Notes of Statistical Digital Signal Processing and Modelling (TODO)

Gralerfics

December 18, 2025

Contents

1. Linear Algebra

(TODO) Solutions, rank, inverses, eigenvalues, quadratic forms and positive definiteness, matrix calculus, etc., of systems of linear equations.

2. Probability Theory

2.1. Random Variable

Certain phenomena (such as rolling dice) are difficult to predict due to their random nature; we refer to these as **random phenomena**. Random phenomena occur continuously. Each time we observe and record their outcome, it is called a **random test**, and the recorded result is called a **sample**. A single random test may yield various different outcomes (samples). The collection of all such samples is called the sample space of that random phenomenon, encompassing all possible outcomes that may occur in that random event. A **random event** is a subset of the sample space that contains a portion of the possible outcomes (samples). If the outcome (sample) ω obtained from a single random trial belongs to the sample subspace included in a random event A, then the random event A is said to have occurred; otherwise, it is said not to have occurred.

A random event can be any statement, such as "It will rain tomorrow," which corresponds to the sample space "Whether it rains tomorrow or not." For this random phenomenon, the sample space for whether it rains tomorrow is $\{\text{rain}, \text{no rain}\}$, and "it rains tomorrow" is its subset $\{\text{rain}\}$. Similarly, "the result of rolling a fair 20-sided die is greater than 18" is a random event, and the sample space for rolling a fair 20-sided die is $\{1, 2, \ldots, 20\}$. This random event can be denoted as $\{19, 20\}$ as a random event of this random phenomenon. The sample space of the random phenomenon is $\{1, 2, \ldots, 20\}$, and this random event can be denoted as $\{19, 20\}$. A **random variable** x is used to describe the outcome of a random phenomenon whose results can be quantified. For example, the outcome of rolling a twenty-sided die yields one of twenty integers from 1 to 20. Even sample spaces that do not explicitly involve quantities, such as "whether it will rain tomorrow," can be quantified by arbitrarily defining "rain" as 1 and "no rain" as 0.

> In most lecture notes, random variables are denoted by uppercase letters, but this is merely a difference in notation. To maintain consistency with the notation used in subsequent discussions of stochastic processes, lowercase letters will be used here to represent random variables, with explicit notation indicating their randomness when necessary. Correspondingly, parameters and other independent variables (such as the parameters in the probability density function discussed later) will be denoted by Greek letters like α to avoid confusion.

The probability of a random event A occurring is denoted as $\Pr(A)$, representing the probability of the event happening. The probability of a random event encompassing the entire sample space is 1, since the outcome of the random experiment is necessarily included within it, signifying that the event must occur. Further details regarding probability, conditional probability, total probability, and the sample space set will not be elaborated upon here.

2.2. Probability Density Function (PDF)

The different values of a random variable have different probabilities. For a discrete random variable (one with a finite number of possible values), a mapping that assigns each possible value to its corresponding probability is called the probability mass function (PMF). This function contains all the information about the distribution.

If we consider continuous random variables, where the number of possible values is infinite, the probability of obtaining any specific value is approximately 0. Thus, we can no longer construct a function by assigning probability values to each specific outcome as we do with a PMF. We first define the Cumulative Distribution Function (CDF) for the random variable x:

$$F_x(\alpha) = \Pr\{x \leq \alpha\} \tag{1}$$

Whether curly braces or parentheses are used is just notation and does not change the meaning. In short, it represents the probability that x is less than or equal to the parameter α. Clearly, for a continuous random variable x taking real values, the CDF tends toward 0 as α approaches negative infinity, and toward 1 as α approaches positive infinity. We define the Probability Density Function (PDF) via the CDF:

$$f_x(\alpha) = \frac{\mathrm{d}}{\mathrm{d}\alpha} F_x(\alpha) \tag{2}$$

That is, the PDF is the derivative of the CDF. According to the Fundamental Theorem of Calculus, the integral of the PDF from α_1 to α_2 (i.e., the area under the curve over this interval) equals the probability that the random variable x falls within this interval:

$$\begin{aligned} \int_{\alpha_1}^{\alpha_2} f_x(\alpha)\,\mathrm{d}\alpha &= F_x(\alpha_2) - F_x(\alpha_1) \\ &= \Pr\{x \leq \alpha_2\} - \Pr\{x \leq \alpha_1\} \\ &= \Pr\{\alpha_1 < x \leq \alpha_2\} \end{aligned} \tag{3}$$

Similarly, the PDF contains all the information regarding the distribution of a random variable.

2.3. Joint Distribution

A joint distribution describes the synergistic distributional relationship between multiple random variables. For example, for two random variables x and y, the joint cumulative distribution function is defined as:

$$F_{x,y}(\alpha, \beta) = \Pr\{x \leq \alpha, y \leq \beta\} \tag{4}$$

This is straightforward: it simply places constraints on both random variables simultaneously. Its joint probability density function is:

$$f_{x,y}(\alpha, \beta) = \frac{\partial^2}{\partial\alpha\partial\beta} F_{x,y}(\alpha, \beta) \tag{5}$$

2.4. Mathematical Expectation

Mathematical expectation can be thought of as the average value obtained after performing an infinite number of random trials. By definition, it is the weighted average of all possible outcomes using their respective probabilities. For example, the expectation of a continuous random variable x is:

$$E(x) = \int_{-\infty}^{\infty} \alpha f_x(\alpha)\, \mathrm{d}\alpha \tag{6}$$

If it is a discrete random variable, the definition is even clearer (α represents any possible value):

$$E(x) = \sum_{\alpha} \alpha f_x(\alpha) \tag{7}$$

This is simple, but let's look at an example that is not just the expectation of a single random variable. For instance, the variance of a random variable x is defined as $E\{(x - E(x))^2\}$, and its integral form is:

$$E\{(x - E(x))^2\} = \int_{-\infty}^{\infty} (\alpha - E(x))^2 f_x(\alpha)\, \mathrm{d}\alpha \tag{8}$$

Note that the distribution function used here is still $f_x(\alpha)$ rather than something like $f_x((\alpha - E(x))^2)$ or $f_{(x-E(x))^2}(\alpha)$. What I want to emphasize is that we should add a subscript to E to indicate which random variable's distribution the expectation is being calculated over:

$$E_x(x) = \int_{-\infty}^{\infty} \alpha f_x(\alpha)\, \mathrm{d}\alpha \tag{9}$$

This is because calculating an expectation requires two elements: first, a distribution (the averaging process occurs over this distribution); and second, an expression (the content inside the parentheses). In the defining formula for expectation, the x beneath the distribution function $f_x(\alpha)$ comes from the x in E_x, representing the first element (the distribution), rather than the x in the expression within $E(x)$. The α multiplied before the distribution function is the result of substituting the parameter α into the x of the expression, as shown in example Equation 10.

This is actually a simple matter, but it is easy to get stuck here if one isn't careful, as the description of the distribution in the expectation symbol is often omitted, defaulting to the distribution of the random variables present in the expression. This is mentioned here merely for clarification.

In cases involving multiple random variables, it is even more important to clarify which distribution the expectation is being taken over. For example, for the expectation of an expression involving two random variables, the integral must be changed to a double integral to traverse all combinations of values, and the distribution function must be changed to the joint probability density function:

$$E_{x,y}(xy^*) = \int_{\alpha=-\infty}^{\infty} \int_{\beta=-\infty}^{\infty} \alpha\beta^* f_{x,y}(\alpha,\beta)\,\mathrm{d}\alpha\,\mathrm{d}\beta \tag{10}$$

2.5. Moments, Mean and Variance

Next are some statistical quantities based purely on definitions.

First, we define the ***k*-th order raw moment** of a random variable x as $E\big(x^k\big)$, where the first-order raw moment is the Mean of x, denoted as $m_x = E(x)$.

Next, we define the ***k*-th order central moment** as $E\big\{(x - E(x))^k\big\}$. It can be seen that the second-order central moment is the variance of x, denoted as $\sigma_x^2 = \mathrm{Var}(x)$, where σ_x is called the standard deviation (the square root of the variance).

Other statistical quantities and their corresponding physical meanings will not be detailed here.

2.6. Independence

Independence describes whether two random variables are mutually unaffected. If two random variables x and y are independent, it means the probability of x taking a value α multiplied by the probability of y taking a value β directly yields the probability of them taking those two values simultaneously. This is because the probability of independent events occurring together equals the product of their individual probabilities (refer to the multiplication principle).

Satisfying this condition for all possible values implies that the product of their individual distributions yields the joint distribution, which is the necessary and sufficient condition for independence:

$$f_{x,y}(\alpha,\beta) = f_x(\alpha) f_y(\beta) \tag{11}$$

2.7. Covariance Function

Covariance measures the degree of correlation between random variables. The covariance of two random variables x and y is defined as:

$$c_{xy} = \mathrm{Cov}(x,y) = E\Big\{(x - m_x)\big(y - m_y\big)^*\Big\} \tag{12}$$

Incidentally, note that $c_{xx} = E\Big\{(x - m_x)^2\Big\}$ is the variance of x.

Covariance can be thought of as a measure of the significance of the linear relationship between random variables. Specifically, based on the definition: when x is above its mean, to what extent does y tend to be simultaneously above (or below) its own mean? The greater this extent, the more it indicates that when x is larger, y also tends to be larger (or smaller), suggesting a stronger linear relationship. By subtracting the means, the influence of the absolute values is removed, focusing primarily on the relative trends of the variables.

2.8. Correlation Function

The Correlation Function is defined as the expectation of the inner product of two random variables:

$$r_{xy} = E\{xy^*\} \tag{13}$$

It can be viewed as the covariance function without the means removed. Their relationship can be derived as:

$$\begin{aligned} c_{xy} &= E\Big\{(x - m_x)\big(y - m_y\big)^*\Big\} \\ &= E\{xy^*\} + E\big\{m_x m_y^*\big\} - m_x E\{y^*\} - E\{x\}m_y^* \\ &= E\{xy^*\} + m_x m_y^* - m_x m_y^* - m_x m_y^* \\ &= r_{xy} - m_x m_y^* \end{aligned} \tag{14}$$

They differ only by a constant $m_x m_y^*$, so the physical meaning they express can be considered similar.

2.9. Correlation Coefficient and Orthogonality

Of course, covariance does not remove the units (dimension), so it lacks universality across different combinations of random variables. We typically use normalization to define the correlation coefficient to measure Correlation, such as the widely used Pearson correlation coefficient:

$$\rho_{xy} = \frac{c_{xy}}{\sigma_x \sigma_y} \tag{15}$$

It can be proven that its value ranges within $[-1, 1]$, providing a unified measure of the significance of the linear relationship between random variables. If this correlation coefficient is zero, the two random distributions are said to be Orthogonal.

When the correlation coefficient is 0 (i.e., $c_{xy} = 0$), the two random variables are uncorrelated. A positive correlation coefficient represents positive correlation, and a negative one represents negative correlation. The necessary and sufficient condition for being uncorrelated is:

$$c_{xy} = 0 \quad \Rightarrow \quad r_{xy} - m_x m_y^* = 0 \quad \Rightarrow \quad E\{xy^*\} = E\{x\}E^*\{y\} \tag{16}$$

That is, the expectation of the product of the random variables equals the product of their individual expectations.

Note that correlation and independence are not the same thing.

First, being uncorrelated does not necessarily mean independence. For example, if two random variables satisfy $y = x^2$, they are completely dependent, yet the correlation coefficient is still 0. As mentioned, correlation mainly measures the

significance of a linear relationship. In this case, when x is negative, y decreases as x increases; when x is positive, y increases as x increases. These two parts of the relationship cancel each other out symmetrically, resulting in no linear component, even though a quadratic relationship exists.

However, independence does imply that variables are uncorrelated. Since the distributions are entirely unrelated, the joint density is simply the product of the individual densities, and by definition, $E\{xy^*\} = E\{x\}E^*\{y\}$ follows.

In summary: Independence $\Rightarrow$ Uncorrelated, but the reverse is not necessarily true.

3. Random Process

A random process $\{x(n)\}$ is essentially a sequence composed of random variables.

These random variables can be independent and identically distributed (i.i.d.), such as white noise, but they often are not; in the real world, most are not. The index of this sequence can represent time, or it can represent something else. For convenience, we will assume we are studying time series, where the index n represents different points in time.

As before, the focus of studying randomness is on finding the "invariants" within the change, such as statistical characteristics. Therefore, we begin by defining a series of statistical features.

3.1. Statistic

3.1.1. Mean

A random process does not necessarily have a single uniform mean; the mean is still defined for each individual random variable, resulting in a sequence:

$$m_x(n) = E\{x(n)\} \tag{17}$$

3.1.2. Auto-correlation

For a random process $x(n)$, the auto-correlation function is the correlation function of two random variables at specified indices k and l:

$$r_x(k,l) = r_{x(k),x(l)} = E\{x(k)x^*(l)\} \tag{18}$$

From this formula, each r_x is only related to two specific random variables and has no immediate connection to the process as a whole. However, this changes when we introduce Wide-Sense Stationarity (WSS).

3.1.3. Auto-covariance

Analogous to covariance, we define auto-covariance:

$$c_x(k,l) = E\{[x(k) - m_x(k)][x(l) - m_x(l)]^*\} = r_x(k,l) - m_x(k)m_x^*(l) \tag{19}$$

3.1.4. Cross-correlation

Cross-correlation involves two random processes. For example, the cross-correlation of $\{x(n)\}$ and $\{y(n)\}$ is defined as:

$$r_{xy}(k,l) = r_{x(k),y(l)} = E\{x(k)y^*(l)\} \tag{20}$$

3.1.5. Cross-covariance

Similarly, cross-covariance is defined as:

$$c_{xy}(k,l) = E\Big\{[x(k) - m_x(k)]\big[y(l) - m_y(l)\big]^*\Big\} = r_{xy}(k,l) - m_x(k)m_y^*(l) \tag{21}$$

These definitions are listed here for completeness, though they will not be used extensively later.

3.2. Structural Invariance

Notice that the statistics defined previously focus on discrete random variables and do not strongly relate to the overall random process. Next, we will focus on some structural invariants of the entire random process. These properties provide a rational basis for estimating distributions when the true distribution is unknown and only finite samples are available.

Specifically, while random variables are idealizations described by a PDF, we usually study actual signals without knowing their true distribution. We must rely on samples to estimate the distribution. We can view a specific signal $x[n]$ as a single Realization of a random process. Each sample at a given point in time is a single observation of the corresponding random variable in the process. Figure 1 shows a series of possible realizations obtained after many trials of a random process, with one highlighted.

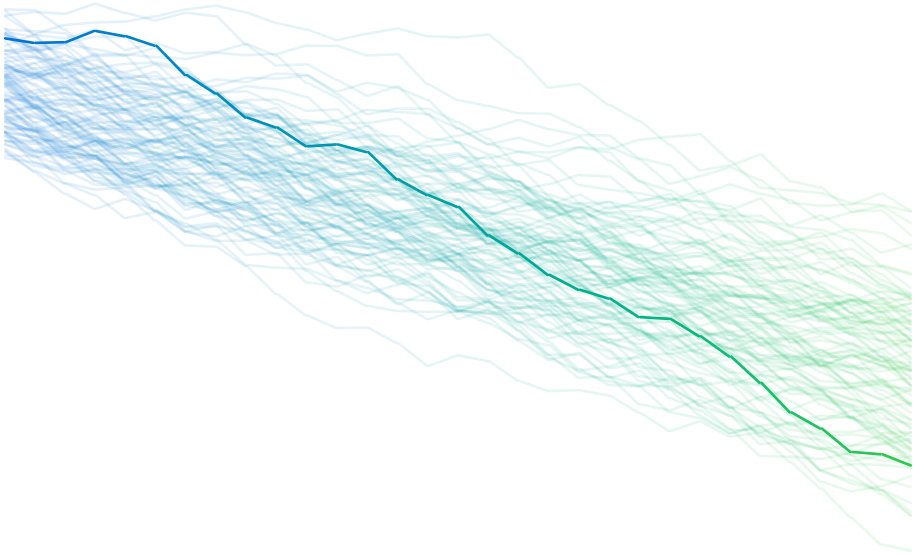

Figure 1: Different realizations of a random process

3.2.1. Stationarity

Our **first question** is: In a random process, do certain statistical features change over time?

