Array Processing

Jacob Reinhold
jreinhold@gmail.com

Abstract

Array processing is concerned with extracting information from signals via an array of sensors. The most prominent problem in the field is to determine the location of an energy-radiating source relative to the location of the array, i.e., the estimation of the direction-of-arrival of a signal in the presence of noise and interfering signals. In this paper, we review the concepts necessary to study array processing and give an overview of the subject.

Contents

1 Introduction

The motivation to create this document was to create a streamlined document that gave the basics necessary to work in the domain of array processing. Further, having a searchable reference is often useful to recall obscure definitions and which makes understanding papers easier when starting out.

We assume that the reader is familiar with linear algebra and undergraduate probability/statistics; the first section and appendices have some information on the subjects. To learn array processing, we must first review probability and stochastic processes, as well as ways to analyze those processes. The probability and stochastic processes section requires some knowledge of real analysis. If you are familiar with probability and stochastic processes in an engineering or physics capacity, then you might just skip the section. Next, we build the filtering systems that will be used in array processing, just without the arrays. Following that, we talk about spectrum estimation, which is at the heart of most array processing applications. Finally, we use all of the previous information in the array processing domain.

These notes are generally structured around the textbook: “Statistical and Adaptive Signal Processing” by Manolakis, et al. [1] and “Practical Array Processing” by Sullivan [2]. Some additional information comes from “Discrete Random Signals and Statistical Signal Processing” by Therrien [3]. Much of the probability section is lifted from course notes from a measure-theoretic probability course taught by Žitković at the University of Texas at Austin in Fall 2016 [4]. Some other sources are used, but to a lesser extent and are additionally listed in the references.

2 Probability

This will be a very high-level overview of some of the core concepts that will be used throughout these notes. If any of the terms are used without definition, then you can probably skip over them since they aren’t very important to the concept (they are only included for correctness and rigour). This section is sort of a hodgepodge of measure-theory and practical engineering probability, so your mileage may vary.

Let’s get some notation out of the way:

- Let $\mathcal{L}^0$ be the set of all (Lebesgue) measurable functions. Note that this is a vector space.
- Let $\mathcal{L}^p$ be the set of all measurable functions on a measure space, e.g., $(S, \mathcal{S}, \mu)$, satisfying the following condition:
$$\int |f|^p \, d\mu < \infty.$$

Let’s begin this section with a rigorous mathematical definition of a probabilistic model. Let Ω represent the sample space, $\mathcal{F}$ be a family of events (a σ-algebra on Ω), and $\mathbb{P}$ be a probability measure (a set function mapping $\mathcal{F} \to [0, \infty]$) on $\mathcal{F}$. A *probability space* is defined as the triple $(\Omega, \mathcal{F}, \mathbb{P})$. Note that we do not generally work with the sample space Ω since it is often too large for analysis. Instead, we focus on studying mappings from Ω to the real numbers, i.e., a *random variable.*

The *distribution* of a random variable, say $X : \Omega \to S$ for some measurable space $(S, \mathcal{S})$, is defined to be the measure μ_X on $\mathcal{S}$; $\mu_X(A) = \mathbb{P}[X \in A] = \mathbb{P}[X^{-1}(A)]$ for $A \in \mathcal{S}$.

It is generally easier to work with the *(cumulative) distribution function* (cdf) of a random variable X, which is defined as:

$$F_X(x) := \mathbb{P}[X \leq x] = \mathbb{P}[X^{-1}((-\infty, x])].$$

If X is a random variable such that the distribution μ_X is absolutely continuous to the Lebesgue measure λ on the Borel set $\mathcal{B}(\mathbb{R})$[1], then the *probability density function* (pdf) f_X is defined to be the (Radon-Nikodym) derivative $\frac{d\mu_X}{d\lambda}$. You can forget all that mumbo-jumbo and think of the pdf as the derivative of the cdf[2].

As an aside, here is some more intuition about Ω taken from `here`:

> One should think of the sample space Ω as a source of all randomness in the system: the elementary event $\omega \in \Omega$ is chosen by a process beyond our control and the exact value of ω is assumed to be unknown. All the other parts of the system are possibly complicated, but deterministic, functions of ω (random variables).

2.1 Definitions

I'm just going to list a bunch of definitions here to speed up the process.

Definition 2.1. The Lebesgue integral with respect to the probability measure $\mathbb{P}$ is called *expectation* and is denoted by $\mathbb{E}$, defined as

$$\mathbb{E}[X] := \int X d\mathbb{P} = \int_\Omega X(\omega)\, \mathbb{P}[d\omega].$$

If X has a pdf f, then $\mathbb{E}[X] = \int_{-\infty}^{\infty} x\, f(x)\, dx$. Note that the expectation operator is linear (among other properties).

Definition 2.2. The *variance* of the random variable X is defined as

$$\mathrm{Var}[X] \equiv \sigma_X^2 := \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2.$$

The variance describes the spread of the distribution.

Definition 2.3. The *skewness* of a random variable X is defined as

$$\mathrm{Skew}[X] := \mathbb{E}\left[\left(\frac{X - \mathbb{E}[X]}{\sigma_X}\right)^3\right]$$

This value describes how the distribution leans (positive is to the right of the mean, negative is to the left of the mean).

[1] the σ-algebra generated by the open sets of $\mathbb{R}$

[2] the Radon-Nikodym derivative being a function that integrates nicely

Definition 2.4. The *kurtosis* of a random variable X is defined as

$$\mathrm{Kurt}[X] := \mathbb{E}\left[\left(\frac{X-\mathbb{E}[X]}{\sigma_X}\right)^4\right]$$

Kurtosis describes how a distribution looks insofar as flatness or pointedness (negative for flat, and positive for pointed).

Definition 2.5. The *characteristic function* of a random variable X is the function $\varphi_X : \mathbb{R} \to \mathbb{C}$ given by

$$\varphi_X(t) := \mathbb{E}[e^{itX}].$$

If X admits a density, then $\varphi_X(t) = \int_{-\infty}^{\infty} e^{itx} f_X(x)\, dx$.[1]

Remark. Let X and Y be independent random variables, and let $Z = X + Y$. Then $\mu_Z = \mu_X * \mu_Y$ where $*$ is convolution. It follows that $F_Z(z) = \int_{\mathbb{R}} F_X(z-y) dF_Y(y)$ (Riemann-Stieltjes integral—don't worry about it, if you are curious look `here`). A more useful, and easily remembered, formulation is $f_Z(z) = (f_X * f_Y)(z)$.

The characteristic function transforms this convolution into multiplication, i.e., $\varphi_Z = \varphi_X \cdot \varphi_Y$.

Definition 2.6. The *covariance* of two random variables X and Y (on the same sample space) is defined by

$$\begin{aligned}\mathrm{Cov}(X,Y) &= \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])^*] \\ &= \mathbb{E}[XY^*] - \mathbb{E}[X]\mathbb{E}[Y]^*.\end{aligned}$$

Definition 2.7. The *correlation* of two random variables X and Y (on the same sample space) is defined by

$$\mathrm{Cor}(X,Y) = \mathbb{E}[XY^*] = \mathrm{Cov}(X,Y) + \mathbb{E}[X]\mathbb{E}[Y]^* \quad (= \langle X, Y\rangle).$$

If $\mathrm{Cor}(X,Y) = 0$, then X and Y are said to be orthogonal.

2.2 Random Vectors

A real-valued vector containing n random variables is called a *random vector* and is denoted by

$$\mathbf{X} = [X_1, X_2, \ldots, X_n].$$

Clearly, $\mathbf{X} \in \mathbb{R}^n$.

The *joint distribution* describes the relationship between elements of a random vector (or more generally any two or more random variables). The measure μ_X on $\mathcal{B}(\mathbb{R}^n)$ given by

$$\mu_{\mathbf{X}}(B) = \mathbb{P}[\mathbf{X} \in B]$$

is the *distribution* of the random vector $\mathbf{X}$. The *marginal distribution* is the distribution of one variable in the vector, i.e.,

$$\begin{aligned}\mu_{X_1}(A) &= \mathbb{P}[X_1 \in A] = \mathbb{P}[X_1 \in A, X_2 \in \mathbb{R}, \ldots, X_n \in \mathbb{R}] \\ &= \mu_{\mathbf{X}}(A \times \mathbb{R} \times \cdots \times \mathbb{R}).\end{aligned}$$

[1] essentially this is the Fourier transform

The *joint cdf* is denoted as $F_{\mathbf{X}}(\mathbf{x})$ and the *joint pdf* is denoted as $f_{\mathbf{X}}(\mathbf{x})$. The marginal density function is defined as

$$f_{x_i}(x) = \underbrace{\int \cdots \int}_{(n-1)} f_{\mathbf{X}}(\mathbf{x})\, dx_1 \cdots dx_{i-1}\, dx_{i+1} \cdots dx_n.$$

Note that $F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} f_{\mathbf{X}}(\mathbf{t})\, d\mathbf{t}$ and $f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial}{\partial X_1} \cdots \frac{\partial}{\partial X_n} F_{\mathbf{X}}(\mathbf{x})$.

