathbf{u}[n]d^*[n]]$$
$$\mathbb{E}[\mathbf{u}[n]d^*[n] - \mathbf{u}[n]\mathbf{u}^H[n]\mathbf{w}_o] = \mathbf{0}$$
$$\mathbb{E}\left[\mathbf{u}[n]\,(d[n] - \mathbf{w}_o^H\mathbf{u}[n])^*\right] = \mathbf{0}$$

Hence,

$$\mathbb{E}[\mathbf{u}[n]e_o^*[n]] = \mathbf{0}$$

This shows that at the optimal point, the error $e_o[n]$ is orthogonal to the input vector $\mathbf{u}[n]$.

Equivalently, in scalar form,

$$\mathbb{E}[u[n-k]\,e_o^*[n]] = 0, \qquad k = 0, 1, \ldots, M-1$$

Hence the Wiener filter achieves its MMSE precisely when the estimation error is orthogonal to each of the input components.

The orthogonality condition also gives a geometric view. At the optimum, the desired signal can be decomposed as

$$d[n] = y_o[n] + e_o[n]$$

where

$$y_o[n] = \mathbf{w}_o^H\mathbf{u}[n] = \sum_{k=0}^{M-1} w_k^* u[n-k]$$

which lies in the subspace $\mathcal{U}_n = \text{span}\{u[n], u[n-1], \ldots, u[n-M+1]\}$, and $e_o[n] \perp \mathcal{U}_n$.

From the vector orthogonality condition

$$\mathbb{E}[\mathbf{u}[n]e_o^*[n]] = \mathbf{0}$$

multiplying on the both side by $\mathbf{w}_o^H$ yields

$$\mathbb{E}[\mathbf{w}_o^H\mathbf{u}[n]e_o^*[n]] = 0$$

Since $\mathbf{w}_o^H\mathbf{u}[n] = y_o[n]$, we obtain

$$\mathbb{E}[y_o[n]e_o^*[n]] = 0$$

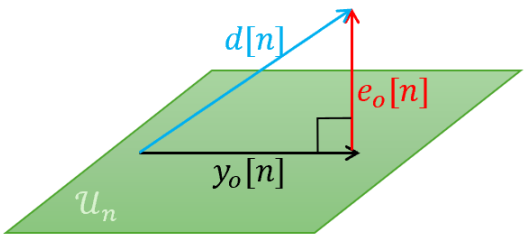


Figure 7: Orthogonality principle in the Wiener filter

Thus the minimum cost can be written as

$$J_{\min} = \mathbb{E}[|e_o[n]|^2] = \mathbb{E}[|d[n]|^2] - \mathbb{E}[|y_o[n]|^2]$$

i.e., the optimal error power equals the difference between the desired-response power and the output power.

4.6 Offline

The Wiener filter provides an optimal linear estimator of the desired signal under the MMSE criterion. In this offline setting, we assume that multiple input–output pairs are available. The input vector is

$$\mathbf{u}[n] = [u[n], u[n-1], \ldots, u[n-M+1]]^T$$

and the corresponding desired response is $d[n]$, which depends not only on the current sample $u[n]$ but also on the past $M-1$ inputs. From these data, we compute the correlation matrix $\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$ and the cross-correlation vector $\mathbf{p} = \mathbb{E}[\mathbf{u}[n]d^*[n]]$. With these, the optimal weight vector is obtained as

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}.$$

Once the coefficients are determined, they remain fixed. Given a new input vector $\mathbf{u}[n]$ while the true $d[n]$ is unknown, the filter produces an estimate

$$\hat{d}[n] = \mathbf{w}_o^H \mathbf{u}[n],$$

which serves as the best linear MMSE approximation of the desired signal. Here the index n should be interpreted carefully: during the offline computation it enumerates the available data samples, whereas in application it denotes discrete time.

In summary, the Wiener filter in its classical form is an offline solution: the coefficients are first computed from available data and then applied unchanged. In practice, however, the true statistics $\mathbf{R}$ and $\mathbf{p}$ are rarely available in closed form or may vary with time. This limitation motivates iterative numerical methods, such as steepest descent, and eventually adaptive filters that update the coefficients online directly from the data.