We introduce the concept of Stationarity. The main idea is to measure whether the statistical characteristics of any sub-sequence remain unchanged after being delayed by an arbitrary amount of time. To have identical statistical features, the simplest way is to require the distributions to be identical, which defines Strict-Sense Stationarity (SSS):

$$\forall k, \{n_1, n_2, ..., n_m\}, f_{x(n_1),x(n_2),...,x(n_m)}(\cdot) = f_{x(n_1+k),x(n_2+k),...,x(n_m+k)}(\cdot) \tag{22}$$

The requirements for SSS are too high and usually unnecessary. Therefore, we define Wide-Sense Stationarity (WSS), focusing only on the consistency of first and second-order statistics:

1. $$m_x(n) = m_x \tag{23}$$
2. $$r_x(k, l) = r_x(k - l) \tag{24}$$
3. $$c_x(0) < \infty \tag{25}$$

That is, for a WSS process: the mean is independent of time, and the auto-correlation function depends only on the time difference, not on absolute time.

The third condition—finite variance—is often omitted in many texts.

Finite variance is equivalent to a finite mean-square value (differing only by the square of the mean). Physically, this represents finite power. Engineers often ignore this condition because most physical processes have finite power.

Mathematically, this ensures the existence of the second moment. However, since the second condition requires $r_x(0)$ to exist, and the existence of an expectation implies convergence to a finite value, the third condition is effectively implied.

This allows us to take sub-sequences from a single realization at different times and claim their average properties are consistent. For a stationary signal, we can split a sufficiently long sample into multiple segments and assert that these segments share the same underlying statistical features.

3.2.2. Ergodicity

The stationarity derived from the first question allows us to use segments of a long sample as different samples. Our **second question** is: Can a single, sufficiently long realization represent the entire ensemble of infinite possible realizations (the sample space)? If I split this long sample to estimate the distribution, is the result the true distribution of the random process? Can one realization represent all aspects of the process?

We use Ergodicity to describe this property. It is built upon stationarity; we must ensure statistical features don't change over time, otherwise, different segments would have different properties, making a long sample useless.

First, we need a typical example of "stationary but non-ergodic" to show why this concept is necessary. Let the random process $x(n)$ be such that $x(0)$ is a random variable z drawn once, and thereafter $x(n) = x(n-1)$. This is called a "random constant" because every realization is a constant signal, but the constant value differs between realizations depending on the initial draw.

Analyzing this process: first, it is definitely stationary because a shift in time clearly does not affect the distribution. We can also prove it satisfies WSS: the mean $m_x = E\{x(0)\}$ is independent of n; the auto-correlation $r_x(k,l) = E\{x(k)x^*(l)\} = E\{x(0)x^*(0)\} = \sigma_x^2$ is independent of time.

Now consider ergodicity. The ensemble mean is $E\{z\}$, but for a long realization, the time average is whatever value $x(0)$ happened to be for that specific trial. No matter how long the sample is, the average remains that value, which may not equal $E\{z\}$. In other words, we cannot estimate the properties of z using only one realization, even if it is infinitely long.

In summary, stationarity ensures that a sufficiently long sample yields a meaningful time average, while ergodicity ensures that this time average equals the ensemble average (mean of all possible samples). Thus, estimates derived this way are correct.

In physics and statistical mechanics, this concept is often translated as "各态历经性" (all-state traversing), which is more descriptive. It suggests that as time progresses, the system will eventually experience and exhibit all its possible macroscopic states. Only when a single realization has the potential to manifest all features of the distribution can we estimate the macroscopic distribution from a single sample.

We can also look at this from the perspective of biased and unbiased estimation. For a stationary signal, we can estimate a meaningful statistic, but we cannot guarantee if it is unbiased; ergodicity guarantees that it is.

The strict definition of ergodicity is abstract and complex. Like WSS, we only consider ergodicity in terms of the mean and correlation for our needs. Of course, the following discussion assumes at least WSS.

First is **Mean-Ergodicity**. We define the sample mean, which is the temporal average of a single realization:

$$\hat{m}_x^{(N)} = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \tag{26}$$

Considering the original definition, the time average (sample mean) equals the ensemble average:

$$\lim_{N\to\infty} E\left\{\left|\hat{m}_x^{(N)} - m_x\right|^2\right\} = 0 \tag{27}$$

Since we cannot truly verify this from the real distribution, we provide a **necessary and sufficient** condition for mean ergodicity (proof omitted):

$$\lim_{N\to\infty} \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} c_x(n-m) = 0 \tag{28}$$

That is, the auto-covariance function decays fast enough. An intuitive interpretation is that the correlation between random variables does not persist too long in time; a long-term average can "dilute" the dependencies.

There is also a **sufficient** condition for quick judgment in simple cases:

$$c_x(0) < \infty \quad \text{and} \quad \lim_{k\to\infty} c_x(k) = 0 \tag{29}$$

Or an even stronger condition: the absolute summability of the auto-covariance function is a **sufficient** condition for mean ergodicity:

$$\sum_{k=-\infty}^{\infty} |c_x(k)| < \infty \tag{30}$$

Next is **Correlation-Ergodicity**. We similarly define the sample auto-correlation function. Since the auto-correlation of a WSS process only depends on the time difference, all pairs in the sample with a time difference k can be used to estimate the k-th term of the auto-correlation function. we sum them and take the average:

$$\hat{r}_x^{(N)}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n+k]x^*[n] \tag{31}$$

You might notice that in this definition, a total of $N-k$ pairs were used, yet we divide by N rather than $N-k$.

This actually makes the estimate biased. However, if we only consider this for the definition of correlation ergodicity where N tends to infinity, the effect of k becomes negligible. That is, for this definition, we only require asymptotic unbiasedness.

In Section 9.1.1, we will revisit the use of this sample auto-correlation function for actual estimation and the explanation for dividing by N instead of $N-k$.

Similarly, the definition is that the time average equals the ensemble average:

$$\lim_{N\to\infty} E\left\{ \left| \hat{r}_x^{(N)}(k) - r_x(k) \right|^2 \right\} = 0 \tag{32}$$

We provide the practical **necessary and sufficient** condition without proof:

$$\forall k, \lim_{N\to\infty} \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \left| r_x(n-m) - r_x(n-m+k) \right|^2 = 0 \tag{33}$$

Additionally, the absolute summability condition Equation 30 is also a **sufficient** condition for correlation ergodicity.

3.3. Power Spectrum

(TODO) Add content regarding energy signals, power signals, and Power Spectral Density (PSD).

According to the Wiener–Khinchin Theorem, the PSD of a WSS process is the Fourier transform of its auto-correlation function.

3.4. Filtering Random Processes

(TODO)

If the filter coefficients $h[n]$ are finite-length and zero outside $[0, N-1]$, the power of the output process can be expressed using the auto-correlation matrix of the input $x(n)$ and the filter coefficient vector:

$$\sigma_y^2 = E\{|y(n)|^2\} = \boldsymbol{h}^H \boldsymbol{R}_x \boldsymbol{h} \tag{34}$$

3.5. Random Process and Digital Signals

Our random process $x(n)$ is a distribution, while our digital signal $x[n]$ can be seen as a realization sampled from it. We have defined characteristics like mean and auto-correlation for random processes, and we can perform operations like mean and auto-correlation on digital signals. We now want to examine the differences and connections between the two.

Clarifying this allows us to understand how to use samples to estimate distribution features, which has been our consistent goal. We will explore this from several common perspectives.

3.5.1. Auto-correlation Estimation with Multiple Realizations

If we have a large number of sampling results (realizations) of a random process, we can directly estimate its distribution features, such as the auto-correlation function, using the definition (Equation 18).

Suppose we have N realizations of the process $\{x(n)\}$, each of length M. The result of the i -th realization is denoted as $x_i[n]$, or as a vector (where M can be finite or infinite; here it is finite for demonstration):

$$\boldsymbol{x}_i = \begin{pmatrix} x_i[0] \\ x_i[1] \\ \vdots \\ x_i[M-1] \end{pmatrix} \tag{35}$$

We arrange the data from multiple samplings into a matrix:

$$X = [\boldsymbol{x}_0 \quad \boldsymbol{x}_1 \quad \dots \quad \boldsymbol{x}_{N-1}] = \begin{bmatrix} x_0[0] & x_1[0] & \dots & x_{N-1}[0] \\ x_0[1] & x_1[1] & \dots & x_{N-1}[1] \\ \vdots & \vdots & \ddots & \vdots \\ x_0[M-1] & x_1[M-1] & \dots & x_{N-1}[M-1] \end{bmatrix} \tag{36}$$

By definition, $r_x(k,l) = E\{x(k)x^*(l)\}$. Since E is the expectation over the ensemble, we simply use $x_i[k]$ and $x_i[l]$ from different realizations to estimate it:

$$\hat{r}_x(k,l) = \frac{1}{N}\sum_{i=0}^{N-1} x_i[k]x_i^*[l] \tag{37}$$

We write the estimated auto-correlation values as a matrix, which can be computed from X, leading to a concise matrix expression:

$$
\begin{aligned}
\hat{\boldsymbol{R}}_x &:= \begin{bmatrix} \hat{r}_x(0,0) & \hat{r}_x(0,1) & \dots & \hat{r}_x(0,M-1) \\ \hat{r}_x(1,0) & \hat{r}_x(1,1) & \dots & \hat{r}_x(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_x(M-1,0) & \hat{r}_x(M-1,1) & \dots & \hat{r}_x(M-1,M-1) \end{bmatrix} \\
&= \frac{1}{N}\sum_{i=0}^{N-1} \begin{bmatrix} x_i[0]x_i^*[0] & x_i[0]x_i^*[1] & \dots & x_i[0]x_i^*[M-1] \\ x_i[1]x_i^*[0] & x_i[1]x_i^*[1] & \dots & x_i[1]x_i^*[M-1] \\ \vdots & \vdots & \ddots & \vdots \\ x_i[M-1]x_i^*[0] & x_i[M-1]x_i^*[1] & \dots & x_i[M-1]x_i^*[M-1] \end{bmatrix} \\
&= \begin{bmatrix} x_0[0] & x_1[0] & \dots & x_{N-1}[0] \\ x_0[1] & x_1[1] & \dots & x_{N-1}[1] \\ \vdots & \vdots & \ddots & \vdots \\ x_0[M-1] & x_1[M-1] & \dots & x_{N-1}[M-1] \end{bmatrix} \begin{bmatrix} x_0[0] & x_0[1] & \dots & x_0[M-1] \\ x_1[0] & x_1[1] & \dots & x_1[M-1] \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1}[0] & x_{N-1}[1] & \dots & x_{N-1}[M-1] \end{bmatrix}^* \\
&= \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^H
\end{aligned}
\tag{38}
$$

(TODO) In cases where there are not many independent realizations, define $\boldsymbol{x}_i$ as $[x[i], ..., x[i+L-1]]^T$. Refer to Slides Lec11 P6.

3.5.2. Auto-correlation Estimation with Correlation-Ergodicity

As seen in the previous section, the larger the number of realizations N, the more accurate the estimate. However, if we have only one sample ($N = 1$), the estimate using that method will be extremely imprecise. In this case, if the random process is ergodic, we are allowed to use information from different times within that single sample to perform the estimation.

We define the auto-correlation function of a finite-length digital signal $x[n]$ (length N, note that N here is not the number of realizations above) as:

$$
R_{xx}(k) = \sum_{n=0}^{N-1-k} x[n+k]x^*[n] \tag{39}
$$

From a formulaic perspective, this can be understood as a measure of the similarity of a signal to itself at different time delays.

In practice, any definition expressing this meaning can be called auto-correlation; there are many variations with minor differences, such as whether to divide by the number of samples (taking an average) or whether to use delay k or lead k (which just flips the result). Use whichever is convenient.

We choose this definition here because the MATLAB cross-correlation function `xcorr` is defined this way (from the documentation):

$$
\hat{R}_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x_{n+m}y_n^*, & m \geq 0, \\ \hat{R}_{yx}^*(-m), & m < 0. \end{cases} \tag{40}
$$

Recall the sample auto-correlation function $\hat{r}_x^{(N)}(k)$ defined in Equation 31; its form is almost identical to the signal auto-correlation function $R_{xx}(k)$ we defined, differing only by a coefficient.

According to previous definitions, if the process is correlation-ergodic, then this sample auto-correlation function $\hat{r}_x^{(N)}(k)$ can be used to correctly estimate the auto-correlation function $r_x(k)$ of the random distribution. Consequently, the signal cross-correlation function $\hat{R}_{xy}(m)$, which has a nearly identical form, carries the same physical meaning and can also be used to estimate $r_x(k)$ (with appropriate coefficient adjustments).

In summary, the existence of correlation ergodicity allows us to use the signal auto-correlation of a sample to estimate the auto-correlation characteristics of the random process when the number of realizations is insufficient.

3.5.3. Auto-correlation Estimation (Comprehensive)

If multiple realizations (samples) are available and ergodicity holds, we can combine both advantages for estimation.

Specifically, in Section 3.5.1, we estimated $r_x(k, l)$. If ergodicity holds (which implies stationarity), then the auto-correlation function depends only on the time difference. We can then use all $\hat{r}_x(n, n+k)$ values (those located on the same diagonal in $\hat{\boldsymbol{R}}_x$) to estimate $r_x(k)$.

4. Digital Signal Processing

(TODO) Main content: DTFT, z-transform, frequency domain characteristics, stability, power, energy, etc.

4.1. Spectral Analysis

(TODO) Spectra of non-periodic signals are continuous, while those of discrete signals are periodic; Dirichlet kernel, windowing, DFT; zero-padding for increased density; zero-crossing and resolution; harmonic height and resolution, other window types; WSS random signals direct transform, averaged periodogram, BPSK example converging to the Dirichlet kernel.

For a long signal that does not satisfy the stationarity assumption, analyzing its spectrum over all time is not very meaningful because it changes over time. Therefore, we usually cut the signal into small segments for analysis. The method of cutting is direct truncation, assuming function values outside the segment are 0. We define the Dirichlet kernel function:

$$w_R[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{41}$$

The truncated signal is:

$$x_N[n] = x[n] w_R[n] \tag{42}$$

Multiplication in the time domain corresponds to convolution in the frequency domain. Specifically:

$$X_N(\omega) = \frac{1}{2\pi} \{X * W_R\}(\omega) \tag{43}$$

Examining this operation from a graphical and intuitive perspective:

(TODO) Spectrum of W_R, frequency shifting, the intuition of replacing an impulse function with a peak of a certain width.

5. Optimization

(TODO) Mainly regarding Least Squares, Lagrange Multipliers, etc.

(TODO) Need to write about splitting complex variables into themselves and their conjugates, and the reason for taking partial derivatives with respect to the conjugate during solving.

6. Signal Modelling

Actually, modelling can be seen as a process of signal compression. A parameterized model uses a smaller number of parameters than the number of signal samples to represent complex signals, achieving more efficient storage and transmission.

The parameters obtained from compression can also be regarded as descriptions of the essential characteristics of the signal, such as the physical laws underlying it. This allows us to use the model to perform prediction or extrapolation on the unknown parts of the signal.

Now, our goal is to model a given digital signal $x[n]$, which means finding a model $H(z)$ such that its output signal $\hat{x}[n]$ is as close as possible to the target signal $x[n]$.