2.3 Independence

Note that the σ-algebra generated by a random variable, say $X : \Omega \to \mathbb{R}^n$, is defined to be

$$\sigma(X) := \{X^{-1}(B) \mid B \in \mathcal{B}(\mathbb{R}^n)\},$$

and it is the smallest σ-algebra such that X is measurable[1].

Let $(X, \cdot, \mu)$ be a measure space. Then any two σ-algebras $\mathcal{A}, \mathcal{B}$ on that measure space are said to be independent if for any $A \in \mathcal{A}$ and $B \in \mathcal{B}$, we have

$$\mu(A \cap B) = \mu(A)\mu(B).$$

Definition 2.8. Random variables $X_1, \ldots, X_n$ are said to be *independent* if the σ-algebras $\sigma(X_1), \ldots, \sigma(X_n)$ are independent.

2.4 Correlation and Covariance Matrices

Let $\mathbf{X}, \mathbf{Y}$ be random vectors of length M, N, respectively.

Definition 2.9. The *autocorrelation matrix* is defined by

$$\mathbf{R}_{\mathbf{X}} := \mathbb{E}[\mathbf{X}\mathbf{X}^H] = \begin{bmatrix} r_{11} & \cdots & r_{1M} \\ \vdots & \ddots & \vdots \\ r_{M1} & \cdots & r_{MM} \end{bmatrix}.$$

Note that $r_{ii} = \mathbb{E}[|X_i|^2]$ are the second-order moments and $r_{ij} = \mathbb{E}[X_i X_j^*] = r_{ji}^*$ measures the *correlation*.

Definition 2.10. The *autocovariance matrix* is defined by

$$\Gamma_{\mathbf{X}} := \mathbb{E}\left[(\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^H\right] = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1M} \\ \vdots & \ddots & \vdots \\ \gamma_{M1} & \cdots & \gamma_{MM} \end{bmatrix}.$$

Note that $\gamma_{ii} = \sigma_{X_i}^2$ and γ_{ij} measures the *covariance* between X_i and X_j.

[1] X is $(\mathcal{F}, \mathcal{B}(\mathbb{R}^n))$-measurable if $X^{-1}(B) \in \mathcal{F}$ for each $B \in \mathcal{B}(\mathbb{R}^n)$.

Definition 2.11. The *correlation coefficient* between X_i and X_j is defined to be

$$\rho_{ij} := \frac{\gamma_{ij}}{\sigma_i \sigma_j}.$$

This quantity measures the statistical similarity between two random variables.

Definition 2.12. The *cross-correlation matrix* is defined by

$$\mathbf{R}_{\mathbf{XY}} := \mathbb{E}[\mathbf{X}\,\mathbf{Y}^H] = \begin{bmatrix} \mathbb{E}[\mathbf{X}_1\mathbf{Y}_1^*] & \cdots & \mathbb{E}[\mathbf{X}_1\mathbf{Y}_N^*] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\mathbf{X}_M\mathbf{Y}_1^*] & \cdots & \mathbb{E}[\mathbf{X}_M\mathbf{Y}_N^*] \end{bmatrix}.$$

The *cross-covariance* matrix, $\Gamma_{\mathbf{XY}} := \mathbf{R}_{\mathbf{XY}} - \boldsymbol{\mu}_{\mathbf{X}}\boldsymbol{\mu}_{\mathbf{Y}}{}^H$. These quantities can be interpreted similarly to the corresponding auto- matrices.

2.5 Law of Large Numbers and Central Limit Theorem

Theorem 2.1 (Weak law of large numbers)**.** *Let $\{X_n\}_{n\in\mathbb{N}}$ be an iid[1] sequence of random variables with the (common) distribution μ and the characteristic function $\varphi = \varphi_\mu$ such that $\varphi'(0)$ exists. Then $c = -i\varphi'(0)$ is a real number[2] and*

$$\frac{1}{n}\sum_{k=1}^{n} X_k \to c \text{ in probability.}$$

Theorem 2.2 (Central Limit Theorem[3])**.** *Let $\{X_n\}_{n\in\mathbb{N}}$ be an iid sequence of random variables with $0 < \mathrm{Var}[X_1] < \infty$. Then*

$$\frac{\sum_{k=1}^{n}(X_k - \mu)}{\sqrt{\sigma^2 n}} \xrightarrow{\mathcal{D}} \chi,$$

where $\chi \sim N(0,1)$, $\mu = \mathbb{E}[X_1]$ and $\sigma^2 = \mathrm{Var}[X_1]$.

2.6 Some Other Definitions that Don't Quite Fit Anywhere Else

Definition 2.13. This definition comes from `here`. Let X_1 and X_2 be independent copies of a random variable X. Then X is said to be *stable* if for any constants $a > 0$ and $b > 0$ the random variable $aX_1 + bX_2$ has the same distribution as $cX + d$ for some constants $c > 0$ and d. The distribution is said to be *strictly stable* if this holds with $d = 0$.

Definition 2.14. $\|f\|_{\mathcal{L}^p} = \left(\int |f|^p \, d\mu\right)^{\frac{1}{p}}$.

Definition 2.15 (Conjugate exponents)**.** We say that $p, q \in [1, \infty]$ are *conjugate exponents* if $\frac{1}{p} + \frac{1}{q} = 1$

[1] independent and identically distributed

[2] $\mathbb{E}[X^k] = (-i)^k \varphi^{(k)}(0)$

[3] note that the random variables do not necessarily have to be identically distributed for the sequence to converge in distribution to normal; see the Lindeberg-Feller theorem

Proposition 2.1 (Hölder's inequality). *Let $p, q \in [1, \infty]$ be conjugate exponents. For $f \in \mathcal{L}^p$ and $g \in \mathcal{L}^q$, we have*

$$\int |fg| \, d\mu \leq \|f\|_{\mathcal{L}^p} \|g\|_{\mathcal{L}^q}.$$

The equality holds if and only if there exist constants $\alpha, \beta \geq 0$ with $\alpha + \beta > 0$ such that $\alpha |f|^p = \beta |g|^q$.

Corollary 2.2.1 (Cauchy-Schwarz inequality). *For $f, g \in \mathcal{L}^2$, we have*

$$\int |fg| \, d\mu \leq \|f\|_{\mathcal{L}^2} \|g\|_{\mathcal{L}^2}.$$

Corollary 2.2.2 (Minkowski's inequality). *For $f, g \in \mathcal{L}^p$, $p \in [1, \infty]$, we have*

$$\|f + g\|_{\mathcal{L}^p} \leq \|f\|_{\mathcal{L}^p} + \|g\|_{\mathcal{L}^p}.$$

Definition 2.16. A subset K of a vector space is said to be *convex* if $\alpha x + (1 - \alpha) y \in K$, whenever $x, y \in K$ and $\alpha \in [0, 1]$.

I'm putting stars around Jensen's (informal defintion) since it shows up everywhere.

Proposition 2.2 (Jensen's inequality (formal)). *Suppose that $\mu(S) = 1$ (i.e., μ is a probability measure) and that $\varphi : \mathbb{R} \to \mathbb{R}$ is a convex function. For a function $f \in \mathcal{L}^1$ we have $\varphi(f) \in \{f \in \mathcal{L}^0 \mid f^- \in \mathcal{L}^1\}$ and*

$$\int \varphi(f) \, d\mu \geq \varphi\left(\int f \, d\mu\right).$$

Proposition 2.3 (★ Jensen's inequality ★). *If φ is convex, then*

$$\mathbb{E}[\varphi(X)] \geq \varphi(\mathbb{E}[X]).$$

2.7 Conditional Expectation (and Probability)

Definition 2.17. The *indicator* function is defined as:

$$\mathbb{1}_A(x) = \begin{cases} 1 & x \in A, \\ 0 & x \notin A. \end{cases}$$

Definition 2.18 (Conditional Probability). For events A, B and $\mathbb{P}[B] > 0$, recall the conditional probability of A given B is defined to be

$$\mathbb{P}[A|B] = \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}.$$

Definition 2.19. Let $\mathcal{G}$ be a sub-σ-algebra of $\mathcal{F}$, and let $X \in \mathcal{L}^1$ be a random variable. Then the random variable ξ is the *conditional expectation of X with respect to $\mathcal{G}$*—and denote it by $\mathbb{E}[X|\mathcal{G}]$—if

1. $\xi \in \mathcal{L}^1$,
2. ξ is $\mathcal{G}$-measurable,
3. $\mathbb{E}[\xi \mathbb{1}_A] = \mathbb{E}[X \mathbb{1}_A]$ for all $A \in \mathcal{G}$.

Note that conditional expectation is linear.