5 Method of Steepest Descent

The Wiener solution is a one step procedure which involves a matrix inversion operation. Sometimes it is impractical to perform matrix inversion operation. So we desired to develop a recursive form of solution. To make the Wiener filter recursive, each weight become adjustable, i.e. **time-varying weight vector**

$$\mathbf{w}[\mathbf{n}] = \begin{bmatrix} w_0[n] \\ w_1[n] \\ \vdots \\ w_{M-1}[n] \end{bmatrix}$$

At time n, the filter produces an estimate

$$\hat{d}[n|\mathcal{U}_n] = \mathbf{w}^H[n]\mathbf{u}[n]$$

Hence, the time-varying cost function is

$$J(\mathbf{w}[n]) = \mathbb{E}\left[\left|d[n] - \hat{d}[n|\mathcal{U}_n]\right|^2\right] = \mathbf{w}^H[n]\mathbf{R}\mathbf{w}[n] - \mathbf{w}^H[n]\mathbf{p} - \mathbf{p}^H\mathbf{w}[n] + \sigma_d^2$$

We want to update $\mathbf{w}[n]$ with n so as the cost function to reach the bottom of quadratic surface where $(\mathbf{w}[n], J(\mathbf{w}[n])) = (\mathbf{w}_o, J_{min})$.

Appendix A: The ROC of One-Sided Z-Transform

For a one-sided z-transform:

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

We aim to show that the region of convergence (ROC) is:

$$\text{ROC} = \{z \in \mathbb{C} \mid |z| > \max(|p_i|)\}$$

where p_i are the poles of $H(z)$.

Since $H(z)$ is a rational function (i.e., a ratio of polynomials, corresponding to an LTI system described by a LCCDE), it can be decomposed via partial fraction expansion by the fundamental theorem of algebra.

$$H(z) = \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2} + \cdots + \frac{A_N}{z - p_N}$$

Taking the inverse z-transform, we obtain:

$$h[n] = A_1 p_1^n u[n] + A_2 p_2^n u[n] + \cdots + A_N p_N^n u[n]$$

Substituting into the z-transform definition:

$$H(z) = \sum_{n=0}^{\infty} h[n] z^{-n} = \sum_{n=0}^{\infty} \left(A_1 p_1^n z^{-n} + \cdots + A_N p_N^n z^{-n} \right)$$

Each term becomes a geometric series:

$$\sum_{n=0}^{\infty} A_i p_i^n z^{-n} = A_i \sum_{n=0}^{\infty} \left(\frac{p_i}{z} \right)^n$$

This converges if and only if:

$$\left| \frac{p_i}{z} \right| < 1 \quad \Leftrightarrow \quad |z| > |p_i|$$

Therefore, the entire $H(z)$ converges if all terms converge, i.e.:

$$|z| > \max(|p_i|)$$

Hence, the ROC for the one-sided z-transform is:

$$\boxed{\text{ROC} = \{z \in \mathbb{C} \mid |z| > \max(|p_i|)\}}$$

Appendix B: Positive Definiteness and Nonsingularity of R

In Section 3.5, we stated that in the presence of additive white Gaussian noise (AWGN):

$$\mathbf{u}[n] = \mathbf{s}[n] + \mathbf{v}[n] \quad \Rightarrow \quad \mathbf{R} = \mathbf{R}_s + \sigma_v^2 \mathbf{I}$$

where $\mathbf{R}$ is positive definite and nonsingular. This appendix provides a detailed derivation and geometric interpretation of this important property.

Definitions

For a Hermitian matrix $\mathbf{A} \in \mathbb{C}^{M \times M}$:

- $\mathbf{A}$ is **positive definite** ($\mathbf{A} \succ 0$) if $\mathbf{x}^H \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$
- $\mathbf{A}$ is **positive semi-definite** ($\mathbf{A} \succeq 0$) if $\mathbf{x}^H \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \neq \mathbf{0}$
- $\mathbf{A}$ is **nonsingular** if $\det(\mathbf{A}) \neq 0$ (equivalent statement is $\mathbf{A}^{-1}$ exists)

Note: The necessity of Hermitian A

If $\mathbf{A}$ is Hermitian, i.e. $\mathbf{A} = \mathbf{A}^H$, then for any complex vector $\mathbf{x}$, the scalar $\mathbf{x}^H \mathbf{A} \mathbf{x}$ is always real. Hence it makes sense to compare it with 0 in order to determine definiteness.