6.1. Autoregressive and Moving Average (ARMA) Model

We can use different types of models to represent signals according to practical situations. Here we take the Autoregressive and Moving Average (ARMA) model, specifically $\mathrm{ARMA}(p, q)$, which is commonly used in time series analysis, as an example. Its transfer function is defined as follows:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{\sum_{k=0}^{q} b[k] z^{-k}}{1 + \sum_{k=1}^{p} a[k] z^{-k}} = \frac{B(z)}{A(z)} \tag{44}$$

Let the input and output signals be $x[n]$ and $y[n]$, with $X(z)$ and $Y(z)$ as their corresponding Z-transform functions. Although the index of $a[n]$ starts from 1 here formally, we can actually take $a[0] = 1$ to obtain a more unified form. Thus, by definition, we have $Y(z)A(z) = X(z)B(z)$, which transforms into the time domain as:

$$a[n] * y[n] = b[n] * x[n] \tag{45}$$

Expanding this yields the classic form of the Linear Constant Coefficient Difference Equation (LCCDE):

$$y[n] + \sum_{k=1}^{p} a[k] y[n-k] = \sum_{k=0}^{q} b[k] x[n-k] \tag{46}$$

Written in detail:

$$\begin{aligned} &y[n] + a[1]y[n-1] + \ldots + a[p]y[n-p] \\ &= b[0]x[n] + b[1]x[n-1] + \ldots + b[q]x[n-q] \end{aligned} \tag{47}$$

Obviously, this system is a typical Linear Shift-Invariant (LSI) system with favorable properties.

6.1.1. Autoregressive (AR) Model

If there is only the autoregressive part, i.e., $\mathrm{AR}(p) = \mathrm{ARMA}(p, 0)$:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b[0]}{1 + \sum_{k=1}^{p} a[k] z^{-k}} \tag{48}$$

In the time domain:

$$y[n] + \sum_{k=1}^{p} a[k]y[n-k] = b[0]x[n] \tag{49}$$

Written in detail:

$$a[0]y[n] + a[1]y[n-1] + ... + a[p]y[n-p] = b[0]x[n] \tag{50}$$

This model considers the output at the current time $y[n]$ to be a linear combination of the previous p outputs $y[n-1], ..., y[n-p]$ and the current input $x[n]$, hence it is called an autoregressive model. Since this model has no zeros, it is also known as an All-Pole Model.

6.1.2. Moving Average (MA) Model

If there is only the moving average part, i.e., $\mathrm{MA}(q) = \mathrm{ARMA}(0, q)$:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{q} b[k]z^{-k} \tag{51}$$

In the time domain:

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] \tag{52}$$

Written in detail:

$$y[n] = b[0]x[n] + b[1]x[n-1] + ... + b[q]x[n-q] \tag{53}$$

This model considers the output at the current time $y[n]$ to be a linear combination of the previous q inputs $x[n], x[n-1], ..., x[n-q]$, hence it is called a moving average model. Since this model has no poles, it is also known as an All-Zero Model.

6.2. Signal Models

As a discrete-time system, $H(z)$ cannot directly represent a signal; it needs an input to generate an output. We treat the output signal as the model's estimate of the target signal $\hat{x}[n]$, and encapsulate the input signal $x[n]$ as part of the model.

6.2.1. Deterministic Modelling

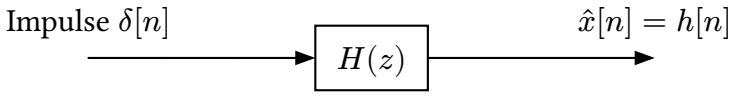

Figure 2: Signal model with deterministic input

We can choose a known, deterministic signal and fix it as the system input to stably obtain the desired output signal $\hat{x}[n]$, making its values as close as possible to the target signal $x[n]$. This is used for modeling deterministic signals and is called Deterministic Modelling.

This input signal can be chosen according to the practical situation; an input signal that matches the characteristics of the target signal can sometimes reduce the burden of model

fitting. Here we can choose to use the simplest unit impulse signal $\delta[n]$ as the input signal, which makes the system output $\hat{x}[n]$ the unit impulse response $h[n]$ of the system.

6.2.2. Stochastic Modelling

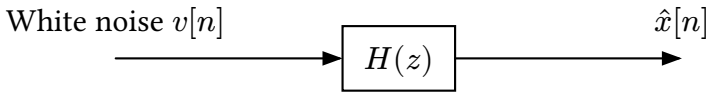

Figure 3: Signal model with stochastic input

We can also choose to use random noise with a known distribution as the input, obtaining an output signal $\hat{x}[n]$ whose statistical characteristics (such as mean and autocorrelation function) match those of the target signal $x[n]$. This is used for modeling stochastic processes and is called Stochastic Modelling.

We can choose to use white noise $v[n]$ with zero mean and variance σ_v^2 as the input signal. The basis for this is that its autocorrelation function is $r_v[k] = \sigma_v^2\delta[k]$, and its Fourier transform yields a constant power spectral density $P_v(\omega) = \sigma_v^2$, meaning it has a uniform energy distribution across all frequencies.

This characteristic ensures that we can obtain an output signal $\hat{x}[n]$ with any frequency characteristic by filtering it. Meanwhile, the uniform energy across all frequency components ensures that the statistical characteristics of the output signal are independent of the input signal.

7. Deterministic Modelling Identification

After the model is established, we still need to perform parameter identification, which means determining the values of the parameter sequences $a[k]$ and $b[k]$. The goal of selecting parameters is to make the model output $\hat{x}[n]$ as close as possible to the target signal $x[n]$.

7.1. Least Squares (LS) Method

First, let's discuss Deterministic Modelling. We hope that the model output $\hat{x}[n]$ can accurately reproduce the target signal $x[n]$, meaning the value of each sampling point should be as close as possible. Define the error signal $e'[n] = x[n] - \hat{x}[n]$. The model parameters can be determined by minimizing the mean square error $\mathcal{E}_{\text{LS}} = \sum_{n=0}^{\infty} |e'[n]|^2$, as shown in Figure 4.

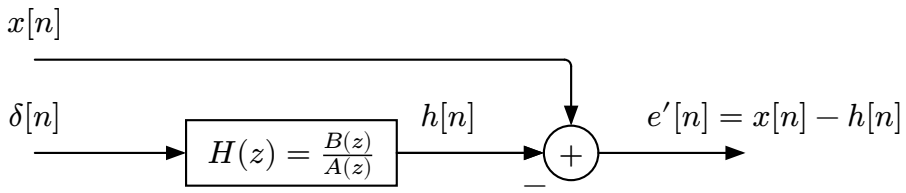

Figure 4: Diagram for deterministic model identification (intractable)

This optimization problem can be solved by solving the following system of equations (for the reason of taking partial derivatives with respect to the conjugate of the variables, see Section 5):

$$\begin{cases} \frac{\partial \mathcal{E}_{\text{LS}}}{\partial a^*[k]} = 0, & k = 1, 2, ..., p \\ \frac{\partial \mathcal{E}_{\text{LS}}}{\partial b^*[k]} = 0, & k = 0, 1, ..., q \end{cases} \tag{54}$$

However, this system of equations is non-linear and very complex to solve.

7.2. Padé Approximation

Notice that the ARMA model we use has $p + q + 1$ parameters, which means it has $p + q + 1$ degrees of freedom. Therefore, theoretically, we can use it to perfectly fit the first $p + q + 1$ samples of the signal. Let's consider this task first.

We perform a transformation on the form of the transfer function:

$$H(z) = \frac{B(z)}{A(z)} \quad \Rightarrow \quad H(z)A(z) = B(z) \quad \Rightarrow \quad h[n] * a[n] = b[n] \tag{55}$$

Expanding the convolution gives:

$$h[n] + \sum_{k=1}^{p} a[k]h[n-k] = b[n] \tag{56}$$

For the unit impulse input $\delta[n]$, the system output $h[n]$ is our estimated result $\hat{x}[n]$. To completely fit the first $p + q + 1$ samples, directly substitute $h[n] = x[n], 0 \leq n \leq p + q$, to get:

$$x[n] + \sum_{k=1}^{p} a[k]x[n-k] = \begin{cases} b[n], & n = 0, 1, ..., q \\ 0, & n = q+1, q+2, ..., q+p \end{cases} \tag{57}$$

This is the linear system of equations we want, containing $a[\cdot]$, $b[\cdot]$, and the known constant $x[n]$.

The above transformation process, reflected in the system block diagram, is multiplying both paths by the denominator $A(z)$ of $H(z)$. That is, let the new target error be $E(z) = A(z)E'(z) = A(z)X(z) - B(z)$, as shown in Figure 5.

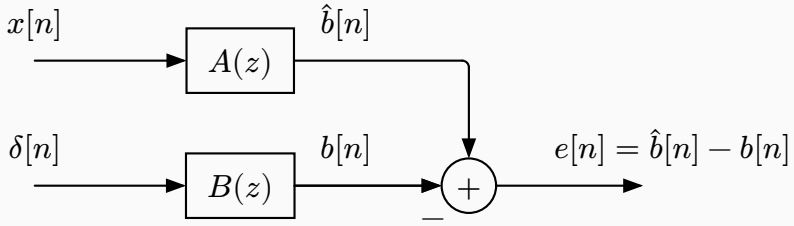

Figure 5: Diagram for deterministic model identification

The output of the unit impulse signal passing through $B(z)$ is $b[n]$. By subtracting it from $\hat{b}[n]$ (which can be viewed as an estimate of the coefficient sequence $b[n]$) obtained by passing the target signal $x[n]$ through $A(z)$, we get a new error.

The equations listed after such operations are linear.

Next is the solution to Equation 57. This system of equations contains the same number of equations and unknowns. If it is non-singular, it can be solved for a unique solution. To be clearer, let's draw the matrix:

$$\underbrace{\begin{bmatrix} x[0] & 0 & 0 & \dots & 0 \\ x[1] & x[0] & 0 & \dots & 0 \\ x[2] & x[1] & x[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[p] & x[p-1] & x[p-2] & \dots & x[0] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[q] & x[q-1] & x[q-2] & \dots & x[q-p] \end{bmatrix}}_{\boldsymbol{X}_0}
\underbrace{\begin{bmatrix} x[q+1] \\ x[q+2] \\ \vdots \\ x[q+p] \end{bmatrix}}_{\boldsymbol{x}_{q+1}}
\underbrace{\begin{bmatrix} x[q] & x[q-1] & \dots & x[q-p+1] \\ x[q+1] & x[q] & \dots & x[q-p+2] \\ \vdots & \vdots & \ddots & \vdots \\ x[q+p-1] & x[q+p-2] & \dots & x[q] \end{bmatrix}}_{\boldsymbol{X}_q}
\underbrace{\begin{bmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix}}_{\bar{\boldsymbol{a}}}
= \underbrace{\begin{bmatrix} b[0] \\ b[1] \\ b[2] \\ \vdots \\ b[p] \\ \vdots \\ b[q] \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{b}} \tag{58}$$

Here, we first use the bottom half (the last p rows) to solve for $\bar{\boldsymbol{a}}$ (i.e., $a[\cdot]$):

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}_{q+1} & \boldsymbol{X}_q \end{bmatrix} \boldsymbol{a} = \boldsymbol{0} \quad &\Leftrightarrow \quad \begin{bmatrix} \boldsymbol{x}_{q+1} & \boldsymbol{X}_q \end{bmatrix} \begin{bmatrix} 1 \\ \bar{\boldsymbol{a}} \end{bmatrix} = \boldsymbol{0} \\ \Rightarrow \quad \boldsymbol{X}_q \bar{\boldsymbol{a}} = -\boldsymbol{x}_{q+1} \quad &\Rightarrow \quad \bar{\boldsymbol{a}} = -\boldsymbol{X}_q^{-1} \boldsymbol{x}_{q+1} \end{aligned} \tag{59}$$

Note that $\boldsymbol{X}_q$ is a non-symmetric Toeplitz matrix, for which efficient specialized methods like the Trench algorithm exist for solving its inverse. Next, substitute into the upper half (the first $q+1$ rows) to obtain $b[\cdot]$:

$$\boldsymbol{b} = \boldsymbol{X}_0 \begin{bmatrix} 1 \\ \bar{\boldsymbol{a}} \end{bmatrix} \tag{60}$$

The Padé method is straightforward, but it clearly presents several issues:

1. It does not guarantee that the resulting system is stable;
2. It only constrains the first $p+q+1$ samples of the model output $\hat{x}[n]$ and the target signal $x[n]$ to be identical, and the matching performance beyond that may be poor;
3. $\boldsymbol{X}_q$ might be singular and thus unsolvable.

Regarding the case where $\boldsymbol{X}_q$ is singular and unsolvable, it can be considered that there is an issue with the default assumption of $a[0] = 1$ in the model. If the model is modified to let $a[0] = 0$, then the equations, although singular, are not unsolvable; rather, the solution is not unique.

If it is not an all-pole model, the new transfer function obtained this way will have a factor z in both the numerator and denominator, which can be canceled out. This is essentially a pole-zero cancellation occurring at zero. In terms of results, it is equivalent to a reduction in the model order, meaning there is redundancy in the model order.

7.3. Prony's Method

The Padé method uses all degrees of freedom on the first $p+q+1$ terms of the sequence, whereas the idea of Prony's method is simple: reduce the fitting requirements for this initial segment of the sequence to obtain a better fit from the perspective of the overall signal.

7.3.1. Prony Normal Equations

We first derive Prony's method from a more formal perspective. Specifically, we follow the idea of the Padé method to transform the problem into a linear one, as shown in Figure 5. We write the expression for the entire signal error, not just the first $p+q+1$ terms:

$$e[n] = \begin{cases} x[n] + \sum_{k=1}^{p} a[k] x[n-k] - b[n], & n = 0, 1, ..., q \\ x[n] + \sum_{k=1}^{p} a[k] x[n-k], & n > q \end{cases} \tag{61}$$

In Prony's method, we first solve for $a[\cdot]$ by minimizing the mean square error:

$$\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} |e[n]|^2 = \sum_{n=q+1}^{\infty} \left| x[n] + \sum_{k=1}^{p} a[k]x[n-k] \right|^2 \tag{62}$$

Considering only the error for the $n > q$ part is to make this part depend only on $a[\cdot]$. This is based on the need to solve for $a[\cdot]$ and $b[\cdot]$ in steps. It may sacrifice some accuracy in terms of definition, but its impact relative to an infinitely long $x[n]$ is not significant. Next, we formally take the partial derivatives and set them to zero to calculate the optimal values:

$$\frac{\partial \varepsilon_{p,q}}{\partial a^*[k]} = \sum_{n=q+1}^{\infty} \frac{\partial [e[n]e^*[n]]}{\partial a^*[k]} = \sum_{n=q+1}^{\infty} e[n] \frac{\partial e^*[n]}{\partial a^*[k]} = 0, \quad k = 1, 2, ..., p \tag{63}$$

From the definition Equation 61, we know that $\frac{\partial e^*[n]}{\partial a^*[k]} = x^*[n-k]$. Substituting this, we get:

$$\sum_{n=q+1}^{\infty} e[n]x^*[n-k] = 0, \quad k = 1, 2, ..., p \tag{64}$$

This equation expresses the orthogonal relationship between the minimum error and the signal, known as the Orthogonality principle. We continue by substituting the definition Equation 61 (note that the letter k is used, so we use l instead to avoid confusion):

$$\sum_{n=q+1}^{\infty} \left(x[n] + \sum_{l=1}^{p} a[l]x[n-l] \right) x^*[n-k] = 0, \quad k = 1, 2, ..., p \tag{65}$$

Moving terms and rearranging the order of the summation symbols, we can obtain:

$$\sum_{l=1}^{p} a[l] \left(\sum_{n=q+1}^{\infty} x^*[n-k]x[n-l] \right) = - \sum_{n=q+1}^{\infty} x^*[n-k]x[n], \quad k = 1, 2, ..., p \tag{66}$$

To simplify the expression, we denote:

$$r_x(k,l) := \sum_{n=q+1}^{\infty} x^*[n-k]x[n-l] \tag{67}$$

We can incidentally observe that $r_x(k,l) = r_x^*(l,k)$. Substituting this into the original equation gives:

$$\sum_{l=1}^{p} a[l] r_x(k,l) = -r_x(k,0), \quad k = 1, 2, ..., p \tag{68}$$

These are called the **Prony normal equations**. Written in matrix form:

$$\begin{bmatrix} r_x(1,1) & r_x(1,2) & \dots & r_x(1,p) \\ r_x(2,1) & r_x(2,2) & \dots & r_x(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p,1) & r_x(p,2) & \dots & r_x(p,p) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} r_x(1,0) \\ r_x(2,0) \\ \vdots \\ r_x(p,0) \end{bmatrix} \tag{69}$$

Denoted as:

$$\boldsymbol{R}_x \bar{\boldsymbol{a}} = -\boldsymbol{r}_x \tag{70}$$

It can be found that $\boldsymbol{R}_x$ is a Hermitian matrix. After obtaining $a[\cdot]$, we can substitute it back into Equation 61, setting the error to 0 for $n = 1, 2, \dots, q$, to obtain $b[k]$.