Definition 2.20. Let $\mathcal{G}$ be a sub-σ-algebra of $\mathcal{F}$. The *conditional probability* of $A \in \mathcal{F}$, given $\mathcal{G}$—denoted by $\mathbb{P}[A|\mathcal{G}]$—is defined by

$$\mathbb{P}[A|\mathcal{G}] = \mathbb{E}[\mathbb{1}_A|\mathcal{G}].$$

Remark. When the random vector (X, Y) admits a joint density $f_{X,Y}(x,y)$, and $f_Y(y) > 0$, then the conditional density is defined as

$$f_{X|Y=y}(x) := \frac{f_{X,Y}(x,y)}{f_Y(y)}.$$

Then $\mathbb{P}[X \in A|Y = y] := \int_A f_{X|Y=y}(x,y)\,dx$.

Remark. With the introduction of conditional expectation, we can start to think of σ-algebras as representing information, i.e., the ability to answer questions. In probability, what we are interested in are the values of $\omega \in \Omega$ which actually cause an event. However, as stated before, Ω is too large for this kind of analysis. Thus we are interested in determining if the "true" ω is in some event, say A. Fix $\mathcal{G} \subseteq \mathcal{F}$.

1. $\omega \in \Omega$ so $\Omega \in \mathcal{G}$.
2. If the true $\omega \in A$, then $\omega \notin A^c$ and vice versa. So for $A \in \mathcal{G}$ we have $A^c \in \mathcal{G}$
3. Let $\{A_n\}_{n\in\mathbb{N}}$ be a sequence of events of which we know the answer to the question: "Is $\omega \in A_n$?" Then we know how to answer the question to the union of all such events. Thus $\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{G}$.

You have to know the definition of a σ-algebra, but—if you do—clearly we have shown that information can be represented by a σ-algebra.

3 Stochastic Processes

A *stochastic process* is a family of random variables $\{X_t\}_{t\in\mathcal{T}}$. When $\mathcal{T} = \mathbb{N}$, then $\{X_t\}_{t\in\mathcal{T}}$ is a *discrete-time* stochastic process. When $\mathcal{T} = [0, \infty)$, then it is called a *continuous-time* stochastic process.

Note that $\mathcal{T} \subseteq [0, \infty)$, so a stochastic process is just a generalization of a random vector/variable (if $\mathcal{T} = \{1\}$). In these notes, we will only talk about discrete-time stochastic processes so $\mathcal{T}$ will generally always equal $\mathbb{N}$ or equivalently $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

3.1 Second Moment Statistical Description

I need to define these here, but I will talk about what a "second moment" is in the next section.

Definition 3.1. The *autocorrelation sequence* of a discrete-time random process $\{X_n\}_{n\in\mathbb{N}}$ is defined as the joint moment of the random variables X_{n_1} and X_{n_2}, that is,

$$r_{XX}(n_1, n_2) = \mathbb{E}[X_{n_1} X_{n_2}^*].$$

This value provides a quantification of the dependence between values of the process at two different times (n_1 and n_2).

The *cross-correlation sequence* is defined similarly (change the second X to a Y from another stochastic process $\{Y_n\}_{n\in\mathbb{N}}$ defined on the same sample space).

Definition 3.2. The *autocovariance sequence* of $\{X_n\}_{n\in\mathbb{N}}$ is defined by

$$\begin{aligned}\gamma_{XX}(n_1, n_2) &= \mathbb{E}[(X_{n_1} - \mu_{X_{n_1}})(X_{n_2} - \mu_{X_{n_2}})^*]\\ &= r_{XX}(n_1, n_2) - \mu_{X_{n_1}}\mu^*_{X_{n_2}}\end{aligned}$$

The *cross-covariance sequence* is defined similarly (change the second X to a Y from another stochastic process $\{Y_n\}_{n\in\mathbb{N}}$ defined on the same sample space).

Definition 3.3. The *normalized cross-correlation* of two stochastic processes $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ is defined by

$$\rho_{XY}(n_1, n_2) = \frac{\gamma_{XY}(n_1, n_2)}{\sigma_X(n_1)\sigma_Y(n_2)}.$$

Definition 3.4.

- A stochastic process is called *independent* if it is sequence of independent random variables.
- A stochastic process is called *uncorrelated* if it is a sequence of uncorrelated random variables.
- A stochastic process is called *orthogonal* if it is a sequence of orthogonal random variables, i.e., $\mathbb{E}[X_i X_j^*] = 0$ for all $i, j \in \mathbb{N}$ where $i \neq j$ for a stochastic process $\{X_n\}_{n\in\mathbb{N}}$.

The above three definitions can be generalized to handle two random processes and the processes are labeled the same.

3.2 Stationarity

Definition 3.5. A random process $\{X_n\}_{n\in\mathbb{N}}$ is called *stationary* if statistics determined for X_i are equal to those for X_j for every $j \in \mathbb{N}$.

Definition 3.6. A stochastic process $\{X_n\}_{n\in\mathbb{N}}$ is called *stationary of order* N if

$$f_X(X_1, \ldots, X_N) = f_X(X_{1+k}, \ldots, X_{N+k})$$

for any value k. If $\{X_n\}_{n\in\mathbb{N}}$ is stationary for all order $N = 1, 2, \ldots$, then it is said to be *strict-sense stationary* (SSS). An example of a SSS process is an iid sequence.

Definition 3.7. A random signal $\{X_n\}_{n\in\mathbb{N}}$ is called *wide-sense stationary (WSS)* if

1. $\mathbb{E}[|X_i|^2] < \infty$, i.e., $\{X_n\}_{n\in\mathbb{N}} \in \mathcal{L}^2$.[1]
2. $\mathbb{E}[X_i] = \mu_X$ for all $i \in \mathbb{N}$.
3. $\mathrm{Var}[X_i] = \sigma_X^2$ for all $i \in \mathbb{N}$.

[1] the notation here is bad, but I mean each realization of the random process is in $\mathcal{L}^2$

4. The autocorrelation depends only on the distance $l = n_1 - n_2$, i.e.,

$$r_X(n_1, n_2) = r_X(n_1 - n_2) = r_X(l) = \mathbb{E}[X_{n+l}X_n^*] = \mathbb{E}[X_n X_{n-l}^*]$$

Remark. Note that SSS and WSS stochastic processes are martingales.

Definition 3.8. Two random signals $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are called *jointly wide-sense stationary* if each is WSS and their cross-correlation depends only on $l = n_1 - n_2$, i.e.,

$$r_{XY}(l) = \mathbb{E}[X_n Y_{n-l}^*] = r_{XY}(l) - \mu_X \mu_Y^*.$$

Here are some properties of WSS stochastic processes:

1. The average power of a WSS process $\{X_n\}_{n\in\mathbb{N}}$ satisfies[1]:

$$r_X(0) = \sigma_X^2 + |\mu_X|^2 \geq 0 \quad r_X(0) \geq |r_X(l)| \text{ for all } l.$$

2. The autocorrelation sequence $r_X(l)$ is a conjugate symmetric function of lag l, i.e.,

$$r_X^*(-l) = r_X(l).$$

3. The autocorrelation sequence $r_X(l)$ is positive semi-definite.

3.2.1 Moments

The first four moments of a stationary random process are

1. $\mu_x = \mathbb{E}[x(n)]$
2. $r_x(l) = \mathbb{E}[x^*(n)x(n+l)]$
3. $m_x^{(3)}(l_1, l_2) = \mathbb{E}[x^*(n)x(n+l_1)x(n+l_2)]$
4. $m_x^{(4)}(l_1, l_2, l_3) = \mathbb{E}[x^*(n)x^*(n+l_1)x(n+l_2)x(n+l_3)]$

3.3 Ergodicity

Ergodicity implies that all the statistical information can be obtained from any single representative member of the ensemble, where the *ensemble* is the set of all realizable time series of a stochastic process. Thus, so far, when we have referred to the expectation of a stochastic process, we actually been referring to the *ensemble average*. However, we would like to obtain statistical information from one realization instead of the whole ensemble. The only way to do this is through the *time average*, defined to be

$$\langle(\cdot)\rangle := \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} (\cdot) \quad \text{and} \quad \langle(\cdot)\rangle_N := \frac{1}{2N+1} \sum_{n=-N}^{N} (\cdot)$$

[1] $r_X(0)$ is the total average power of $\{X_n\}_{n\in\mathbb{N}}$

For every ensemble average, we can define a corresponding time average:

$$\begin{aligned}\text{Mean value} &= \langle X_n \rangle \\ \text{Mean square} &= \langle |X_n|^2 \rangle \\ \text{Variance} &= \langle |X_n - \langle X_n \rangle|^2 \rangle \\ \text{Autocorrelation} &= \langle X_n X_{n-l}^* \rangle \\ \text{Autocovariance} &= \langle (X_n - \langle X_n \rangle)(X_{n-l} - \langle X_n \rangle)^* \rangle \\ \text{Cross-correlation} &= \langle X_n Y_{n-l}^* \rangle \\ \text{Cross-covariance} &= \langle (X_n - \langle X_n \rangle)(Y_{n-l} - \langle Y_n \rangle)^* \rangle\end{aligned}$$

A random signal $\{X_n\}_{n\in\mathbb{N}}$ is called *ergodic* if its ensemble averages equal appropriate time averages. However, there are several degrees of ergodicity.