Indeed,

$$(\mathbf{x}^H \mathbf{A} \mathbf{x})^* = \mathbf{x}^* \mathbf{A}^* \mathbf{x}^T = \mathbf{x}^* \mathbf{A}^T \mathbf{x}^T \quad (\text{since } \mathbf{A} = \mathbf{A}^H \implies \mathbf{A}^* = \mathbf{A}^T).$$

Now note that $\mathbf{x}^* \mathbf{A}^T \mathbf{x}^T$ is a scalar (a 1×1 matrix), and any scalar equals its own transpose. Thus

$$\mathbf{x}^* \mathbf{A}^T \mathbf{x}^T = (\mathbf{x}^* \mathbf{A}^T \mathbf{x}^T)^T = \mathbf{x}^H \mathbf{A} \mathbf{x}.$$

Therefore $\mathbf{x}^H \mathbf{A} \mathbf{x}$ equals its own conjugate and must be real.

Key Relationship: Positive Definite ⇒ Nonsingular

For any Hermitian matrix $\mathbf{A}$, the following are equivalent:

1. $\mathbf{A}$ is positive definite
2. All eigenvalues of $\mathbf{A}$ are strictly positive: $\lambda_i > 0$ for all i
3. $\mathbf{A}$ is nonsingular and $\mathbf{A}^{-1}$ exists

Proof Chain:

$$\mathbf{A} \succ 0 \Leftrightarrow \text{All } \lambda_i > 0 \Leftrightarrow \det(\mathbf{A}) = \prod_{i=1}^{M} \lambda_i > 0 \Leftrightarrow \mathbf{A}^{-1} \text{ exists}$$

Geometric Interpretation

The relationship between positive definiteness and nonsingularity has a clear geometric meaning:

- **Eigenvalues as stretching factors:** Each eigenvalue λ_i represents how much the matrix stretches vectors along the corresponding eigenvector direction.
- **Zero eigenvalue = information loss:** If $\lambda_i = 0$ for some i, then any vector along the i-th eigenvector gets mapped to zero:

  $$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i = 0 \cdot \mathbf{v}_i = \mathbf{0}$$

  This means the transformation "flattens" the space, losing information irreversibly.
- **Positive eigenvalues = invertible transformation:** When all $\lambda_i > 0$, every direction gets stretched (not flattened), so the transformation preserves all information and can be reversed.
- **Volume interpretation:** The determinant $\det(\mathbf{A}) = \prod \lambda_i$ represents the volume scaling factor. If any $\lambda_i = 0$, the volume becomes zero (singular), meaning the transformation is not invertible.

Mathematical Derivation

Step 1: Signal Model Setup

Consider the signal model with additive white Gaussian noise:

$$\mathbf{u}[n] = \mathbf{s}[n] + \mathbf{v}[n]$$

where:

- $\mathbf{s}[n]$: desired random signal vector
- $\mathbf{v}[n]$: additive white Gaussian noise vector
- $\mathbb{E}[\mathbf{s}[n]\mathbf{v}^H[n]] = \mathbf{0}$ (signal and noise are uncorrelated)
- $\mathbb{E}[\mathbf{v}[n]\mathbf{v}^H[n]] = \sigma_v^2\mathbf{I}$ (white noise with variance σ_v^2)

Step 2: Derive the Correlation Matrix

The correlation matrix of the observed signal is:

$$\mathbf{R} = \mathbb{E}[\mathbf{u}[n]\mathbf{u}^H[n]]$$

Substituting the signal model:

$$\begin{aligned}\mathbf{R} &= \mathbb{E}[(\mathbf{s}[n] + \mathbf{v}[n])(\mathbf{s}[n] + \mathbf{v}[n])^H] \\ &= \mathbb{E}[\mathbf{s}[n]\mathbf{s}^H[n]] + \mathbb{E}[\mathbf{s}[n]\mathbf{v}^H[n]] + \mathbb{E}[\mathbf{v}[n]\mathbf{s}^H[n]] + \mathbb{E}[\mathbf{v}[n]\mathbf{v}^H[n]]\end{aligned}$$