However, we need to note one thing: the original problem of minimizing $e[n]$ is still a joint non-linear least squares problem, where $a[\cdot]$ and $b[\cdot]$ are coupled. Therefore, our method of solving for $a[\cdot]$ and $b[\cdot]$ sequentially in two steps is actually a simplification and **cannot guarantee global minimization of the original error**.

7.3.2. An Equivalent Perspective from Pseudoinverse

In the derivation of the previous section, we naturally applied the least squares method to treat the problem as an optimization problem. In fact, we can also directly let all $\hat{x}[n] = x[n]$, resulting in an overdetermined system of equations:

$$\underbrace{\begin{bmatrix} x[0] & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[q] & x[q-1] & x[q-2] & \dots & x[q-p] \\ x[q+1] & x[q] & x[q-1] & \dots & x[q-p+1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[q+p] & x[q+p-1] & x[q+p-2] & \dots & x[q] \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}}_{\boldsymbol{X}_0} \underbrace{\begin{bmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix}}_{\bar{\boldsymbol{a}}} = \begin{bmatrix} b[0] \\ \vdots \\ b[q] \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \tag{71}$$

$\boldsymbol{x}_{q+1}$ (first column of $\boldsymbol{X}_0$) and $\boldsymbol{X}_q$ (remaining columns of $\boldsymbol{X}_0$)

Next, we can "internalize" the process of minimizing the mean square error into the process of solving for the least squares solution of this system using the pseudoinverse. The two approaches are essentially equivalent; the former has a smoother logic, while the latter helps in understanding the problem from the perspective of linear space. Solving using the pseudoinverse yields:

$$\bar{\boldsymbol{a}} = -\boldsymbol{X}_q^{+} \boldsymbol{x}_{q+1} = -\left(\boldsymbol{X}_q^H \boldsymbol{X}_q\right)^{-1} \boldsymbol{X}_q^H \boldsymbol{x}_{q+1} \tag{72}$$

That is, the optimal coefficient $\bar{\boldsymbol{a}}$ will be the solution to the following system of equations:

$$\left(\boldsymbol{X}_q^H \boldsymbol{X}_q\right)\bar{\boldsymbol{a}} = -\boldsymbol{X}_q^H \boldsymbol{x}_{q+1} \tag{73}$$

After making the following substitutions, we again obtain the Prony normal equations as in Equation 70:

$$\boldsymbol{R}_x = \boldsymbol{X}_q^H \boldsymbol{X}_q, \quad \boldsymbol{r}_x = \boldsymbol{X}_q^H \boldsymbol{x}_{q+1} \tag{74}$$

It can be verified by calculation that $\boldsymbol{R}_x$ is consistent with the definition in the previous section:

$$\begin{aligned}
\boldsymbol{R}_x &= \boldsymbol{X}_q^H \boldsymbol{X}_q \\
&= \begin{bmatrix} x^*[q] & x^*[q+1] & x^*[q+2] & \dots \\ x^*[q-1] & x^*[q] & x^*[q+1] & \dots \\ \vdots & \ddots & \vdots & \ddots \\ x^*[q-p+1] & x^*[q-p+2] & x^*[q-p+3] & \dots \end{bmatrix} \begin{bmatrix} x[q] & x[q-1] & \dots & x[q-p+1] \\ x[q+1] & x[q] & \dots & x[q-p+2] \\ x[q+2] & x[q+1] & \dots & x[q-p+3] \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \\
&= \sum_{n=q+1}^{\infty} \begin{bmatrix} x^*[n-1]x[n-1] & x^*[n-1]x[n-2] & \dots & x^*[n-1]x[n-p] \\ x^*[n-2]x[n-1] & x^*[n-2]x[n-2] & \dots & x^*[n-2]x[n-p] \\ \vdots & \vdots & \ddots & \vdots \\ x^*[n-p]x[n-1] & x^*[n-p]x[n-2] & \dots & x^*[n-p]x[n-p] \end{bmatrix} \\
&= \begin{bmatrix} r_x(1,1) & r_x(1,2) & \dots & r_x(1,p) \\ r_x(2,1) & r_x(2,2) & \dots & r_x(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p,1) & r_x(p,2) & \dots & r_x(p,p) \end{bmatrix}
\end{aligned} \tag{75}$$

This perspective also provides other useful information: regarding matrices of the form $A^H A$, for any vector $\boldsymbol{a}$, we have $\boldsymbol{a}^H(A^H A)\boldsymbol{a} = (\boldsymbol{A}\boldsymbol{a})^H(\boldsymbol{A}\boldsymbol{a}) = \|\boldsymbol{A}\boldsymbol{a}\|^2 \geq 0$. This indicates that $A^H A$ is a positive semi-definite matrix.

Consequently, the aforementioned Hermitian matrix $\boldsymbol{R}_x = \boldsymbol{X}_q^H \boldsymbol{X}_q$ is also a (positive semi-)definite matrix. This property determines that $A(z)$ is (marginally) stable, thereby addressing a drawback of the Padé method (TODO, is it? Does it also need to be Toeplitz?).

Furthermore, if $\boldsymbol{R}_x$ is a positive definite matrix, its eigenvalues are all positive, meaning the determinant is non-zero, the matrix is invertible, and a solution exists. If $\boldsymbol{R}_x$ is a positive semi-definite matrix containing zero eigenvalues, it is singular, but this actually indicates redundancy in the model order, which can be tried again after reduction.

7.3.3. The Minimum Error and Augmented Normal Equations

Since we are seeking the least squares solution, a minimum error will still exist between the final fitted signal and the true signal. Continuing the derivation from the definitions of $e[n]$ (Equation 61}) and $\varepsilon_{p,q}$ (Equation 62):

$$
\begin{aligned}
\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} |e[n]|^2 &= \sum_{n=q+1}^{\infty} e[n] \left(x[n] + \sum_{k=1}^{p} a[k] x[n-k] \right)^* \\
&= \sum_{n=q+1}^{\infty} e[n] x^*[n] + \sum_{n=q+1}^{\infty} e[n] \left(\sum_{k=1}^{p} a[k] x[n-k] \right)^* \\
&= \sum_{n=q+1}^{\infty} e[n] x^*[n] + \sum_{k=1}^{p} a^*[k] \left(\sum_{n=q+1}^{\infty} e[n] x^*[n-k] \right)
\end{aligned} \tag{76}
$$

Substituting the Orthogonality principle (Equation 64), which holds at the optimal solution, and the definition of $e[n]$ (Equation 61), we get:

$$
\begin{aligned}
\varepsilon_{p,q} = \sum_{n=q+1}^{\infty} e[n] x^*[n] &= \sum_{n=q+1}^{\infty} \left(x[n] + \sum_{k=1}^{p} a[k] x[n-k] \right) x^*[n] \\
&= \left(\sum_{n=q+1}^{\infty} x[n] x^*[n] \right) + \sum_{k=1}^{p} a[k] \left(\sum_{n=q+1}^{\infty} x[n-k] x^*[n] \right)
\end{aligned} \tag{77}
$$

Using the autocorrelation sequence $r_x(k,l)$ (Equation 67), this simplifies to:

$$
\varepsilon_{p,q} = r_x(0,0) + \sum_{k=1}^{p} a[k] r_x(0,k) \tag{78}
$$

Once in this form, we can unify $\varepsilon_{p,q}$ into the equation $\boldsymbol{R}_x \bar{\boldsymbol{a}} = -\boldsymbol{r}_x$ (by moving constants to the leftmost column of the matrix):

$$
\left[\begin{array}{c|cccc}
r_x(0,0) & r_x(0,1) & r_x(0,2) & \dots & r_x(0,p) \\
\hline
r_x(1,0) & r_x(1,1) & r_x(1,2) & \dots & r_x(1,p) \\
r_x(2,0) & r_x(2,1) & r_x(2,2) & \dots & r_x(2,p) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
r_x(p,0) & r_x(p,1) & r_x(p,2) & \dots & r_x(p,p)
\end{array}\right]
\left[\begin{array}{c}
1 \\ \hline a[1] \\ a[2] \\ \vdots \\ a[p]
\end{array}\right]
=
\left[\begin{array}{c}
\varepsilon_{p,q} \\ \hline 0 \\ 0 \\ \vdots \\ 0
\end{array}\right] \tag{79}
$$

Or (where $\boldsymbol{u}_1$ is a unit vector with the first element as 1 and others as 0):

$$
\bar{\boldsymbol{R}}_x \boldsymbol{a} = \varepsilon_{p,q} \boldsymbol{u}_1 \tag{80}
$$

This form is known as the **Augmented normal equations**. (TODO, the book used the same R_x here; for now, an overline is added to distinguish them)

We can also derive this using matrix form, which is more concise. The vector composed of the error sequence is:

$$\boldsymbol{e} = \boldsymbol{X}_q \bar{\boldsymbol{a}} + \boldsymbol{x}_{q+1} \tag{81}$$

According to the properties of the least squares solution, when we take the optimal solution that minimizes this error, it must be orthogonal to all columns of $\boldsymbol{X}_q$. That is:

$$\boldsymbol{X}_q^H \boldsymbol{e} = 0 \quad \Leftrightarrow \quad \boldsymbol{X}_q^H \left(\boldsymbol{X}_q \bar{\boldsymbol{a}} + \boldsymbol{x}_{q+1} \right) = 0 \quad \Leftrightarrow \quad \boldsymbol{R}_x \bar{\boldsymbol{a}} = -\boldsymbol{r}_x \tag{82}$$

For a filter of order p, q, the minimum error we are concerned with is:

$$\varepsilon_{p,q} = \|\boldsymbol{e}\|^2 = \boldsymbol{e}^H \boldsymbol{e} = \left(\boldsymbol{X}_q \bar{\boldsymbol{a}} + \boldsymbol{x}_{q+1} \right)^H \boldsymbol{e} = \boldsymbol{x}_{q+1}^H \boldsymbol{e} \tag{83}$$

The last step is because $\boldsymbol{X}_q^H \boldsymbol{e} = 0$. Continuing to substitute $\boldsymbol{e}$, we eventually get the same equation:

$$\begin{aligned} \varepsilon_{p,q} &= \boldsymbol{x}_{q+1}^H \boldsymbol{e} = \boldsymbol{x}_{q+1}^H \left(\boldsymbol{X}_q \bar{\boldsymbol{a}} + \boldsymbol{x}_{q+1} \right) \\ &= \boldsymbol{x}_{q+1}^H \boldsymbol{x}_{q+1} + \left(\boldsymbol{x}_{q+1}^H \boldsymbol{X}_q \right) \bar{\boldsymbol{a}} \\ &= r_x(0,0) + [r_x(0,1), r_x(0,2), ..., r_x(0,p)] \bar{\boldsymbol{a}} \\ &= r_x(0,0) + \sum_{k=1}^{p} a[k] r_x(0,k) \end{aligned} \tag{84}$$

7.4. Special Case: All-pole Modelling

We shall study the all-pole model (Equation 48), which is common in many physical processes.

7.4.1. All-pole Normal Equations

First, as usual, we treat the process of solving for $a[\cdot]$ as an optimization problem. Referring to the error definition in Equation 62, for $q = 0$ we have:

$$\varepsilon_{p,0} = \sum_{n=1}^{\infty} |e[n]|^2 \tag{85}$$

The definition of $e[n]$ still comes from Equation 61, but since $n - k < 0$ for $n = 0$, the value of the coefficient $x[n-k]$ for $a[\cdot]$ is 0, leaving only $x[0] - b[0]$:

$$e[n] = \begin{cases} x[0] - b[0], & n = 0 \\ x[n] + \sum_{k=1}^{p} a[k] x[n-k], & n > 0 \end{cases} \tag{86}$$

Now we apply a slight modification. Note that $e[0] = x[0] - b[0]$ can be viewed as a constant with respect to $a[\cdot]$. Therefore, when solving for $a[\cdot]$, minimizing $\varepsilon_{p,0}$ is equivalent to minimizing a new error we define, ε_p:

$$\varepsilon_p = \sum_{n=0}^{\infty} |e[n]|^2 \tag{87}$$

The difference is simply the inclusion of $e[0]$. Fortunately, after this change to the error function, we still derive the form of the Prony normal equations (see Equation 68), **but the definition of $r_x(k,l)$ changes** to:

$$r_x(k,l) := \sum_{n=0}^{\infty} x^*[n-k]x[n-l] \tag{88}$$

The difference is that the lower limit of the summation has changed from $q+1=1$ to 0. **This is the effect of changing the error function**. The specific reason requires re-derivation using the new error definition starting from Equation 63.

Actually, it's quite simple. After changing to ε_p, we set its partial derivative with respect to $a[\cdot]$ to zero:

$$\frac{\partial \varepsilon_p}{\partial a^*[k]} = \sum_{n=0}^{\infty} \frac{\partial [e[n]e^*[n]]}{\partial a^*[k]} = \sum_{n=0}^{\infty} e[n]\frac{\partial e^*[n]}{\partial a^*[k]} = 0, \quad k = 1, 2, ..., p \tag{89}$$

From the definition of $e[n]$ above, we still have $\frac{\partial e^*[n]}{\partial a^*[k]} = x^*[n-k]$. Since $x^*[n-k] = 0$ when $n = 0$, it happens to be consistent with $\frac{\partial (x[0]-b[0])}{\partial a^*[k]} = 0$. Thus, it can be unified and substituted to get:

$$\sum_{n=0}^{\infty} e[n]x^*[n-k] = 0, \quad k = 1, 2, ..., p \tag{90}$$

Continuing to substitute the definition of $e[n]$, and again due to the presence of the $x^*[n-k]$ term, the case for $n=0$ can be merged, directly yielding:

$$\sum_{n=0}^{\infty} \left(x[n] + \sum_{l=1}^{p} a[l]x[n-l] \right) x^*[n-k] = 0, \quad k = 1, 2, ..., p \tag{91}$$

Moving terms and rearranging the order of the summation symbols, we can obtain:

$$\sum_{l=1}^{p} a[l] \left(\sum_{n=0}^{\infty} x^*[n-k]x[n-l] \right) = -\sum_{n=q+1}^{\infty} x^*[n-k]x[n], \quad k = ... \tag{92}$$

This equation is consistent with the form of the Prony normal equations, with the only difference being that the definition of $r_x(k,l)$ needs to be as shown in Equation 88, changing to a summation starting from 0.