Definition 3.9. A random process $\{X_n\}_{n\in\mathbb{N}}$ is ergodic *in the mean* if

$$\langle X_n \rangle = \mathbb{E}[X_n].$$

Definition 3.10. A random process $\{X_n\}_{n\in\mathbb{N}}$ is ergodic *in correlation* if

$$\langle X_n X_{n-l}^* \rangle = \mathbb{E}[X_n X_{n-l}^*].$$

Remark. If $\{X_n\}_{n\in\mathbb{N}}$ is ergodic in both mean and correlation, then it is WSS.

Definition 3.11. Two random signals are called *jointly ergodic* if they are individually ergodic and in addition

$$\langle X_n Y_{n-l}^* \rangle = \mathbb{E}[X_n Y_{n-l}^*].$$

3.4 Martingales

The following defintions are not important for these notes, but they are useful in the theory of probability and have application nearly everywhere. I'll see if I can work in some applications of martingales somewhere in the text.

Definition 3.12. A *filtration* is a sequence $\{\mathcal{F}_n\}_{n\in\mathbb{N}_0}$ of sub-σ-algebras of $\mathcal{F}$ such that $\mathcal{F}_n \subseteq \mathcal{F}_{n+1}$, for all $n \in \mathbb{N}_0$. A probability space with a filtration—$(\Omega, \mathcal{F}, \{\mathcal{F}_n\}_{n\in\mathbb{N}_0}, \mathbb{P})$—is called a *filtered probability space*.

In a sense, a filtration captures the idea that as time moves forward we are only gaining new information (the σ-algebras get monotonically bigger as n increases). Anyone who has worked on a software project knows that this is absurd, but let's accept the defintion for now.

Definition 3.13. A stochastic process $\{X_n\}_{n\in\mathbb{N}_0}$ is said to be *adapted* with respect to a filtration $\{\mathcal{F}_n\}_{n\in\mathbb{N}_0}$ if X_n is $\mathcal{F}_n$-measurable for each $n \in \mathbb{N}_0$.

Definition 3.14. Let $\{\mathcal{F}_n\}_{n\in\mathbb{N}_0}$ be a filtration. A stochastic process $\{X_n\}_{n\in\mathbb{N}_0}$ is called an $\{\mathcal{F}_n\}_{n\in\mathbb{N}_0}$-*supermartingale* if

1. $\{X_n\}_{n\in\mathbb{N}_0}$ is $\{\mathcal{F}\}_{n\in\mathbb{N}_0}$-adapted,
2. $X_n \in \mathcal{L}^1$, for all $n \in \mathbb{N}_0$, and

3. $\mathbb{E}[X_{n+1}|\mathcal{F}_n] \leq X_n$, a.s.[1], for all $n \in \mathbb{N}_0$.

A process $\{X_n\}_{n\in\mathbb{N}_0}$ is called a *submartingale* if $\{-X_n\}_{n\in\mathbb{N}_0}$ is a supermartingale. A *(discrete-time) martingale* is a process which is both a supermartingale and a submartingale, i.e., the property 3 is an equality.

Basically, a martingale is a stochastic process such that tomorrow's expected value is equal to the value realized today. To show that a process is a martingale, it suffices to show property 3 (properties 1 and 2 are mostly just technicalities). A random walk (or Wiener process) is an example of a martingale.

4 Analysis of Stochastic Processes

Now that we have introduced stochastic processes, we can talk about how we actually analyze them as engineers and scientists.

4.1 Frequency-Domain Description of Stationary Processes

Definition 4.1. The *power spectral density* (PSD) of a stationary stochastic process $\{X_n\}_{n\in\mathbb{N}}$ is the Fourier transform of its autocorrelation sequence $r_X(l)$.

Definition 4.2. The *cross-power spectral density* of two zero-mean and jointly stationary stochastic processes provides a description of their statistical relations in the frequency domain and is defined as the discrete-time Fourier transformation (DTFT) of their cross-correlation, i.e.,

$$R_{XY}(e^{i\omega}) = \sum_{l=-\infty}^{\infty} r_{XY}(l)\, e^{i\omega l}.$$

Definition 4.3. The *(magnitude squared) coherence function* is defined as

$$C_{XY}(e^{i\omega}) = \frac{|R_{XY}(e^{i\omega})|^2}{R_X(e^{i\omega})\, R_Y(e^{i\omega})}.$$

This is sort of a correlation coefficient in the frequency domain.

4.2 LTI Systems

The notation I have been using for a stochastic process (e.g., $\{X_n\}_{n\in\mathbb{N}}$) is too cumbersome for the next sections and it expresses the wrong idea of what we work with in reality. We are working with a *realization* of the stochastic process $\{X_n\}_{n\in\mathbb{N}}$, which—from here on out—will be referred to as $x(n)$ (a random signal) where n is the index of a sample of a random variable in the stochastic process $\{X_n\}_{n\in\mathbb{N}}$ at time n. From here on out all random signals will be assumed to be stationary, unless otherwise stated.

[1] almost surely, i.e., equal (in this case, less than or equal) everywhere except, possibly, on set of measure 0

Definition 4.4. For an linear time-invariant system[1] (LTI), the output can be described as

$$y(n) = h(n) * x(n)$$

and this is not very surprising if you have seen any LTI system theory ever before. If the convolution exists for all events ζ that controls the random signal $x(n)$ such that $\mathbb{P}[\zeta] = 1$, then we say that we have almost-everywhere (a.e.) convergence (see the second Borel-Cantelli Lemma for more information).

Theorem 4.1. *If $x(n)$ is stationary, $\mathbb{E}[x(n)] < \infty$, and the system $h(n)$ is bounded-input bounded-output (BIBO) stable, then the output $y(n)$ converges a.e. and is stationary. Furthermore, if $\mathbb{E}[|x(n)|^2] < \infty$, then $\mathbb{E}[|y(n)|^2] < \infty$ and $y(n)$ converges in the mean square to the same limit and is setationary.*

Proposition 4.1 (Output mean value)**.** *If $x(n)$ is stationary, then its first moment is the mean value μ_x. The mean of the output is then*

$$\mu_y = \sum_{\mathbb{Z}} h(k)\mathbb{E}[x(n-k)] = \mu_x \sum_{\mathbb{Z}} h(k) = \mu_x H(e^{i0}).$$

Proposition 4.2 (Input-output cross-correlation)**.**

$$r_{xy}(l) = h^*(-l) * r_{xx}(l) \qquad r_{yx}(l) = h(l) * r_{xx}(l).$$

Proposition 4.3 (Output autocorrelation)**.**

$$r_{yy}(l) = h(l) * r_{xy}(l).$$

Proposition 4.4 (Output power)**.**

$$P_y = r_{yy}(0)$$

Remark (Output probability density function)*.* This is a hard problem except in some cases. If $x(n)$ is a Gaussian process (all samples are normally distributed), then the output is also a Gaussian process.

4.3 General Correlation Matrices

Here are some properties of correlation matrices.

Property 4.1. The correlation matrix of a random vector $\mathbf{x}$ is conjugate symmetric or Hermitian, i.e.,

$$\mathbf{R_x} = \mathbf{R_x}^H.$$

Property 4.2. The correlation of a random vector is positive semi-definite.

Property 4.3. The eigenvalues of $\mathbf{R}$ are real and nonnegative.

Property 4.4. Distinct eigenvalues of $\mathbf{R}$ correspond to orthogonal eigenvectors, i.e., if $\mathbf{x}, \mathbf{y}$ are eigenvectors, then $\mathbf{x}^T\mathbf{y} = 0$.

[1] a system whose transfer function (generally denoted by $h(t)$ in time) is both linear and does not change with respect to time

Property 4.5. Let $\{\mathbf{q}_i\}_{i=1}^{M}$ be an orthonormal set of eigenvectors corresponding to the distinct eigenvalues $\{\lambda\}_{i=1}^{M}$ of an $M \times M$ correlation matrix $\mathbf{R}$. Then $\mathbf{R}$ can be diagonalized as

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

where the orthonormal matrix $\mathbf{Q} := [\mathbf{q}_1 \cdots \mathbf{q}_M]$ is known as an *eigenmatrix* and Λ is an $M \times M$ diagonal eigenvalue matrix.

Property 4.6. The determinants of $\mathbf{R}$ and $\mathbf{\Gamma}$ are related by

$$|\mathbf{R}| = |\mathbf{\Gamma}|(1 + \boldsymbol{\mu}_x^H \mathbf{\Gamma}_{\mathbf{x}} \boldsymbol{\mu}_x).$$

4.4 Spectral Dynamic Range

Definition 4.5. The *condition number* $\chi(\cdot) = \lambda_{\max}/\lambda_{\min}$. A matrix is said to be *ill conditioned* when $\chi(\cdot)$ is large and *well conditioned* when $\chi(\cdot)$ is small. This is not well-defined, since it relates to how much the output value of the linear transformation can change for a small change in the input, i.e., sensitivity to error in the system.

Definition 4.6. The *dynamic range* is the ratio between the largest and smallest values that a particular quantity (e.g., for a radio signal the dynamic range would correspond to the maximum amplitude before clipping and the minimum before quantization errors).