Since $\mathbf{s}[n]$ and $\mathbf{v}[n]$ are uncorrelated, the cross terms vanish:

$$\mathbb{E}[\mathbf{s}[n]\mathbf{v}^H[n]] = \mathbb{E}[\mathbf{v}[n]\mathbf{s}^H[n]] = \mathbf{0}$$

Therefore:

$$\boxed{\mathbf{R} = \mathbf{R}_s + \sigma_v^2\mathbf{I}}$$

where:

- $\mathbf{R}_s = \mathbb{E}[\mathbf{s}[n]\mathbf{s}^H[n]]$: signal correlation matrix
- $\sigma_v^2\mathbf{I}$: noise contribution (scaled identity matrix)

Step 3: Prove Positive Definiteness

Signal correlation matrix $\mathbf{R}_s$:
By construction, $\mathbf{R}_s$ is always positive semi-definite. For any non-zero vector $\mathbf{x}$:

$$\mathbf{x}^H\mathbf{R}_s\mathbf{x} = \mathbf{x}^H\mathbb{E}[\mathbf{s}[n]\mathbf{s}^H[n]]\mathbf{x} = \mathbb{E}[|\mathbf{x}^H\mathbf{s}[n]|^2] \geq 0$$

Noise correlation matrix $\sigma_v^2\mathbf{I}$:
The noise term is strictly positive definite. For any non-zero vector $\mathbf{x}$:

$$\mathbf{x}^H(\sigma_v^2\mathbf{I})\mathbf{x} = \sigma_v^2\mathbf{x}^H\mathbf{x} = \sigma_v^2\|\mathbf{x}\|^2 > 0$$

since $\sigma_v^2 > 0$ and $\mathbf{x} \neq \mathbf{0}$.

Combined result:
For any non-zero vector $\mathbf{x}$:

$$\begin{aligned}\mathbf{x}^H\mathbf{R}\mathbf{x} &= \mathbf{x}^H(\mathbf{R}_s + \sigma_v^2\mathbf{I})\mathbf{x} \\ &= \mathbf{x}^H\mathbf{R}_s\mathbf{x} + \sigma_v^2\|\mathbf{x}\|^2 \\ &\geq 0 + \sigma_v^2\|\mathbf{x}\|^2 > 0\end{aligned}$$

Therefore: $\boxed{\mathbf{R} \succ 0}$

Physical Interpretation

Why Adding Noise Helps

Regularization Effect: Even if the signal correlation matrix $\mathbf{R}_s$ is singular (e.g., when signal components are perfectly correlated), adding white noise $\sigma_v^2\mathbf{I}$ "regularizes" the system by:

- Adding energy to all directions uniformly
- Ensuring no direction gets completely flattened to zero
- Guaranteeing the matrix remains invertible

Mathematical Intuition: The noise term $\sigma_v^2\mathbf{I}$ shifts all eigenvalues upward by σ_v^2:

$$\lambda_i(\mathbf{R}) = \lambda_i(\mathbf{R}_s) + \sigma_v^2 > 0$$

Clarification: The regularization effect comes from the isotropic nature of the noise covariance matrix $\sigma_v^2\mathbf{I}$, which uniformly injects energy in all directions. This is not a property unique to white noise — any additive noise with covariance proportional to the identity matrix (not just white, not necessarily Gaussian) would provide the same benefit. Therefore, while AWGN is commonly used in practice, the key property enabling regularization is isotropy.

Summary

In the presence of AWGN, the correlation matrix $\mathbf{R} = \mathbf{R}_s + \sigma_v^2\mathbf{I}$ is:

- **Positive definite**: $\mathbf{x}^H\mathbf{R}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$
- **Nonsingular**: $\det(\mathbf{R}) \neq 0$
- **Invertible**: $\mathbf{R}^{-1}$ exists and is unique

Key Insight: White noise acts as a natural regularizer, preventing matrix singularity and ensuring robust signal processing algorithms.