At this point, we have obtained the "normal equations" for solving $a[\cdot]$ in the case of an all-pole model. However, observing further, since $x[n] = 0$ for $n < 0$, substituting into Equation 88 yields:

$$
\begin{aligned}
r_x(k+1, l+1) &= \sum_{n=0}^{\infty} x^*[n-(k+1)]x[n-(l+1)] \\
&= \sum_{n=0}^{\infty} x^*[n-1-k]x[n-1-l] \\
&= \sum_{n=-1}^{\infty} x^*[n-k]x[n-l] \\
&= x^*[-1-k]x[-1-l] + \sum_{n=0}^{\infty} x^*[n-k]x[n-l] \\
&= 0 + \sum_{n=0}^{\infty} x^*[n-k]x[n-l] \\
&= r_x(k, l), \quad (\forall k, l \geq 0)
\end{aligned} \tag{93}
$$

Thus, we can let:

$$
r_x(k-l) := r_x(k, l) = \sum_{n=0}^{\infty} x^*[n-k]x[n-l] \tag{94}
$$

Which is:

$$
r_x(k) = \sum_{n=0}^{\infty} x^*[n-k]x[n] \tag{95}
$$

Observation shows that $r_x(k)$ is conjugate symmetric, i.e., $r_x(k) = r_x^*(-k)$. Substituting this yields a more concise system of equations applicable to the All-pole model:

$$
\sum_{l=1}^{p} a[l] r_x(k-l) = -r_x(k), \quad k = 1, 2, ..., p \tag{96}
$$

Or:

$$
\begin{bmatrix}
r_x(0) & r_x^*(1) & ... & r_x^*(p-1) \\
r_x(1) & r_x(0) & ... & r_x^*(p-2) \\
\vdots & \vdots & \ddots & \vdots \\
r_x(p-1) & r_x(p-2) & ... & r_x(0)
\end{bmatrix}
\begin{bmatrix}
a[1] \\ a[2] \\ \vdots \\ a[p]
\end{bmatrix}
= -
\begin{bmatrix}
r_x(1) \\ r_x(2) \\ \vdots \\ r_x(p)
\end{bmatrix} \tag{97}
$$

This is known as the **All-pole normal equations**. Since the matrix $\boldsymbol{R}_x$ is conjugate symmetric and Toeplitz, it allows us to use the Levinson-Durbin algorithm for efficient computation.

For the calculation of the minimum error value, similarly, we have:

$$
\varepsilon_p = r_x(0) + \sum_{k=1}^{p} a[k] r_x^*(k) \tag{98}
$$

It can also be written together with the system of equations in a form similar to the Augmented normal equations, which will not be repeated here.

7.4.2. Issues on the Numerator Selection

By conventional methods, after obtaining $a[\cdot]$, we would use Equation 60 to get $b[0] = x[0]$. However, at the end of Section 7.3.1, we mentioned that the step-by-step solution method in the Prony method does not guarantee global optimality. Modifying the value of $b[0]$ here does not necessarily destroy the optimality of the result, because the result wasn't optimal to begin with; conversely, we might even achieve better results by changing the way $b[\cdot]$ is selected.

The so-called better result is not necessarily a reduction in the mean square error value, but might also integrate considerations of other factors. The all-pole model here is an example: if the value of $x[0]$ in the original signal is not so credible due to noise or other interference, to prevent the entire set of model parameters from being biased by this single $b[0] = x[0]$, we prefer to make the energy of the fitted signal $\hat{x}[n]$ (which equals the unit impulse response $h[n]$ in our model) equal to the energy of the target signal $x[n]$:

$$r_{\hat{x}}(0) = r_h(0) = r_x(0) \tag{99}$$

Derivation shows that one should take $b[0] = \sqrt{\varepsilon_p}$.

(TODO) I haven't quite figured out how to derive this value for $b[0]$; the reference book says it will be covered in its section 5.2.3.

7.5. Finite Data Records for All-pole Cases

The previous analysis of the Prony method was based on the assumption that $x[n]$ is defined over the entire positive time domain, from 0 to ∞. Now we need to consider the case where we only have $\boldsymbol{N+1}$ **samples** on $[0, N]$. As for why it is $N+1$ samples instead of N, I don't know; perhaps the author felt the subsequent indexing would be more concise. To avoid errors, I have written it this way as well, although it is somewhat inconvenient.

The following two approaches are commonly used for all-pole models, so we will only discuss all-pole models by default.

7.5.1. Auto-correlation Method

The first method is called the autocorrelation method. We consider applying a rectangular window to $x[n]$, or in other words, treating the parts of $x[n]$ outside $[0, N]$ as having a value of 0:

$$x_N[n] = \begin{cases} x[n], & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases} \tag{100}$$

Then we directly apply the Prony method for the solution. Note that the $r_x(k)$ estimated using the windowed $x_N[n]$ will become:

$$r_x(k) = \sum_{n=0}^{\infty} x_N^*[n-k]x_N[n] = \sum_{n=k}^{N} x^*[n-k]x[n], \quad k = 0, 1, ..., p \tag{101}$$

For ease of presentation, we write out the overdetermined system:

$$\boldsymbol{X}_p \bar{\boldsymbol{a}}_p = -\boldsymbol{x}_1$$

$$\Rightarrow \left[\begin{array}{ccccc} x[0] & 0 & 0 & \dots & 0 \\ x[1] & x[0] & 0 & \dots & 0 \\ x[2] & x[1] & x[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline x[p-1] & x[p-2] & x[p-3] & \dots & x[0] \\ x[p] & x[p-1] & x[p-2] & \dots & x[1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[N-2] & x[N-3] & x[N-4] & \dots & x[N-p-1] \\ x[N-1] & x[N-2] & x[N-3] & \dots & x[N-p] \\ \hline x[N] & x[N-1] & x[N-2] & \dots & x[N-p+1] \\ 0 & x[N] & x[N-1] & \dots & x[N-p+2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & x[N] \end{array}\right] \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \left[\begin{array}{c} x[1] \\ x[2] \\ x[3] \\ \vdots \\ \hline x[p] \\ x[p+1] \\ \vdots \\ x[N-1] \\ x[N] \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{102}$$

The normal equations, except for the definition of $r_x(k)$ being modified as described in Equation 101, remain formally the same as the previous All-pole normal equations (see Equation 96):

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \dots & r_x^*(p-1) \\ r_x(1) & r_x(0) & \dots & r_x^*(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(p) \end{bmatrix} \tag{103}$$

The form of the minimum error for this method is also consistent with Equation 98 in the All-pole analysis, requiring only the modification of the autocorrelation function.

The autocorrelation method truncates the signal directly, even if the signal values outside the interval are non-zero, so the results provided may be **biased** compared to the actual solution.

(TODO) However, this method has an important property: it guarantees that the resulting model is **stable**, which is very useful for cases requiring significant extrapolation or analysis. The book mentions that the proof is covered in Chapter 5 and is omitted here for now.

7.5.2. Covariance Method

The Auto-correlation Method sets parts outside the domain to 0, which essentially changes the form of $x[n]$, as 0 is also a normal signal value. In some cases, this does not achieve the best results.

The second method, the Covariance Method, typically yields more accurate results. It makes no assumptions about the signal itself but instead ignores samples outside the domain during the optimization process.

Following the standard procedure of defining error and solving the optimization problem, if we cannot consider samples outside the domain, the error can only be defined over the valid interval. Based on the previous error definition, calculating $e[n]$ requires $x[n], x[n-1], \ldots, x[n-p]$, so we can only define the error on $[p, N]$:

$$\mathcal{E}_p^C = \sum_{n=p}^{N} |e[n]|^2 \tag{104}$$

We can then take partial derivatives of this error with respect to the coefficients and repeat the derivation process to obtain the normal equations, but we will not expand on that here.

Looking at it from the perspective of an overdetermined system of equations is simpler: it essentially involves deleting the expressions in the autocorrelation method that involve samples outside the domain (such as $x[N+1]$, etc.). Referring to the overdetermined system of the autocorrelation method in Equation 102, extracting only the part between the dashed lines gives the overdetermined system for the covariance method:

$$\begin{bmatrix} x[p-1] & x[p-2] & x[p-3] & \ldots & x[0] \\ x[p] & x[p-1] & x[p-2] & \ldots & x[1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[N-2] & x[N-3] & x[N-4] & \ldots & x[N-p-1] \\ x[N-1] & x[N-2] & x[N-3] & \ldots & x[N-p] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} x[p] \\ x[p+1] \\ \vdots \\ x[N-1] \\ x[N] \end{bmatrix} \tag{105}$$

Its normal equations are the same as the original Prony normal equations (see Equation 68):

$$\begin{bmatrix} r_x(1,1) & r_x(1,2) & \ldots & r_x(1,p) \\ r_x(2,1) & r_x(2,2) & \ldots & r_x(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p,1) & r_x(p,2) & \ldots & r_x(p,p) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} r_x(1,0) \\ r_x(2,0) \\ \vdots \\ r_x(p,0) \end{bmatrix} \tag{106}$$

Doing this actually discards the Toeplitz property of the matrix in the All-pole analysis. Similarly, the autocorrelation function needs to be changed to the finite-data version:

$$r_x(k,l) := \sum_{n=p}^{N} x^*[n-k]x[n-l] \tag{107}$$

The minimum error value also formally follows Equation 78, needing only the modification of the autocorrelation function.

7.6. Example: Channel Inversion

(TODO)

8. Stochastic Modelling Identification

There are two main differences between modelling stochastic processes and modelling deterministic signals. First, in deterministic modelling, since the specific sample values of $x[n]$ are known, the error is defined based on those values; however, in stochastic modelling, we only possess the statistical characteristics of $x[n]$, making the previous definition of $e[n]$ unsuitable. Second is the difference in input signals: since we are modelling a stochastic process, the input is no longer a unit impulse signal but rather white noise with unit variance, as shown in Figure 3.

Given these differences, we must also assume stationarity for the stochastic process being modelled, specifically that the process is Wide-Sense Stationary (WSS).

Similarly, for stochastic processes, we can replace the least squares error in Section 7.1 with the mean square error $\mathcal{E}_{\text{MS}} = E\{|x[n] - \hat{x}[n]|^2\}$ for optimization. However, this leads to the same non-linear problems that are difficult to handle, requiring alternative solutions.

Note that from here on, $x[n]$, $v[n]$, etc., no longer represent specific discrete signals but rather stochastic processes. In practical applications, we may not directly know their statistical characteristics, in which case we must estimate them from specific signals (realizations or samples). If only one realization is available for estimation, the corresponding stationarity and ergodicity assumptions must hold for the results to be meaningful.

For convenience, I will not change all subsequent square brackets to parentheses.

8.1. Autoregressive Moving Average (ARMA) Processes

We first define a type of stochastic process called an ARMA process. Consider filtering white noise $v[n]$ with variance σ_v^2 using the transfer function of an ARMA model (Equation 44) to obtain the output $x[n]$.

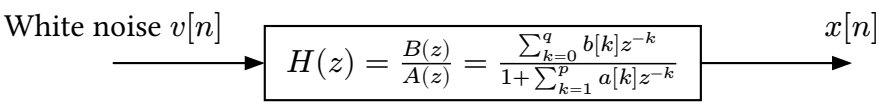

Figure 6: Diagram of ARMA process generation

Assuming $H(z)$ is stable, the output stochastic process $x[n]$ will be WSS (proof omitted). Since the power spectrum of white noise is $P_v(z) = \sigma_v^2$, the power spectrum of $x[n]$ is:

$$P_x(z) = \sigma_v^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)} \tag{108}$$

In the frequency domain, this is:

$$P_x(e^{j\omega}) = \sigma_v^2 \frac{\left|B(e^{j\omega})\right|^2}{\left|A(e^{j\omega})\right|^2} \tag{109}$$

We define a process with a power spectrum of this form as an $\mathrm{ARMA}(p, q)$ process. Note that due to symmetry, its power spectrum has $2p$ poles and $2q$ zeros.

> To clarify once more, Section 6.1 mentioned ARMA models, whereas here we are discussing ARMA processes. The latter refers to the stochastic process that the output signal satisfies when white noise is passed through an ARMA model.

8.1.1. Yule-Walker Equations

In stochastic modelling, we want the output of the constructed model to have the same statistical characteristics as the target process, such as ensuring the output's autocorrelation $r_{x(k)}$ matches that of the target process. Therefore, we need to **establish the statistical relationship between the model output's autocorrelation $r_x(k)$ and the system parameters $a[\cdot]$, $b[\cdot]$, and the unit impulse response $h[n]$.**

By definition, for an ARMA process $x[n]$ derived from $v[n]$, the following equation is satisfied:

$$x[n] + \sum_{l=1}^{p} a[l]x[n-l] = \sum_{l=0}^{q} b[l]v[n-l] \tag{110}$$

We can derive a similar relationship between the autocorrelation of $x[n]$ and the cross-correlation of $x[n]$ with $v[n]$ by multiplying both sides of the equation by $x^*[n-k]$ and taking the expectation:

$$E\left\{x[n]x^*[n-k] + \sum_{l=1}^{p} a[l]x[n-l]x^*[n-k]\right\} = E\left\{\sum_{l=0}^{q} b[l]v[n-l]x^*[n-k]\right\} \tag{111}$$

Which is:

$$E\{x[n]x^*[n-k]\} + \sum_{l=1}^{p} a[l]E\{x[n-l]x^*[n-k]\} = \sum_{l=0}^{q} b[l]E\{v[n-l]x^*[n-k]\} \tag{112}$$

Under the assumption of stationarity, substituting the definitions of autocorrelation and cross-correlation (see Section 3.1) yields:

$$r_x(k) + \sum_{l=1}^{p} a[l]r_x(k-l) = \sum_{l=0}^{q} b[l]r_{vx}(k-l) \tag{113}$$

The presence of the cross-correlation term $r_{vx}(k-l)$ means the equation still contains v. We can replace it using the unit impulse response $h[n]$, which represents system properties, by substituting $x[n] = v[n] * h[n] = \sum_{m=-\infty}^{\infty} v[m]h[n-m]$:

$$
\begin{aligned}
r_{vx}(k-l) &= E\{v[k]x^*[l]\} \\
&= E\left\{v[k]\left(\sum_{m=-\infty}^{\infty} v[m]h[l-m]\right)^*\right\} \\
&= \sum_{m=-\infty}^{\infty} E\{v[k]v^*[m]\}h^*[l-m]
\end{aligned} \tag{114}
$$

The use of $r_{vx}(k-l) = E\{v[k]x^*[l]\}$ here is consistent with the previously defined $r_{vx}(k-l) = E\{v[n-l]x^*[n-k]\}$ due to the stationarity assumption, because $(n-l)-(n-k) = k-l$. This substitution makes the derivation more concise.

Since $v[n]$ is white noise with independent and identical distribution and variance σ_v^2:

$$
E\{v[k]v^*[m]\} = \begin{cases} \sigma_v^2, & m = k \\ 0, & \text{otherwise} \end{cases} \tag{115}
$$

This term is zero whenever $m \neq k$. Substituting this into the previous expression gives:

$$
r_{vx}(k-l) = \sigma_v^2 h^*[l-k] \tag{116}
$$

Thus, we obtain an expression that does not contain v:

$$
r_x(k) + \sum_{l=1}^{p} a[l]r_x(k-l) = \sigma_v^2 \sum_{l=0}^{q} b[l]h^*[l-k] \tag{117}
$$

Finally, considering practical constraints, we **assume the system is causal**, meaning $h[n] = 0$ for $n < 0$. Thus, $h^*[l-k]$ is zero when $l < k$. We can modify the upper and lower limits of the summation on the right side of the equation and denote it as $c[k]$:

$$
\begin{aligned}
c[k] &:= \sum_{l=0}^{q} b[l]h^*[l-k] = \sum_{l=k}^{q} b[l]h^*[l-k] = \sum_{l=0}^{q-k} b[l+k]h^*[l] \\
&= b[k] * h^*[-k]
\end{aligned} \tag{118}
$$

Incidentally, this term is zero when $k > q$. Finally, we obtain the **Yule-Walker Equations**:

$$
r_x(k) + \sum_{l=1}^{p} a[l]r_x(k-l) = \begin{cases} \sigma_v^2 c[k], & 0 \leq k \leq q \\ 0, & k > q \end{cases} \tag{119}
$$

Note that from now on, we will default to the unit variance assumption, i.e., $\sigma_v^2 = 1$.