When $\mathbf{R}_x$ is a correlation matrix of a stationary process, then $\chi(R_x)$ is bounded above by the dynamic range of the PSD $R_x(e^{i\omega})$ of the process $x(n)$. The larger the spread in eigenvalues, the wider (or less flat) the variation of the PSD function. This is related to the dynamic range or to the data spread in $x(n)$, and the result is given by the following theorem.

Theorem 4.2. *Consider a zero-mean stationary random process with PSD*

$$R(e^{i\omega}) = \sum_{l=-\infty}^{\infty} r(l) e^{-i\omega l}$$

then $\min_{\omega} R(e^{i\omega}) \le \lambda_j \le \max_{\omega} R(e^{i\omega})$ *for all* $j = 1, 2, \ldots, M$.

4.5 Innovations Representation

It is often desirable to represent a random vector with a linearly equivalent vector consisting of uncorrelated components. If $\mathbf{x}$ is a correlated random vector and if $\mathbf{A}$ is a nonsingular[1] matrix, then the linear transformation

$$\mathbf{w} = \mathbf{A}\mathbf{x} \tag{1}$$

results in a random vector $\mathbf{w}$ that contains the same "information" as $\mathbf{x}$, so $\mathbf{x}$ and $\mathbf{w}$ are said to be linearly equivalent. If $\mathbf{w}$ is an uncorrelated random vector, then each component "adds" new information (or *innovation*[2]) not present in other components. Such a representation is called an *innovations representation*[3] and provides additional insight into the understanding of random vectors and sequences. It also can simplify calculation and result in computationally efficient implementations.

[1] invertible

[2] an awful choice of wording

[3] emetic

4.5.1 Transformations Using Eigendecomposition

Let $\mathbf{x}$ be a random vector with mean vector $\boldsymbol{\mu}_{\mathbf{x}}$ and covariance matrix $\boldsymbol{\Gamma}_{\mathbf{x}}$. Let $\mathbf{x}_0 = \mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}$.

Orthonormal Transformation Let $\mathbf{Q}_{\mathbf{x}}$ be the eigenmatrix[1] of $\boldsymbol{\Gamma}_{\mathbf{x}}$. Let $\mathbf{A} = \mathbf{Q}_{\mathbf{x}}^H$ as in equation 1. Consider

$$\mathbf{w} = \mathbf{Q}_{\mathbf{x}}^H \mathbf{x}_0$$

then

$$\boldsymbol{\mu}_{\mathbf{w}} = \mathbf{Q}_{\mathbf{x}}^H \mathbb{E}[\mathbf{x}_0] = \mathbf{0}$$

and, it turns out,

$$\boldsymbol{\Gamma}_{\mathbf{w}} = \mathbf{R}_{\mathbf{w}} = \mathbb{E}[\mathbf{Q}_{\mathbf{x}}^H \mathbf{x}_0 \mathbf{x}_0^H \mathbf{Q}_{\mathbf{x}}] = \mathbf{Q}_{\mathbf{x}}^H \boldsymbol{\Gamma}_{\mathbf{x}} \mathbf{Q}_{\mathbf{x}} = \boldsymbol{\Lambda}_{\mathbf{x}}$$

where $\boldsymbol{\Lambda}_{\mathbf{x}}$ is the diagonal eigenvalue matrix of $\boldsymbol{\Gamma}_{\mathbf{x}}$.

Property 4.7.

1. The random vector $\mathbf{w}$ has zero mean, and its components are mutually uncorrelated (and hence orthogonal). Furthermore, if $\mathbf{x}$ is $N(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Gamma}_{\mathbf{x}})$, then $\mathbf{w}$ is $N(\mathbf{0}, \boldsymbol{\Lambda}_{\mathbf{x}})$.
2. The variances of w_i for $i = 1, \ldots, M$ are equal to the eigenvalues of $\boldsymbol{\Gamma}_{\mathbf{x}}$.
3. Since the transformation matrix $\mathbf{A} = \mathbf{Q}_{\mathbf{x}}^H$ is orthonormal, the transformation is called an *orthonormal transformation* and the distance measure

$$d^2(\mathbf{x}_0) := \mathbf{x}_0^H \boldsymbol{\Gamma}_{\mathbf{x}}^{-1} \mathbf{x}_0,$$

called the *Mahalanobis distance*, is preserved under the transformation. It is related to the log-likelihood function.

Isotropic (or Whitening) Transformation In the orthonormal transformation, the resulting autocorrelation matrix $\mathbf{R}_{\mathbf{w}}$ is diagonal but not an identity matrix. This can be achieved by an additional linear mapping $\boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2}$. Let

$$\mathbf{y} = \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} \mathbf{w} = \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} \mathbf{Q}_{\mathbf{x}}^H \mathbf{x}_0$$

and

$$\mathbf{R}_{\mathbf{y}} = \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} \mathbf{Q}_{\mathbf{x}}^H \boldsymbol{\Gamma}_{\mathbf{x}} \mathbf{Q}_{\mathbf{x}} \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} = \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} \boldsymbol{\Lambda}_{\mathbf{x}} \boldsymbol{\Lambda}_{\mathbf{x}}^{-1/2} = \mathbf{I}.$$

This is called an *isotropic*[2] *(or whitening) transformation* because all components of $\mathbf{y}$ are zero-mean, uncorrelated random variables with unit variance.

4.5.2 Transformations Using Triangular Decomposition

The linear transformations in the previous section were based on diagonalization of Hermitian matrices through eigenvalue-eigenvector decomposition (important in detection and estimation tasks). Triangular matrix decomposition leads to transformations that result in causal or anticausal linear filtering of associated sequences (so these are important in linear filtering tasks).

[1] the matrix of eigenvectors

[2] directionally invariant

Lower-diagonal-upper decomposition Any Hermitian, positive matrix $\mathbf{R}$ can be factored as

$$\mathbf{R} = \mathbf{L}\mathbf{D}_L\mathbf{L}$$

where $\mathbf{L}$ is a *unit lower triangular*[1] matrix, $\mathbf{D}_L$ is a diagonal matrix with positive elements, and $\mathbf{L}^H$ is a *unit upper triangular* matrix.

Since $\mathbf{L}$ is unit lower triangular, $|\mathbf{R}| = \prod_{i=1}^{M} \xi_i^l$ where ξ_i^l are the diagonal elements of $\mathbf{D}_L$. Define

$$\mathbf{w} = \mathbf{L}^{-1}\mathbf{x} := \mathbf{B}\mathbf{x}$$

then

$$\mathbf{R}_\mathbf{w} = \mathbb{E}[\mathbf{w}\mathbf{w}^H] = \mathbf{L}^{-1}\mathbb{E}[\mathbf{x}\mathbf{x}^H]\mathbf{L}^{-H} = \mathbf{L}^{-1}\mathbf{R}\mathbf{L}^{-H} = \mathbf{D}_L$$

which implies that the components of $\mathbf{w}$ are orthogonal, and the elements ξ_i^l are their second moments.

The matrix $\mathbf{B}$ can be interpreted as causual linear filtering. this transformation is used in optimal linear filtering and prediction problems.

A similar LDU decomposition of autocovariance matrices can be performed which will result in a $\mathbf{w}$ such that components are uncorrelated and the elements ξ_i^l of $\mathbf{D}_L$ are variances.

Note that the *upper-diagonal-lower* decomposition is nearly the same, but with the obvious differences, and it is used for anticausal filtering.

4.6 Estimation Theory

Up until now, we have assumed that the probability distributions associated with the problem under consideration were known. In most practical applications, this is not the case. Thus, we want to obtain the properties and parameters of random variables and processes to collecting and analyzing finite sets of measurements.

4.6.1 Properties of Estimators

Suppose that we collect N observations $\{x(n)\}_0^{N-1}$ from a stationary stochastic process and use them to estimate a parameter θ of the process using some function $\hat{\theta}[\{x(n)\}_0^{N-1}]$. The same results can be used on a set of measurements $\{x_k(n)\}_{k=1}^{M}$ obtained from M sensors sampling stochastic processes with the same distributions. The function $\hat{\theta}$ is known as an *estimator* and the value taken by the estimator, given a particular set of observations, is called a *(point) estimate*.

Here are two trivial examples of estimators:

1. The mean estimator: $\hat{\mu}_x = \frac{1}{N}\sum_{n=0}^{N-1} x(n)$.
2. The variance estimator: $\hat{\sigma}_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} [x(n) - \hat{\mu}_x]^2$.

[1] *unit lower/upper* means that the all diagonal elements equal 1

If we repeat this a large number of times (see law of large numbers, Theorem 2.1), we will obtain a large number of estimates which we can aggregate into a single random variable whose histogram approximates the distribution; this is called the *sampling distribution*.

The sampling distribution of a "good" estimator should be concentrated as closely as possible around the parameter that it estimates.

Definition 4.7 (Bias of estimator)**.** The *bias* of an estimator $\hat{\theta}$ of a parameter θ is defined as

$$B(\hat{\theta}) := \mathbb{E}[\hat{\theta}] - \theta$$

while the *normalized* bias is defined as

$$\epsilon_b = \frac{B(\hat{\theta})}{\theta}, \quad \theta \neq 0.$$

When $B(\hat{\theta}) = 0$, then the estimator is said to be *unbiased* and the pdf centered at the true value θ.

Definition 4.8 (Variance of estimator)**.** The *variance* of the estimator $\hat{\theta}$ is defined by

$$\mathrm{Var}(\hat{\theta}) = \sigma_{\hat{\theta}}^2 := \mathbb{E}[|\hat{\theta} - \mathbb{E}[\hat{\theta}]|^2]$$

which measures the spread of the pdf of $\hat{\theta}$ around its mean.[1]

Definition 4.9 (Normalized standard deviation)**.** The *normalized standard deviation* is defined by

$$\epsilon_r := \frac{\sigma_{\hat{\theta}}}{\theta}, \quad \theta \neq 0.$$

Definition 4.10 (Mean square error)**.** The *mean square error* (MSE) of the estimator is given by

$$\mathrm{MSE}(\theta) = \mathbb{E}[|\hat{\theta} - \theta|^2] = \sigma_{\hat{\theta}}^2 + |B|^2.$$

The *normalized MSE* is defined as

$$\epsilon = \frac{\mathrm{MSE}(\theta)}{\theta}, \quad \theta \neq 0.$$

Remark. Minimizing MSE *can* lead to an increase in bias.

Definition 4.11 (Cramér-Rao lower bound)**.** If it is possible to minimize the MSE and have bias equal to zero, then the variance is also minimized. Such estimators are called *minimum variance unbiased* estimators, and they attain an important minimum bound on the variance of the estimator called the *Cramér-Rao lower bound* (CRLB), or *minimum variance bound*. If $\hat{\theta}$ is unbiased, then it follows that $\mathbb{E}[\hat{\theta} - \theta] = 0$, which can be expressed as

$$\int \cdots \int (\hat{\theta} - \theta) f_{\mathbf{x};\theta}(\mathbf{x};\theta) d\mathbf{x} = 0$$

[1] In machine learning, when the bias of a *hypothesis* (equivalent to the estimator in estimation theory) is "high" it is generally a good idea to incorporate more features into the hypothesis or higher-order polynomial (or non-linear) terms. When the variance of the hypothesis is high, it is often a good idea to get more training data such that the hypothesis better fits the data. There is also the idea of adding a regularization term to the hypothesis which trades off bias and variance. High variance often means that the hypothesis is *overfit* to the data, and does not generalize well to unseen data. High bias often means that the hypothesis is *underfit* to the data, i.e., the hypothesis doesn't perform well even on the training set.

where $f_{\mathbf{x};\theta}(\mathbf{x};\theta)$ is the joint density of the random vector $\mathbf{x}$ which depends on the fixed but unknown parameter θ... (derivation)... presto bingo! the CRLB can be expressed as

$$\mathrm{Var}(\hat{\theta}) \geq -\frac{1}{\mathbb{E}\left[\frac{\partial^2 \ln f_{\mathbf{x};\theta}(\mathbf{x};\theta)}{d\theta^2}\right]}$$

Definition 4.12. The function $\ln f_{\mathbf{x};\theta}(\mathbf{x};\theta)$ is called the *log likelihood function* of θ. The CRLB expresses the minimum error variance of any estimator $\hat{\theta}$ of θ in terms of the joint density $f_{\mathbf{x};\theta}(\mathbf{x};\theta)$ of observations. So every unbiased estimator must have a variance greater than a certain number. An unbiased estimate that satisfies the CRLB with equality is called an *efficient* estimate. If there exists an efficient estimate, then it can be obtained as a unique solution to the likelihood equation

$$\frac{\partial \ln f_{\mathbf{x};\theta}(\mathbf{x};\theta)}{\partial\theta} = 0.$$

The solution of the above is called the *maximum likelihood* (ML) estimate.

Remark. If the efficient estimate does not exist, then the ML estimate will not achieve the lower bound and it is difficult to ascertain how closely the variance of any estimate will approach the bound.

Definition 4.13 (Consistency of estimator)**.** If the MSE of the estimator can be made to approach zero as sample size N becomes large, then both the bias and variance will tend to zero. The sampling distribution will tend to concentrate around θ and will converge to a Dirac measure $\delta_\theta(\cdot)$ as $N \to \infty$. An estimator with this property is called a *consistent* estimator.

Definition 4.14 (Confidence interval)**.** If we know the sampling distribution of an estimator, we can use the observations to compute an interval that has a specified probability of covering the unknown true parameter value. This interval is called a *confidence interval* and the coverage probability is called the *confidence level*.

Remark. The confidence interval is a random variable, but not the parameter.

Remark. The variance of the estimator increases as the amount of correlation among samples of $x(n)$ increases. For this reason, estimation of long-memory processes[1] are processes with infinite variance are very difficult.

5 Optimal Linear Filters

Optimal linear filters are filters that minimize the mean square error (MSE). The mimimum MSE (MMSE) criterion leads to a theory of linear filtering that only involves *known* second moment statistics of both the signal (desired system response) and the (additive) noise, i.e., the autocorrelation function and—more useful—the power spectral density (which is equivalent to the autocorrelation).

[1] the rate of decay of statistical dependence of two points with increasing time interval decays more slowly than an exponential decay. Look `here` for more information.

5.1 Optimal Signal Estimation

We will formulate and solve the following estimation problem: Given a set of data $x_k(n)$ for $1 \leq k \leq M$, determine an estimate $\hat{y}(n)$, of the desired response $y(n)$, using the rule (estimator)

$$\hat{y}(n) := H[x_k(n)], \qquad 1 \leq k \leq M$$

which, in general, is a nonlinear function of the data.

Definition 5.1. The difference between the estimated response $\hat{y}(n)$ and the desired response $y(n)$ is

$$e(n) := \hat{y}(n) - y(n)$$

and is known as the *error signal*.

We want to find an estimator whose output approximates the desired response as closely as possible according to a performance criterion. This is called an *optimal estimator* or *optimal signal processor*.

Remark. If the criterion of performance or the assumptions about the statistics of the processed signals change, the corresponding optimal filter will change as well. So an optimal estimator is designed for a specific performance criterion and assumption, and is only *optimal* under those conditions.

Thus, the design of an optimal estimator involves the following steps:

1. Selection of a computational structure with well-defined parameters for the implementation of the estimator.
2. Selection of a criterion of performance or cost function that measures the performance of the estimator under some assumptions about the statistical properties of the signals to be processed.
3. Optimization of the performance criterion to determine the parameters of the optimal estimator.
4. Evaluation of the optimal value of the performance criterion to determine whether the optimal estimator satisfies the design specifications.

In most applications, negative and positive errors are equally harmful. In these applications we choose a criterion that weights both equally. Some functions that satisfy the requirement are the absolute value of the error $|e(n)|$ or squared error $|e(n)|^2$ or some other power of $|e(n)|$.

Remark. Squared error emphasizes outliers heavier than absolute value (among other nice properties).

Note that we want to design an estimator that performs well across the entire ensemble of a random signal, i.e., performs well on average. Since at any time $y(n), x_k(n)$ for $k \in [1, M]$, and $e(n)$ are random variables, we need a criterion that involves the ensemble or time averaging of some function of $|e(n)|$. Here are some:

1. The mean square error criterion

   $$P(n) := \mathbb{E}[|e(n)|^2]$$

   which leads, in general, to a nonlinear optimal estimator.

2. The mean αth-order error criterion, i.e., $\mathbb{E}[|e(n)|^{\alpha}]$. Useful for certain types of non-Gaussian statistics (better than MSE).
3. The sum of squared errors (SSE)

$$E(n_i, n_f) := \sum_{n=n_i}^{n_f} |e(n)|^2$$

which, if it is divided by $n_f - n_i + 1$, provides an estimate of the MSE.

5.2 Linear Mean Square Error Estimation

This sections develops the theory of linear MSE estimation, concentrating on linear estimators. The problem can be stated as:

Design an estimator that provides an estimate $\hat{y}(n)$ of the desired response $y(n)$ using a linear combination of the data $x_k(n)$ for $k \in [1, M]$ such that the MSE $\mathbb{E}[|y(n) - \hat{y}(n)|^2]$ is minimized.

We formulate the estimation problem at a fixed time N, so we drop the index notation and we can restate the problem as follows:

Estimate a random variable y (desired response) from a set of related random variables $x_1, x_2, \ldots, x_M$ (data) using the linear estimator

$$\hat{y} = \mathbf{c}^H \mathbf{x} \tag{2}$$

where

$$\mathbf{x} = [x_1 \; x_2 \; \cdots \; x_M]^T$$

is the *input data vector* and

$$\mathbf{c} = [c_1 \; c_2 \; \cdots \; c_M]^T$$

is the *parameter* or *coefficient vector* of the estimator.