This achieves our goal: establishing the statistical relationship between the model output's autocorrelation $r_x(k)$, the system parameters $a[\cdot]$ and $b[\cdot]$, and the unit impulse response $h[n]$.

As a side note, for $k > q$:

$$r_x(k) = -\sum_{l=1}^{p} a[l] r_x(k-l) \tag{120}$$

The values of the autocorrelation function can be extrapolated using filter parameters and known autocorrelation values.

For clarity, we also present its matrix form:

$$\begin{bmatrix} r_x(0) & r_x(-1) & \dots & r_x(-p) \\ r_x(1) & r_x(0) & \dots & r_x(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(q) & r_x(q-1) & \dots & r_x(q-p) \\ \hline r_x(q+1) & r_x(q) & \dots & r_x(q-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(q+p) & r_x(q+p-1) & \dots & r_x(q) \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = \sigma_v^2 \begin{bmatrix} c[0] \\ c[1] \\ \vdots \\ c[q] \\ \hline 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} c[0] \\ c[1] \\ \vdots \\ c[q] \\ \hline 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \tag{121}$$

8.1.2. Modified Yule-Walker Equation (MYWE) Method

The Yule-Walker equations can be used to solve for filter parameters from the autocorrelation function, but due to the presence of $h^*[l]$, it remains a difficult non-linear problem.

To clarify again, in this problem we are modelling the stochastic process rather than a specific signal. Thus, the statistical characteristics of the target process (such as the autocorrelation function) are considered known. If the value of the autocorrelation function $r_v(k)$ is unknown, it must be estimated as $\hat{r}_v(k)$ from some realizations (samples) using statistical methods.

Returning to the problem of parameter identification, we can follow the approach of the Padé method by solving in steps to **approximate** the optimal result. First, we use the portion where $q < k \leq q + p$ to estimate $a[\cdot]$, with the corresponding equations:

$$\begin{bmatrix} r_x(q) & r_x(q-1) & \dots & r_x(q-p+1) \\ r_x(q+1) & r_x(q) & \dots & r_x(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(q+p-1) & r_x(q+p-2) & \dots & r_x(q) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} r_x(q+1) \\ r_x(q+2) \\ \vdots \\ r_x(q+p) \end{bmatrix} \tag{122}$$

This system of equations is called the **Modified Yule-Walker equations** (MYWE), and the method is thus called the MYWE method. Note that the form of this system is identical to that of Equation 59 in the Padé method, except the values of $x[n]$ are replaced by the autocorrelation function. This matrix is also Toeplitz, allowing for accelerated solutions using algorithms like the Trench algorithm.

After obtaining $a[\cdot]$, the **second step is to solve for $b[\cdot]$**. Substituting $a[\cdot]$ back into the Yule-Walker equations gives the values of $c[\cdot]$. However, since $c[k] := b[k] * h^*[-k]$ and $h[k]$ even depends on $b[k]$, solving for $b[\cdot]$ is extremely difficult. The lecture slides state "We skip this," seemingly not intending to address this part. Related content in the reference book starts around page 190, mentioning several methods.

First, knowing $a[\cdot]$, we can construct an AR filter $A(z)$ to filter $x[n]$ and obtain a new process $y[n]$:

$$P_x(z) = \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)} \quad \overset{A(z)}{\longrightarrow} \quad P_y(z) = B(z)B^*(1/z^*) \tag{123}$$

This process is an MA process, which can then be handled using the methods in Section 8.3 to estimate $b[\cdot]$.

Second, we can avoid explicit filtering, though the essence is likely the same. After solving for $c[\cdot]$ through the upper part of the Yule-Walker equations, the Laplace transform of the positive axis is obtained (since only the positive axis values can be solved through the Yule-Walker equations):

$$[C(z)]_+ = \sum_{k=0}^{\infty} c[k]z^{-k} \tag{124}$$

Correspondingly, though unknown, the Laplace transform of the negative axis is:

$$[C(z)]_- = \sum_{k=-\infty}^{-1} c[k]z^{-k} = \sum_{k=1}^{\infty} c[-k]z^{k} \tag{125}$$

From the definition $c[k] := b[k] * h^*[-k]$, the power spectrum of the MA process is:

$$\begin{aligned} C(z) &= B(z)H^*(1/z^*) = B(z)\frac{B^*(1/z^*)}{A^*(1/z^*)} \\ \Rightarrow \quad P_y(z) &\equiv C(z)A^*(1/z^*) = B(z)B^*(1/z^*) \end{aligned} \tag{126}$$

Expanding this:

$$P_y(z) = C(z)A^*(1/z^*) = [C(z)]_+A^*(1/z^*) + [C(z)]_-A^*(1/z^*) \tag{127}$$

Since the negative axis values of $a[k]$ are 0, $A^*(1/z^*)$ contains only positive powers of z, as does $[C(z)]_+$ (TODO, did the book use a minus sign here?). Thus, the causal part of $P_y(z)$ is:

$$\left[P_y(z)\right]_+ = [C(z)]_+A^*(1/z^*)]_+ \tag{128}$$

Therefore, despite not knowing the values of $c[k]$ on the negative axis, we can use this equation and the known positive axis values of $c[\cdot]$ along with $a[\cdot]$ to solve for $\left[P_y(z)\right]_+$, then obtain the full $P_y(z)$ through conjugate symmetry. Finally, spectral factorization is performed to obtain the coefficients $b[\cdot]$:

$$P_y(z) = B(z)B^*(1/z^*) \tag{129}$$

(TODO, should I copy it over?) There is a clear example on page 192 of the reference book.

8.1.3. Extended Yule-Walker Equation Method

Correspondingly, in the first step, we can use an approach similar to Prony's method by including all equations where $k > q$. The resulting overdetermined system is called the **Extended Yule-Walker equations**.

A least squares solution is then formally sought, using a process similar to Prony's method, the details of which will not be repeated.

8.2. Autoregressive (AR) Processes

We again consider the all-pole case. Since only $b[0]$ remains, the equations simplify significantly:

$$r_x(k) + \sum_{l=1}^{p} a[l] r_x(k-l) = |b[0]|^2 \delta(k), \quad k \geq 0 \tag{130}$$

Since the complex issue of $c[k]$ is absent, Prony's method can be directly applied. First, solve for $a[\cdot]$ using all equations except the first one; note that the system is identical to Equation 97. Then, use the first equation to derive $b[0]$. This is called the Yule-Walker method (a confusing naming convention).

(TODO, what is the 1/N at the front? Although it seems to cancel out when solving for a[·], and b[·] is unaffected due to direct selection. Also, what about the asterisk? See page 194, eq 4.153 of the book.) Once again, if the autocorrelation of the target process is unknown, it must be estimated from samples (satisfying the ergodicity assumption), for example:

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] x^*[n-k] \tag{131}$$

This makes the approach equivalent to the previously mentioned Auto-correlation method (Section 7.5.1), which aligns with intuition.

8.3. Moving Average Processes

For MA processes, substituting into the Yule-Walker equations yields:

$$r_x(k) = \sum_{l=0}^{q} b[l] b^*[l-k] = b[k] * b^*[-k] \tag{132}$$

Thus:

$$P_x(z) = B(z) B^*(1/z^*) \tag{133}$$

In summary, the results are obtained by taking the z-transform of the autocorrelation function $r_x(k)$ to get the power spectrum $P_x(z)$, followed by spectral factorization. For example:

$$r_x(k) = 17\delta(k) + 4[\delta(k-1) + \delta(k+1)] \tag{134}$$

Taking the z-transform:

$$P_x(z) = 17 + 4z^{-1} + 4z = (4 + z^{-1})(4 + z) \tag{135}$$

Thus:

$$B(z) = 4 + z^{-1} \quad \text{or} \quad B(z) = 1 + 4z^{-1} \tag{136}$$

Additionally, there are methods such as Durbin's method, which are not recorded here.

9. Spectrum Estimation

Estimation of the power spectrum is clearly a very useful tool. For example, under the assumption of additive noise and the signal being uncorrelated with the noise, a non-causal Wiener smoothing filter has the following frequency response:

$$H(e^{j\omega}) = \frac{P_{dx}(e^{j\omega})}{P_x(e^{j\omega})} = \frac{P_d(e^{j\omega})}{P_d(e^{j\omega}) + P_v(e^{j\omega})} \tag{137}$$

If the power spectral densities of the target signal and the noise are known, the frequency response can be calculated directly; if unknown, the result can be obtained through spectrum estimation. Other examples include the detection and tracking of narrow-band signals.

In short, this section considers the estimation of the power spectral density (PSD) of wide-sense stationary (WSS) stochastic processes. Specifically, given a random signal generated by a stochastic process, how can we effectively estimate the spectrum of that process?

The most direct idea comes from the Wiener–Khinchin Theorem: we know that the power spectrum can be obtained by the Fourier transform of the autocorrelation function:

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_x(k) e^{jk\omega} \tag{138}$$

For a random signal $x[n]$ generated by an ergodic stochastic process, we can obtain the autocorrelation function of the process as follows:

$$r_x(k) = \lim_{N\to\infty} \left\{ \frac{1}{2N+1} \sum_{n=-N}^{N} x[n+k]x^*[n] \right\} \tag{139}$$

This is because, under the ergodicity assumption, we can unbiasedly estimate the statistical characteristics of the process using a single infinitely long realization.

However, this approach clearly has some issues: first, the samples we possess are often finite in length—for example, signals like seismic waves are inherently short, and signals like speech only approximately satisfy the stationarity assumption over very short durations; second, signal samples themselves often contain noise.

Methods for spectrum estimation can be divided into two categories: one is the Nonparametric approach, which starts by estimating the autocorrelation of the sequence and then transforms it to obtain the power spectrum; the other is the Parametric approach, usable when there is prior knowledge of the stochastic process model, which starts by estimating the model parameters and then calculates the power spectrum from the model.

9.1. Nonparametric Spectrum Estimation

9.1.1. Periodogram

The beginning of this chapter mentioned the method of estimating the autocorrelation function from samples and then performing a Fourier transform to obtain the power spectrum.

Even if the sample length is finite (e.g., N), we use these available samples to directly estimate the autocorrelation function:

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n+k]x^*[n], \quad k = 0, 1, ..., N-1 \tag{140}$$

This effectively assumes that the values of $x[n]$ outside $[0, N-1]$ are all 0, which is equivalent to setting the summation range from 0 to $N-1-k$, meaning there are $N-k$ terms in the sum. However, since we divide by N, the result is a biased estimation:

$$\begin{aligned} E\{\hat{r}_x(k)\} &= \frac{1}{N} \sum_{n=0}^{N-1-k} E\{x[n+k]x^*[n]\} \\ &= \frac{1}{N} \sum_{n=0}^{N-1-k} r_x(k) \\ &= \frac{N-k}{N} r_x(k), \quad k = 0, 1, ..., N-1 \end{aligned} \tag{141}$$

As N approaches infinity, the expectation equals the actual value, so it is Asymptotically Unbiased. The parts for negative k can be obtained by flipping the result using the conjugate symmetry property of WSS autocorrelation functions, so it can be rewritten as:

$$E\{\hat{r}_x(k)\} = w_B(k) r_x(k) \tag{142}$$

Where:

$$w_B(k) = \begin{cases} \frac{N-|k|}{N}, & |k| \leq N \\ 0, & |k| > N \end{cases} \tag{143}$$

This is a Bartlett (triangular) window.

Next, we use the Fourier transform to estimate the power spectrum:

$$\hat{P}(e^{j\omega}) = \sum_{k=-N+1}^{N-1} \hat{r}_x(k) e^{-jk\omega} \tag{144}$$

The two-step process above primarily follows the definition. Once the principle is understood, we can simplify it to obtain the spectral estimate directly from the random signal $x[n]$. First, the assumption that $x[n] = 0$ outside $[0, N-1]$ can be replaced by the process of multiplying by a Dirichlet kernel (Equation 41), i.e., $x_N[n] = x[n]w_R[n]$, whose Fourier transform is:

$$X_N(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x_N[n] e^{-j\omega n} = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \tag{145}$$

Since:

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=-\infty}^{\infty} x_N[n+k] x_N^*[n] = \frac{1}{N} x_N[k] * x_N^*[-k] \tag{146}$$

From the properties of the DTFT, the power spectrum estimate is obtained as:

$$\hat{P}_{\text{per}}(e^{j\omega}) = \frac{1}{N} X_N(e^{j\omega}) X_N^*(e^{j\omega}) = \frac{1}{N} \left| X_N(e^{j\omega}) \right|^2 \tag{147}$$

A power spectrum estimate satisfying this definition is called a **periodogram**.

9.1.1.1. An Equivalent Perspective from a Filter Bank

The previous derivation was based on the Wiener–Khinchin theorem, following the logic of "estimate autocorrelation function → Fourier transform to get spectral estimate". Next, we interpret the meaning of the periodogram from a different perspective.

An intuitive idea is that if we know the power value of each frequency component of a signal, we can **directly assemble its power spectral density plot**. So, how do we find the power of a specific frequency component? Naturally, we can first use a very narrow band-pass filter to extract that frequency component and then find a way to obtain its power; Parseval's theorem can be used here, as seen later.

Specifically, we define a set of FIR filters of length N as follows:

$$h_i[n] = \frac{1}{N} e^{jn\omega_i} w_R[n] = \begin{cases} \frac{1}{N} e^{jn\omega_i}, & 0 \le n < N \\ 0, & \text{otherwise} \end{cases} \tag{148}$$

The Fourier transform of these filters is:

$$H_i(e^{j\omega}) = \sum_{n=0}^{N-1} h_i[n] e^{-jn\omega} = e^{-j(\omega-\omega_i)(N-1)/2} \frac{\sin(N(\omega-\omega_i)/2)}{N \sin((\omega-\omega_i)/2)} \tag{149}$$

The bandwidth of its main lobe is approximately $\Delta\omega = \frac{2\pi}{N}$.

The reason for designing the filter this way is that the filter's unit impulse response $h_i[n]$ convolves with $x[n]$ in the time domain, which corresponds to the product of their Fourier transforms $H_i(e^{j\omega})$ and $X(e^{j\omega})$ in the frequency domain. We hope that after multiplication in the frequency domain, only the component of $X(e^{j\omega})$ at frequency ω_i remains, with all others being zero—effectively sampling it. Thus, ideally, $H_i(e^{j\omega})$ should be $\delta(\omega - \omega_i)$, whose time-domain expression is $\frac{1}{2\pi} e^{jn\omega_i}$.

Furthermore, since our sequence length is finite, we cannot implement an ideal $h_i[n]$ and must truncate it at length N. Here, we set the coefficient before $e^{jn\omega_i}$ to $1/N$ to ensure $\left| H_i(e^{j\omega}) \right|_{\omega=\omega_i} = 1$, simplifying calculations.

Note that we are now examining the true power spectral density $P_x(e^{j\omega_i})$ of the stochastic process. Therefore, all instances of $x[n]$ below refer to the **stochastic process** behind the signal, examining its overall behavior after filtering rather than a specific signal.