Unless stated otherwise, all random variables have zero-mean. The number M of data components used is called the *order* of the estimator. The operation in equation 2 is known as the *linear combiner*. The MSE

$$P := \mathbb{E}[|e|^2] \tag{3}$$

where $e := y - \hat{y}$ is a function of the parameters $c_k \in \mathbf{c}$. Minimization of equation 3 with respect to parameters leads to a linear estimator, denoted by $\mathbf{c}_o$, that is optimal in the MSE sense. The parameter vector $\mathbf{c}_o$ is known as the *linear MMSE (LMMSE) estimator* and $\hat{y}_o$ as the LMMSE estimate.

5.2.1 Error Performance Surface

The error surface for the LMMSE is given by

$$P(\mathbf{c}) = P_y - \mathbf{c}^H\mathbf{d} - \mathbf{d}^H\mathbf{c} + \mathbf{c}^H\mathbf{R}\mathbf{c}$$

where

$$P_y := \mathbb{E}[|y^2|]$$

is the power of the desired response,

$$\mathbf{d} := \mathbb{E}[\mathbf{x}y^*]$$

is the cross-correlation vector between the data vector $\mathbf{x}$ and the desired response y, and

$$\mathbf{R} := \mathbb{E}[\mathbf{x}\mathbf{x}^H]$$

is the correlation matrix of the data vector $\mathbf{x}$. Let's hope that $P(\mathbf{c})$ is convex!

Remark. The existence of the optimal estimator is guaranteed if the correlation matrix $\mathbf{R}$ is positive definite—this is almost always the case in real applications.

Also, the worst case scenario is that $\mathbf{x}$ and y are uncorrelated, i.e., $\mathbf{d} = 0$, because there is no linear estimator that can reduce the MSE.

Definition 5.2. Let $\tilde{\mathbf{c}}$ be the deviation from the optimal vector $\mathbf{c}_o$, i.e., $\mathbf{c} = \mathbf{c}_o + \tilde{\mathbf{c}}$. Then the *excess MSE* is defined as

$$\text{Excess MSE} := P(\mathbf{c}) - P(\mathbf{c}_o) = \tilde{\mathbf{c}}^H\mathbf{R}\tilde{\mathbf{c}}.$$

5.2.2 Summary

Theorem 5.1 (Orthogonality theorem). *Let $\hat{y} = \mathbf{a}^H\mathbf{x}$, where $\mathbf{a}$ is a vector of coefficients, $\mathbf{x}$ is a vector of observations, and $\hat{y}$ is the estimation. Let $e = y - \hat{y}$ be the error in estimation. Then $\mathbf{a}$ minimizes the mean square error $\sigma_e^2 = \mathbb{E}[|e|^2]$ if $\mathbf{a}$ is chosen such that $\mathbb{E}[x_i e^*] = \mathbb{E}[e x_i^*] = 0$, for $i = 1, 2, \ldots, N$, that is, if the error is orthogonal to the observations. Further, the minimum mean square error is given by $\sigma_e^2 = \mathbb{E}[ye^*] = \mathbb{E}[ey^*]$.*

1. The optimal estimator and the MMSE depend only on the second-order moments of the desired response and the data.
2. The error performance surface of the optimal estimator is a quadratic function of its coefficients. If the data correlation matrix is positive definite, then this function is convex.
3. If the data correlation matrix $\mathbf{R}$ is positive definite, then any deviation from the optimum increases the MMSE according to Definition 5.2. he resulting excess MSE depends on $\mathbf{R}$ only.
4. When the estimator operates with the optimal set of coefficients, the error e_o is uncorrelated (orthogonal) to both the data $\mathbf{x}$ and the optimal estimate $\hat{y}_o$.
5. The MMSE, the optimal estimator, and the optimal estimate can be expressed in terms of the eigenvalues and eigenvectors of the data correlation matrix.

6. The general estimator
$$\hat{y} := h(\mathbf{x})$$
that minimizes the MSE
$$P = \mathbb{E}[|y - h(\mathbf{x})|^2]$$
with respect to $h(\mathbf{x})$ is given by the mean conditional density, i.e.,
$$\hat{y}_o := h_o(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = \int_{-\infty}^{\infty} y f_y(y|\mathbf{x})\, dy$$
and is a nonlinear function of $\mathbf{x}$. If the desired response and the data are jointly Gaussian, the linear MMSE estimator is the *best* in the MMSE sense.

Linear MMSE estimation involves the following *computational* steps:

1. $\mathbf{R} = \mathbb{E}[\mathbf{x}\mathbf{x}^H], \mathbf{d} = \mathbb{E}[\mathbf{x}y^*]$	Normal equations $\mathbf{R}\mathbf{c}_o = \mathbf{d}$
2. $\mathbf{R} = \mathbf{L}\mathbf{D}\mathbf{L}^H$	Triangular decomposition
3. $\mathbf{L}\mathbf{D}\mathbf{k} = \mathbf{d}$	Forward substitution $\rightarrow \mathbf{k}$
4. $\mathbf{L}^H\mathbf{c}_o = \mathbf{k}$	Backward substitution $\rightarrow \mathbf{c}_o$
5. $P_o = \mathbf{k}^H\mathbf{D}\mathbf{k}$	MMSE computation
6. $e = y - \mathbf{c}_o^H\mathbf{x}$	Computation of residuals

For more details look in [1], Chapter 6.

6 Linear Prediction

Linear prediction deals with the problem of estimating the value $x(n)$ of a signal at a specific time $n = n_0$, as a linear combination of a set of disjoint values. In this section, we will only consider forward prediction, but note that backward "prediction" based on future values is also possible. Thus we are concerned with the case when we predict the current value $x(n)$ on previous values, i.e.,
$$\hat{x}(n) = \mathbf{a}^H\mathbf{x}(n) \quad \text{for} \quad \mathbf{a} = [-a_1 \cdots - a_P]^T,\ \mathbf{x}(n) := [x(n-1) \cdots x(n-P)]^T$$
where $\mathbf{a}$ are the linear prediction coefficients. We define the error in the estimate as
$$e(n) = x(n) - \hat{x}(n). \tag{4}$$
$\mathbf{a}$ and the prediction error variance $\sigma_e^2 = \mathbb{E}[|e(n)|^2]$ are the *linear prediction parameters*.

For convenience, let's prepend a 1 to $\mathbf{a}$ and $x(n)$ to $\mathbf{x}$ and call this new vector $\tilde{\mathbf{x}}$, then $e(n) = \mathbf{a}^H\mathbf{x}(n)$. To find the optimal filter coefficients apply the Orthogonality Theorem 5.1. Then
$$\mathbb{E}[\tilde{\mathbf{x}}(n)e^*(n)] = \sigma_e^2\boldsymbol{\imath}$$
where $\boldsymbol{\imath} \in \mathbb{C}^{P+1}$ and it is the unit vector pointing in the direction of the first coordinate.

Since $\tilde{\mathbf{R}}_\mathbf{x} = \mathbb{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H]$, it follows that
$$\tilde{\mathbf{R}}_\mathbf{x}\mathbf{a} = \sigma_e^2\boldsymbol{\imath}, \tag{5}$$
which represents the normal equations (the equations that solve for the optimal parameter values).

6.1 Autoregressive Model

Let's model the signal as the output of an all-pole (i.e., no zeros in the z-domain[1]) filter driven by white noise. Note that this makes the signal an innovations process. We can then model the process, $x(n)$ as a recursive system of the form

$$x(n) = \mathbf{c}^T\mathbf{x} + w(n)$$

which is equivalent to equation 4, but we are switching out $\mathbf{a}$ for $\mathbf{c}$ and e for w. Clearly, though, this is just linear regression where the "dependent" variable $x(n)$ is represented as a linear combination of the "independent" variables $x(n-1)$ to $x(n-P)$. Since they both belong to the same random process, $x(n)$ is called an *autoregressive* or AR process (regressed upon itself).

Remark. We use a slight alteration of the normal equations (Equation 5) to find the optimal values for $\mathbf{c}$, in this case, which is called the *Yule-Walker equations*. The only difference is that the conjugate transpose of the autocorrelation matrix is used.

The fact that AR model parameters satisfy Yule-Walker equations provides a practical method for signal modeling. Suppose we want to represent $x(n)$ by some AR model. The correlation function $R_x(l)$ can be estimated and used in the Yule-Walker equations to solve for the model parameters. This procedure is identical to that involved in solving for the filter coefficients in a linear prediction problem. Cool. Refer to Chapter 8 in [3] for more implementation details.

7 Least-Squares Error Estimation

In this section, we will deal with the design and properties of linear combiners, finite impulse response filters, and linear predictors that are optimal in the least-squares error (LSE) sense. This differs from optimal filters or previously discussed linear predictors, since for LSE, we do not require knowledge a priori of the second order moments. Thus, here, we use the minimization of the sum of the squares of the estimation error as the criterion of performance for the design of optimal filters. This method is known as *least-squares error (LSE) estimation*, which requires the measurement of both the input signal and the desired response signal.

Why estimate values of a known desired response signal? This is useful in several respects:

1. Want to obtain a mathematical model describing input-output behavior of an actual system.
2. Linear predictive coding, the prediction error or prediction coefficients are useful.
3. When the desired response is not available—but the data does not change significantly over a number of sets—then one complete training set can be used to design the estimator, which can be applied to the remaining incomplete sets.