Thus, filtering $x[n]$ through $h_i[n]$ results in another stochastic process $y_i[n]$:

$$y_i[n] = x[n] * h_i[n] = \sum_{k=n-N+1}^{n} x[k]h_i[n-k] = \frac{1}{N}\sum_{k=n-N+1}^{n} x[k]e^{j(n-k)\omega_i} \tag{150}$$

Since $\left|H_i(e^{j\omega})\right|_{\omega=\omega_i} = 1$, at the frequency point ω_i, the power spectral density values of the input $x[n]$ and the output $y_i[n]$ should be identical:

$$P_x(e^{j\omega_i}) = P_y(e^{j\omega_i}) \tag{151}$$

Next, we need to find this $P_y(e^{j\omega_i})$ to obtain the desired $P_x(e^{j\omega_i})$. This is done by considering Parseval's theorem:

$$E\left\{|y_i[n]|^2\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_y(e^{j\omega})\left|H_i(e^{j\omega})\right|^2 \mathrm{d}\omega \tag{152}$$

If the bandwidth of $H_i(e^{j\omega})$ is narrow enough and the sidelobes are small enough to approximate an ideal band-pass filter, we can assume that $P_y(e^{j\omega}) = P_y(e^{j\omega_i})$ is uniform within the passband and zero within the stopband. Thus:

$$E\left\{|y_i[n]|^2\right\} \approx \frac{1}{2\pi}(\Delta\omega \cdot P_y(e^{j\omega_i})) = \frac{1}{N}P_y(e^{j\omega_i}) = \frac{1}{N}P_x(e^{j\omega_i}) \tag{153}$$

This conclusion shows that we can use the power of $y_i[n]$ to estimate $P_x(e^{j\omega_i})$:

$$P_x(e^{j\omega_i}) \approx NE\left\{|y_i[n]|^2\right\} \tag{154}$$

Now, since $y_i[n]$ represents a stochastic process here, we cannot examine its specific values. The problem becomes how to estimate this power $\hat{E}\left\{|y_i[n]|^2\right\}$ from actual samples.

At this stage, we have **many choices**, but to derive the periodogram, we use a one-point average to estimate it:

$$\hat{E}\left\{|y_i[n]|^2\right\} = |y_i[N-1]|^2 = \frac{1}{N^2}\left|\sum_{k=0}^{N-1} x[k]e^{-jk\omega_i}\right|^2 \tag{155}$$

This is actually an extreme and imprecise approximation. Even though a one-point average is an unbiased estimate of power, it is far too susceptible to fluctuations in the actual signal values. **However, it can be said that this is merely a way to arrive at the definition of the periodogram, providing one mode of interpretation.** Using this estimation method, we finally obtain:

$$\hat{P}_x(e^{j\omega_i}) = N|y_i[N-1]|^2 = \frac{1}{N}\left|\sum_{k=0}^{N-1} x[k]e^{-jk\omega_i}\right|^2 \tag{156}$$

This is consistent with the definition of the periodogram in Equation 147.

In Section 9.1.5 later, we will use a filter bank approach similar to this one, but based on Equation 34, we use the autocorrelation matrix and filter coefficients to obtain a better power estimate.

9.1.1.2. Performance of the Periodogram

We now evaluate the performance of the periodogram as a power spectrum estimate. First, we certainly hope that the periodogram calculated from samples converges to the actual power spectrum of the stochastic process. Due to its random nature, we must consider convergence in a statistical sense, such as mean-square convergence:

$$\lim_{N\to\infty} E\Big\{\Big[\hat{P}_{\text{per}}(e^{j\omega}) - P_x(e^{j\omega})\Big]^2\Big\} = 0 \tag{157}$$

To satisfy this, we need its mean to be asymptotically unbiased and its variance to approach zero as the sample size becomes sufficiently large:

$$\begin{aligned}\lim_{N\to\infty} E\Big\{\hat{P}_{\text{per}}(e^{j\omega})\Big\} &= P_x(e^{j\omega}) \\ \lim_{N\to\infty} \text{Var}\Big\{\hat{P}_{\text{per}}(e^{j\omega})\Big\} &= 0\end{aligned} \tag{158}$$

In other words, we want the periodogram to be a Consistent Estimate of the power spectral density. **To state the conclusion first**: the first condition holds, but the second does not.

Specifically, consider the **first condition**, asymptotic unbiasedness. Starting from the sample autocorrelation, its expectation is as shown in Equation 142. From the derivation of the periodogram, we obtain:

$$\begin{aligned}E\Big\{\hat{P}_{\text{per}}(e^{j\omega})\Big\} &= E\left\{\sum_{k=-N+1}^{N-1} \hat{r}_x(k)e^{-jk\omega}\right\} = \sum_{k=-N+1}^{N-1} E\{\hat{r}_x(k)\}e^{-jk\omega} \\ &= \sum_{k=-N+1}^{N-1} w_B(k)r_x(k)e^{-jk\omega}\end{aligned} \tag{159}$$

This is the Fourier transform of the product of the autocorrelation function and a window function. Thus, from the frequency-domain convolution property:

$$E\Big\{\hat{P}_{\text{per}}(e^{j\omega})\Big\} = \frac{1}{2\pi} P_x(e^{j\omega}) * W_B(e^{j\omega}) \tag{160}$$

Where the frequency-domain expression for the Bartlett window is:

$$W_B(e^{j\omega}) = \frac{1}{N}\left[\frac{\sin(N\omega/2)}{\sin(\omega/2)}\right]^2 \tag{161}$$

Note that the window function here is applied to the autocorrelation sequence, so it is called a Lag window, to distinguish it from the Data window applied directly to the data later. This is a conceptual distinction with no practical significance.

Note that as $N \to \infty$, $W_B(e^{j\omega})$ converges to an impulse function (with a periodic integral of 2π and area concentrated at the origin, i.e., $2\pi\delta(\omega)$; proof omitted). Thus, the periodogram satisfies the condition for asymptotic unbiasedness.

From the perspective of spectral plots, what was originally an assembly of ideal impulse functions to form the entire spectrum now becomes an assembly using the Bartlett window function. The more samples, the closer the window function is to an ideal impulse signal, and the more accurate the spectral estimate, as shown in Figure 7.

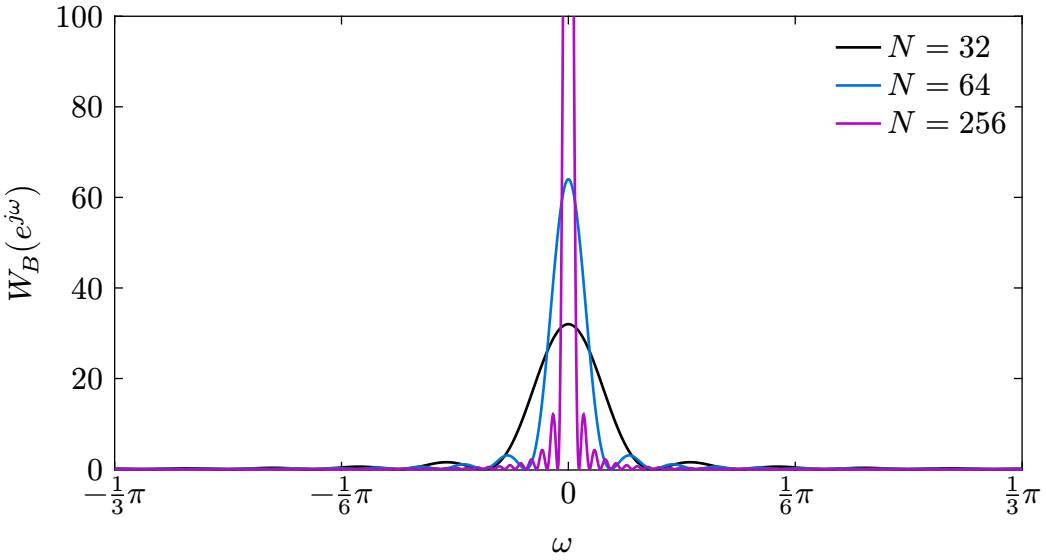

Figure 7: The Fourier transforms of Bartlett windows

The main lobe bandwidth of this function is approximately $\frac{2\pi}{N}$. Two impulse signals that are too close to each other may result in the overlap and merging of two peaks after convolution with the window function, making them impossible to distinguish clearly. Therefore, the resolution is defined as the 6dB bandwidth of the window function here:

$$\operatorname{Res}\left[\hat{P}_{\text{per}}(e^{j\omega})\right] = (\Delta\omega)_{6\text{dB}} = 0.89\frac{2\pi}{N} \tag{162}$$

-6dB is approximately 0.5, meaning that at about this position, the value where the two peaks overlap is half of the peak value. Considering the convolution process in Equation 160, this will result in only one peak remaining after the two peaks overlap, making the result indistinguishable.

Consider a detail in actual calculation: the numerical value of $\Delta\omega$ given here is in the sense of measurement after the Fourier transform is normalized to the interval $[-\pi, \pi]$.

Usually, our indicators are in actual spectrum units, in which case we need to divide by the sampling rate. For example, for a signal with a sampling rate of 10kHz, if we specify that the resolution must reach at least 10Hz, then $\Delta\omega$ should be $2\pi \times \frac{10\text{Hz}}{10\text{kHz}}$.

Second condition: Consider whether the variance tends to zero. Since the periodogram has a second-order relationship with the samples, calculating the variance now involves the calculation of the fourth-order moment of the stochastic process, which is too complex. However, we can consider the special case where the stochastic process is Gaussian white noise with variance σ_x^2. After a series of calculations (see Section 8.2.2, page 404 of the reference book), the variance in this case is found to be:

$$\operatorname{Var}\left\{\hat{P}_{\text{per}}(e^{j\omega})\right\} = \sigma_x^4 \tag{163}$$

This is independent of N and will not converge to zero as it grows. In fact, if we consider the general case, we have the following approximation (which is $\left(\sigma_x^2\right)^2$ for the case of Gaussian white noise):

$$\operatorname{Var}\left\{\hat{P}_{\text{per}}(e^{j\omega})\right\} \approx P_x^2(e^{j\omega}) \tag{164}$$

Therefore, the conclusion is that the second condition is not satisfied, meaning **the periodogram is not a consistent estimate of the power spectral density**.

9.1.2. Modified Periodogram

Naturally, we think about making some improvements. Let's set aside the previous derivations for a moment and directly consider modifying the definition, then verify the results. Looking back at the definition of the periodogram:

$$\hat{P}_{\text{per}}(e^{j\omega}) = \frac{1}{N}\left|X_N(e^{j\omega})\right|^2 = \frac{1}{N}\left|\sum_{n=-\infty}^{\infty} x[n]w_R[n]e^{-j\omega n}\right|^2 \tag{165}$$

The formula reflects the process of performing spectral estimation after applying a Dirichlet kernel (i.e., a rectangular window, $w_R[n]$) to the original signal. An intuitive thought is: what would be the effect if a different window function were used here?

(TODO) Pages 408 and 409 of the book derive the expectation and variance. Note that there seems to be an inconsistency in the definition of $w_B(k)$ in the book; we use the normalized $w_B(k) = \frac{1}{N}w_R(k) * w_R(-k) = \sum_{n=-\infty}^{\infty} w_R(k)w_R(n-k)$ here to remain consistent with the previous sections, so there will be a slight coefficient difference from the book that needs to be adjusted.

We define the Modified periodogram as:

$$\hat{P}_M(e^{j\omega}) = \frac{1}{NU}\left|\sum_{n=-\infty}^{\infty} x[n]w[n]e^{-jn\omega}\right|^2 \tag{166}$$

Where N is the length of the window function, and the constant U is the power of the window function, which is the average of energy over time (it will later be explained that this is to make the modified periodogram asymptotically unbiased):

$$U = \frac{1}{N}\sum_{n=0}^{N-1}|w[n]|^2 \tag{167}$$

9.1.2.1. Performance of Modified Periodogram

Similarly, we evaluate the performance of the modified periodogram. First is the Bias; from a similar derivation, we have:

$$E\left\{\hat{P}_M(e^{j\omega})\right\} = \frac{1}{2\pi NU}P_x(e^{j\omega}) * \left|W(e^{j\omega})\right|^2 \tag{168}$$

Where $W(e^{j\omega})$ is the Fourier transform of $w[n]$. From the previous setting of U, we have:

$$U = \frac{1}{N}\sum_{n=0}^{N-1}|w[n]|^2 = \frac{1}{2\pi N}\int_{-\pi}^{\pi}\left|W(e^{j\omega})\right|^2 \mathrm{d}\omega \tag{169}$$

That is:

$$\int_{-\pi}^{\pi}\frac{1}{2\pi NU}\left|W(e^{j\omega})\right|^2 \mathrm{d}\omega = 1 \tag{170}$$

This makes $E\left\{\hat{P}_M(e^{j\omega})\right\}$ tend toward the power spectral density as $N \to \infty$, making it asymptotically unbiased, which is the purpose of setting U this way.

Next is the variance. Adding a data window does not help reduce variance, so just like Equation 164:

$$\mathrm{Var}\left\{\hat{P}_M(e^{j\omega})\right\} \approx P_x^2(e^{j\omega}) \tag{171}$$

That is, the modified periodogram is also not a consistent estimate of the power spectral density.

9.1.2.2. Trade-off between Resolution and Confusion

If it doesn't affect the bias and variance of the estimate, what exactly does adding a data window affect?

The Fourier transforms of different data windows have different shapes, mainly reflected in the Main lobe and Sidelobe. Referring to Section 4.1, the former will affect the resolution of the spectral estimate, while the latter will introduce sidelobe interference and confusion.

We define resolution as the 3dB bandwidth of the data window's main lobe; a larger value indicates less clarity:

$$\mathrm{Res}\left[\hat{P}_{\mathrm{per}}(e^{j\omega})\right] = (\Delta\omega)_{3\mathrm{dB}} \tag{172}$$

Note that the resolution defined when analyzing the Periodogram earlier was the 6dB bandwidth of the Bartlett window instead of 3dB, but they are actually consistent.

This is because the previous analysis was on the window applied to the autocorrelation sequence (i.e., the lag window); note that in Equation 160, the term convolved with the power spectral density is $W_B(e^{j\omega})$. In contrast, the analysis here is on the data window; note that in Equation 168, the term convolved with the power spectral density is $\frac{1}{NU}\left|W(e^{j\omega})\right|^2$. There is a square relationship between the two.

Therefore, the -6dB point of the former is consistent with the -3dB point of the latter, both being the half-power points relative to the signal.

The approximate values for sidelobe suppression and resolution of common window functions are summarized in Table 1.

	Sidelobe (dB)	Resolution
Rectangular	-13	$0.89(2\pi/N)$
Bartlett	-27	$1.28(2\pi/N)$
Hanning	-32	$1.44(2\pi/N)$
Hamming	-43	$1.30(2\pi/N)$
Blackman	-58	$1.68(2\pi/N)$

Table 1: Properties of a few commonly used windows with length N

It can be observed that often, the better the sidelobe suppression, the worse the resolution (i.e., the larger the 3dB bandwidth). This is a Trade-off.

9.1.3. Periodogram Averaging

Thus far, none of the above methods can provide a consistent estimate of the power spectral density due to non-convergent variance. In the following, we obtain the desired consistent estimate through several methods of averaging periodograms.

Consider that previously for a random variable x, we obtained a consistent estimate of its mean $E\{x\}$ by collecting a large number of uncorrelated measurement samples and calculating the

sample mean. By analogy, theoretically, we need to use multiple uncorrelated realizations of the stochastic process $x(n)$, calculate the periodogram for each, and then average them to estimate the expectation of the periodogram.

Specifically, suppose we have K uncorrelated realizations $x_i[n]$, each of length L, with the total number of sample points being $N = LK$. Calculate the periodogram for each realization:

$$\hat{P}_{\text{per}}^{(i)}(e^{j\omega}) = \frac{1}{L}\left|\sum_{n=0}^{L-1} x_i[n]e^{-jn\omega}\right|^2, \quad i = 0, 1, ..., K-1 \tag{173}$$

Then average them to get the final spectral estimate:

$$\hat{P}_x(e^{j\omega}) = \frac{1}{K}\sum_{i=0}^{K-1}\hat{P}_{\text{per}}^{(i)}(e^{j\omega}) = \frac{1}{N}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} x_i[n]e^{-jn\omega}\right|^2 \tag{174}$$

As usual, we evaluate its bias and variance. First, because it is just another average, the expectation is the same as before:

$$E\left\{\hat{P}_x(e^{j\omega})\right\} = E\left\{\hat{P}_{\text{per}}^{(i)}(e^{j\omega})\right\} = \frac{1}{2\pi}P_x(e^{j\omega}) * W_B(e^{j\omega}) \tag{175}$$

Thus, it is asymptotically unbiased as $L \to \infty$. Then consider the variance; since the different realizations are uncorrelated:

$$\begin{aligned}\text{Var}\left\{\hat{P}_x(e^{j\omega})\right\} &= \frac{1}{K^2}\text{Var}\left\{\sum_{i=0}^{K-1}\hat{P}_{\text{per}}^{(i)}(e^{j\omega})\right\} \\ &= \frac{1}{K}\text{Var}\left\{\hat{P}_{\text{per}}^{(i)}(e^{j\omega})\right\} \approx \frac{1}{K}P_x^2(e^{j\omega})\end{aligned} \tag{176}$$

In summary, using this averaging-based method can provide a **consistent estimate** of the power spectral density when both L and K tend to infinity.

9.1.3.1. Bartlett's Method

In general practical situations, we do not have that many independent realizations. However, if we have a sufficiently long realization and the underlying stochastic process satisfies the ergodicity assumption, we can cut it into small segments and use them as uncorrelated realizations to obtain a spectral estimate, known as Bartlett's method.

Let the signal length be N, cut into K non-overlapping segments (to ensure they are uncorrelated as much as possible), each of length L. If we further let:

$$x_i[n] = x[n + iL], \quad n = 0, 1, ..., L-1; \quad i = 0, 1, ..., K-1 \tag{177}$$

The notation becomes consistent with the previous analysis. We directly obtain the formula:

$$\begin{aligned}\hat{P}_B(e^{j\omega}) &= \frac{1}{K}\sum_{i=0}^{K-1}\left(\frac{1}{L}\left|\sum_{n=0}^{L-1}x[n+iL]e^{-jn\omega}\right|^2\right) \\ &= \frac{1}{N}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1}x[n+iL]e^{-jn\omega}\right|^2\end{aligned} \tag{178}$$

The bias and variance are consistent with the previous analysis of the averaging method. However, even if they do not overlap, there will certainly be some correlation between the segmented sequences (ergodicity only guarantees that correlation gradually vanishes over a long time), so the variance might be slightly smaller than analyzed previously. Nevertheless, we still approximately consider the segments to be uncorrelated and take this approximate value for the variance.

Next, let's analyze the impact of this method on resolution. Since the original sequence of length N is cut into small segments of length L, the actual sequence length used when calculating the periodogram is only L, so the resolution is:

$$\mathrm{Res}\left[\hat{P}_B(e^{j\omega})\right] = 0.89\frac{2\pi}{L} = 0.89K\frac{2\pi}{N} \tag{179}$$

It can be seen that compared to calculating the periodogram using the entire sequence of length N, the resolution has deteriorated by a factor of K; this is the trade-off.

9.1.3.2. Welch's Method

Bartlett's method uses periodogram averaging. Next, we follow this idea but use modified periodogram averaging, known as Welch's method.

The idea of the modified periodogram is to apply a window function to the data, which can actually weaken the correlation of the signal at the edges of two adjacent segments. Therefore, we can consider relaxing the requirements and allow for **overlapping** when segmenting the data. Each segment is still of length L, but the starting points of each segment are only spaced by D samples (overlap occurs when $D < L$):

$$x_i[n] = x[n+iD], \quad n = 0, 1, ..., L-1; \quad i = 0, 1, ..., K-1 \tag{180}$$

The total number of sample points is then $N = L + D(K-1)$. Typically, we use a 50% overlap, i.e., $D = L/2$. The data window is applied to each segment, and the window length is consistent with the segment length L. Thus, we obtain the spectral estimate for Welch's method as:

$$\begin{aligned}\hat{P}_W(e^{j\omega}) &= \frac{1}{K}\sum_{i=0}^{K-1}\left(\frac{1}{LU}\left|\sum_{n=0}^{L-1}w[n]x[n+iD]e^{-jn\omega}\right|^2\right) \\ &= \frac{1}{KLU}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1}w[n]x[n+iD]e^{-jn\omega}\right|^2\end{aligned} \tag{181}$$

Analyzing the indicators as usual: first, the bias is consistent with the Modified periodogram, substituting N with L:

$$E\left\{\hat{P}_W(e^{j\omega})\right\} = E\left\{\hat{P}_M^{(i)}\right\} = \frac{1}{2\pi LU} P_x(e^{j\omega}) * \left|W(e^{j\omega})\right|^2 \tag{182}$$

The variance is related to the degree of overlap and is difficult to calculate; we only consider the case of 50% overlap:

$$\mathrm{Var}\left\{\hat{P}_W(e^{j\omega})\right\} \approx \frac{9}{8K} P_x^2(e^{j\omega}) \approx \frac{9}{16}\frac{L}{N} P_x^2(e^{j\omega}) \tag{183}$$

The coefficient $9/8$ seems to indicate that the variance performance has slightly worsened, but since the number of segments $K \approx \frac{2N}{L}$ has nearly doubled, the variance performance is actually improved.

9.1.4. Periodogram-based Methods Summary

The reference book also mentions the Blackman-Tukey method, which is omitted here. Aside from that, we have analyzed the mean, variance, resolution, and other performances of each method. A summary is provided in Figure 8.

	Definition $\hat{P}_x(e^{j\omega})$	Expectation	Variance (approx.)	Resolution $\Delta\omega$
Periodogram	$\frac{1}{N}\left\|\sum_{n=-\infty}^{\infty} x[n]w_R[n]e^{-j\omega n}\right\|^2$	$\frac{1}{2\pi}P_x(e^{j\omega}) * W_B(e^{j\omega})$	$P_x^2(e^{j\omega})$	$0.89\frac{2\pi}{N}$
Modified periodogram	$\frac{1}{NU}\left\|\sum_{n=-\infty}^{\infty} x[n]w[n]e^{-j\omega n}\right\|^2$	$\frac{1}{2\pi NU}P_x(e^{j\omega}) * \left\|W(e^{j\omega})\right\|^2$	$P_x^2(e^{j\omega})$	See Table 1
Bartlett's $N = KL$	$\frac{1}{N}\sum_{i=0}^{K-1}\left\|\sum_{n=0}^{L-1} x[n+iL]e^{-j\omega n}\right\|^2$	$\frac{1}{2\pi}P_x(e^{j\omega}) * W_B(e^{j\omega})$	$\frac{1}{K}P_x^2(e^{j\omega})$	$0.89K\frac{2\pi}{N}$
Welch's (50% overlap) $N = L + D(K-1)$	$\frac{1}{KLU}\sum_{i=0}^{K-1}\left\|\sum_{n=0}^{L-1} w[n]x[n+iD]e^{-j\omega n}\right\|^2$	$\frac{1}{2\pi LU}P_x(e^{j\omega}) * \left\|W(e^{j\omega})\right\|^2$	$\frac{9}{16}\frac{L}{N}P_x^2(e^{j\omega})$	Window dependent

Figure 8: Properties of a few commonly used windows with length N

Note that in the modified periodogram $U = \frac{1}{N}\sum_{n=0}^{N-1}|w[n]|^2$, whereas in Welch's method, since each segment is of length L, its $U = \frac{1}{L}\sum_{n=0}^{L-1}|w[n]|^2$.

The reference book defines two indicators to measure the performance of the above methods. The first is Variability:

$$\mathcal{V} = \frac{\mathrm{Var}\left\{\hat{P}_x(e^{j\omega})\right\}}{E^2\left\{\hat{P}_x(e^{j\omega})\right\}} \tag{184}$$

Simply put, it is the normalized variance. The second is the Figure of merit:

$$\mathcal{M} = \mathcal{V}\Delta\omega \tag{185}$$

Which is the product of variability and resolution; the smaller this value, the better. Incidentally, such indicators defined as the product of two quantities generally multiply the variables involved in a trade-off. Consequently, we will find that the figures of merit for these nonparametric estimation methods are quite similar.

9.1.5. Minimum Variance (MV) Spectrum Estimation

First, we expand on the idea of feeding signals into filter banks as discussed in Section 9.1.1.1. In 凵 section, the filters were fixed and independent of the data $x[n]$, termed "data independent." In such cases, if some filters happen to pass too much energy through their sidelobes, it leads to significant interference.

The Minimum Variance (MV) Spectrum Estimation method introduced in this section aims to design a filter for each frequency point based on the input signal $x[n]$, such that each filter: 1. Has a gain of unity at the target frequency ω_i, passing it without loss; 2. Minimizes the energy passed through the sidelobes. This yields better estimation results.

We first define notation: let $g_i[n]$ be a p-th order complex-valued FIR band-pass filter. To satisfy the **first requirement**, we should have:

$$G_i(e^{j\omega_i}) = \sum_{n=0}^{p} g_i[n]e^{-jn\omega_i} = 1 \tag{186}$$

For convenience, we write this in vector form. Let $\boldsymbol{g}_i = [g_i[0], g_i[1], ..., g_i[p]]^T$ and $\boldsymbol{e}_i = \left[1, e^{j\omega_i}, ..., e^{jp\omega_i}\right]^T$. The constraint becomes:

$$\boldsymbol{g}_i^H \boldsymbol{e}_i = \boldsymbol{e}_i^H \boldsymbol{g}_i = 1 \tag{187}$$

Note that while $\boldsymbol{g}_i^H \boldsymbol{e}_i = \left(\boldsymbol{e}_i^H \boldsymbol{g}_i\right)^*$, it is set to 1 here. Since the result is known to be real, this is acceptable, but remember this equality only holds here.

The **second requirement** involves minimizing the power of the output process. Regarding this power value, from Equation 34:

$$E\left\{|y_i[n]|^2\right\} = \boldsymbol{g}_i^H \boldsymbol{R}_x \boldsymbol{g}_i \tag{188}$$

Thus, we want to minimize this value while satisfying the aforementioned linear constraint, i.e., solve:

$$\min_{\boldsymbol{g}_i} \boldsymbol{g}_i^H \boldsymbol{R}_x \boldsymbol{g}_i \quad \text{s.t. } \boldsymbol{e}_i^H \boldsymbol{g}_i = 1 \tag{189}$$

The solution to this problem is:

$$\begin{aligned} \boldsymbol{g}_i &= \frac{\boldsymbol{R}_x^{-1} \boldsymbol{e}_i}{\boldsymbol{e}_i^H \boldsymbol{R}_x^{-1} \boldsymbol{e}_i} \\ \min_{\boldsymbol{g}_i} \boldsymbol{g}_i^H \boldsymbol{R}_x \boldsymbol{g}_i &= \frac{1}{\boldsymbol{e}_i^H \boldsymbol{R}_x^{-1} \boldsymbol{e}_i} \end{aligned} \tag{190}$$

This is a typical optimization problem that can be solved using the Lagrange multiplier method. The solution process is as follows. Let:

$$L(\boldsymbol{g}_i, \lambda) = \boldsymbol{g}_i^H \boldsymbol{R}_x \boldsymbol{g}_i - \lambda\left(\boldsymbol{e}_i^H \boldsymbol{g}_i - 1\right) \tag{191}$$

Setting the partial derivatives with respect to the two parameters to zero yields (since $\boldsymbol{R}_x$ is Hermitian):

$$\begin{cases} 2\boldsymbol{R}_x \boldsymbol{g}_i - \lambda \boldsymbol{e}_i = 0 \\ \boldsymbol{e}_i^H \boldsymbol{g}_i = 1 \end{cases} \tag{192}$$

From the first equation, we have $\boldsymbol{g}_i = \frac{\lambda}{2}\boldsymbol{R}_x^{-1}\boldsymbol{e}_i$. Substituting this into the second equation and rearranging gives:

$$\frac{\lambda}{2} = \frac{1}{\boldsymbol{e}_i^H \boldsymbol{R}_x^{-1} \boldsymbol{e}_i} \tag{193}$$

Substituting this back into the previous equation yields the solution:

$$\boldsymbol{g}_i = \frac{\boldsymbol{R}_x^{-1}\boldsymbol{e}_i}{\boldsymbol{e}_i^H \boldsymbol{R}_x^{-1} \boldsymbol{e}_i} \tag{194}$$

Substituting this back gives the analytical expression for the minimum value.

Since the derivation above holds for any ω_i, we can directly drop the subscripts and write it as a function of ω:

$$\hat{\sigma}_x^2(\omega) = \frac{1}{\boldsymbol{e}^H \boldsymbol{R}_x^{-1} \boldsymbol{e}} \tag{195}$$

Where $\boldsymbol{e} = \begin{bmatrix} 1 & e^{j\omega} & e^{j2\omega} & \dots & e^{jp\omega} \end{bmatrix}^T$. The corresponding filter parameters for this process are:

$$\boldsymbol{g} = \frac{\boldsymbol{R}_x^{-1}\boldsymbol{e}}{\boldsymbol{e}^H \boldsymbol{R}_x^{-1} \boldsymbol{e}} \tag{196}$$

However, $\hat{\sigma}_x^2(\omega)$ is only an estimate of the output process power; it **cannot yet be used directly as a power spectral density estimate**. We must divide by the filter bandwidth, for reasons analogous to Equation 153. There are multiple ways to define bandwidth; we can take the simplest one, which provides correct estimation results for the white noise example (see the example on page 429 of the book):

$$\frac{\Delta}{2\pi} = \frac{1}{p+1} \tag{197}$$

Thus, we finally obtain the power spectral density estimate:

$$\hat{P}_{\text{MV}}(e^{j\omega}) = \frac{\hat{\sigma}_x^2(\omega)}{\Delta/2\pi} = \frac{p+1}{\boldsymbol{e}^H \boldsymbol{R}_x^{-1} \boldsymbol{e}} \tag{198}$$

This is called the **minimum variance spectrum estimate**. Note that it uses the autocorrelation matrix $\boldsymbol{R}_x$ of the stochastic process. If we only have sample data, we need to use the estimated $\hat{\boldsymbol{R}}_x$:

$$\begin{gathered} \hat{\boldsymbol{R}}_x = \frac{1}{K} \sum_{i=0}^{K-1} \boldsymbol{x}_i \boldsymbol{x}_i^H \\ \boldsymbol{x}_i = [x[i] \quad x[i+1] \quad x[i+2] \quad \dots \quad x[i+L-1]]^T \end{gathered} \tag{199}$$

Here, we segment the signal samples of length N into K overlapping segments, each of length L, with starting point intervals of only $D = 1$, i.e., $K = N - L + 1$, to estimate the sample autocorrelation matrix. **For dimension matching, the previous filter order is related to the sequence length by $\boldsymbol{L = p + 1}$**, yielding:

$$\hat{P}_{\text{MV}}(e^{j\omega}) = \frac{L}{\boldsymbol{e}^H \boldsymbol{R}_x^{-1} \boldsymbol{e}} \tag{200}$$

(TODO) Tired. Let it be.

9.2. Parametric Spectrum Estimation

9.2.1. For Autoregressive (AR) Models

(TODO) the Yule-Walker Method (autocorrelation method) and the covariance method.

9.2.2. Mulitple Signal Clasification (MUSIC)

(TODO)

10. Optimum Filtering

(TODO)

10.1. FIR Wiener Filter

10.1.1. Wiener-Hopf Equations

10.2. Discrete Kalman Filter

11. Adaptive Filtering

(TODO)

11.1. Least Mean Squares (LMS) Algorithm

11.2. Recursive Least Squares (RLS) Algorithm