In summary, if we only have a block of data, then we use the LSE estimator. If we know the second order moments, then we use the MMSE estimator. Note that if the sampled process is ergodic, then as the length of the data increases, the LSE estimator converges towards the MMSE estimator.

[1] The z-domain being the target space of the z-transform of the filter coefficients

Remark. This is just linear regression, e.g., a supervised learning technique in the domain of machine learning (and truly standard fare in any statistical domain). Essentially, we have a labeled set [1] of training data that we are using to make an estimator and the figure of merit for the estimator is that which minimizes the sum of square errors.

7.1 Problem

I'm just going to pose the problem, and then move on.

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \, (\boldsymbol{\theta}^T \mathbf{x} - \mathbf{y})^H (\boldsymbol{\theta}^T \mathbf{x} - \mathbf{y})$$

where $\hat{\boldsymbol{\theta}}$ are the estimated parameters, $\mathbf{x}$ is the input data vector, and $\mathbf{y}$ is the output (or desired response) vector. There are approximately $10^{10^{10}}$ implementations that solve this problem and there are way better explanations online compared to any of the texts used in this paper. Look them up if you need further details.

8 Parametric Spectrum Estimation

First let's talk about nonparametric spectrum estimation methods and preliminaries.

The Fast Fourier Transform (FFT)[2] outputs the frequency content, or spectrum, of a finite length sample of data. Windowing, implicit in taking a finite length sample of data, does also reduce the ability to resolve the exact frequency that we are trying to measure since frequencies are binned like a histogram. To counteract this, it is desirable to get more resolution in a data sample by zero-padding the to-be FFT'ed data. However, this contributes to spectral leakage[3]. To combat spectral leakage, we can use a different window shape (instead of the implied rectangular shape). Thus to reduce spectral leakage, we can use windows such as the Hann, Hamming, Kaiser, or Chebyshev window on the data. The choice depends on the applciation since each filter has different mainlobe width/sidelobe magnitude characteristics. Algorithm 1 details a method to implement zero-padded and windowed FFT.

Algorithm 1 Steps to FFT data with zero-padding and windowing

```
procedure MODIFIED FFT($\mathbf{x}$: data, $\mathbf{w}$: window choice)
    $\mathbf{x}' \leftarrow \mathbf{x} \odot \mathbf{w}$        ▷ $\odot$ is pointwise multiplication, $\mathbf{x}$ and $\mathbf{w}$ must be same length
    $l \leftarrow$ LENGTH($\mathbf{x}$)
    $n \leftarrow \lceil \log_2(l) \rceil$
    $p \leftarrow 2^{n+1}$
    $\mathbf{x}'' \leftarrow$ add zeros to end of $\mathbf{x}'$ such that $\mathbf{x}''$ is of length $p$.
    $\mathbf{f} \leftarrow$ FFT($\mathbf{x}''$)
    return $\mathbf{f}$
```

Algorithm 1 avoids problems with circular convolution and maximizes the efficiency of the FFT making the length a power of two.

[1] i.e., a pair $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is the input vector (feature vector) and $\mathbf{y}$ is the output vector (target vector)

[2] an efficient algorithm—$O(n \log n)$, for n length data when $\log_2(n) \in \mathbb{N}$—for the discrete Fourier Transform (DFT)

[3] peaks in the spectrum appear where there are none

However, this is a very rudimentary method to estimate the spectrum which is called *nonparametric*, since the method does not assume the estimator has a particular form and is data-independent.

Often, we want to look at the spectrum with relation to the power of each frequency, i.e., the power spectral density (mentioned in Definition 4.1). The PSD gives a physical representation of the spectrum versus the FFT which just supplies the amplitude of frequency bins relative to one another.

There are a myriad of other methods to estimate spectral density via nonparametric methods (i.e., with the FFT), such as with the periodogram. The periodogram is considered the worst; it's the magnitude of the FFT squared, which is justified by the Wiener-Khinchin theorem (see C.1)[1]. Other (nonparametric) methods include Bartlett's method[2] and Welch's method[3] among many others.

Parametric methods of spectrum estimation, on the other hand, assume that the available signal segment has the choice of an inappropriate signal model that will lead to erroneous results. Parametric methods allow us to resolve spectral peaks closer than the limit imposed by the amount of data available, as with the nonparametric methods[4]. However, we must have a priori information about the signal and noise to apply parametric methods; notably, the signal is assumed to be a wide-sense stationary process.

Remark. Note that the text from here on out within this section is almost directly, if not directly, copied from [1] (which seems to be almost directly taken from [3]). Since the only way to understand these estimation techniques is to understand their derivation, I just copied them over to expedite the writing process. I added notes where some clarity was needed (for me).

8.1 Minimum-Variance Spectrum Estimation

Note that a spectrum estimator's goal is to determine the power content of a signal at a certain frequency. Thus, we would like to measure the power spectral density $R(e^{i2\pi f})$ at the frequency of interest only and not have our estimate influenced by energy present at other frequencies. Thus we might interpret spectral estimation as a method to determine the ideal frequency-selective filter.

Definition 8.1. A *filter bank* is an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency subband of the original signal.

It turns out that we can derive the minimum-variance spectral estimator by using a filter bank structure in which each of the filters adapts its response to the data. Each filter h_k in the filter bank should pass energy within its bandwidth Δf but reject all other energy, i.e.,

$$|H_k(e^{i2\pi f})|^2 = \begin{cases} \Delta f & |f - f_k| \leq \frac{\Delta f}{2} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where H_k is the frequency response of the filter f_k. The factor $\Delta f \sim 1/M$ accounts for the filter bandwidth, where M is the window length of the data[5]. This term is a normalization term such

[1] since we are using finite length data, it turns out the right way to see the power spectrum in this manner is to use the *circular* autocorrelation of the data and then taking the FFT

[2] average of periodograms taken from multiple segments of the signal to reduce variance

[3] windowed version of Bartlett's that uses overlapping segments

[4] resolution of the DFT is proportional to the amount of input data

[5] DFT based methods resolve frequencies to approximately $\Delta f \sim 1/M$

that the filter does not impart a gain across the bandwidth of the filter. However, such an ideal filter does not exist in practice. Thus we need to design a filter that passes energy at the center frequency while rejecting as much out of band energy as possible.

A filter bank-based spectral estimator should have filters at all frequencies of interest. The filters should have equal spacing in frequency, spanning the fundamental frequency range $-\frac{1}{2} \leq f < \frac{1}{2}$. Let us denote the total number of frequencies by K and the center frequency of the k^{th} filter as

$$f_k = \frac{k-1}{K} - \frac{1}{2}$$

for $k = 1, 2, \ldots, K$. The output of the k^{th} filter is equal to

$$y_k(n) = h_k(n) * x(n) = \mathbf{c}_k^H x(n)$$

where

$$\mathbf{c}_k = [h_k^*(0)\, h_k^*(1)\ \cdots\ h_k^*(M-1)]^T$$

is the impulse response of the k^{th} filter and

$$\mathbf{x}(n) = [x(n)\ \cdots\ x(n-M+1)]$$

is the input data vector.

Let $\mathbf{v}(f)$ be the *frequency vector* which is a vector of complex exponentials at frequency f within the vector $\mathbf{x}(n)$.

$$\mathbf{v}(f) = [1\ e^{-i2\pi f}\ \cdots\ e^{i2\pi f(M-1)}]^T$$

Note that if $\mathbf{c}_k = \mathbf{v}$, then the filter bank just performs a DFT.

The output $y_k(n)$ of the k^{th} filter should ideally give an estimate of the power spectrum at f_k. The output power of the k^{th} filter is

$$\mathbb{E}[|y_k(n)|^2] = \mathbf{c}_k^H \mathbf{R}_x \mathbf{c}_k$$

where $\mathbf{R}_x$ is the correlation matrix of $\mathbf{x}$. Since the ideal filter response in equation 6 cannot be realized, we instead constrain our filter $\mathbf{c}_k$ to have a response at the center frequency f_k of

$$H_k(f_k) = |\mathbf{c}_k^H \mathbf{v}(f_k)|^2 = \frac{1}{M}. \tag{7}$$

This ensures that the center frequency of our bandpass filter is at the frequency f_k. To eliminate as much out-of-band energy as possible, the filter is formulated as the filter that minimizes its output power subject to the center frequency constraint in equation 7, i.e.,

$$\min\ \mathbf{c}_k^H \mathbf{R}_x \mathbf{c}_k \qquad \text{subject to} \qquad \mathbf{c}_k^H \mathbf{v}(f_k) = \frac{1}{\sqrt{M}}.$$

This constraint requires the filter to have a response of $1/\sqrt{M}$ to a frequency vector at the frequency of interest while minimizing energy from all other frequencies. The solution to this constrained optimization problem can be found via Lagrange multipliers, and it is

$$\mathbf{c}_k = \frac{\sqrt{M}\mathbf{R}_x^{-1}\mathbf{v}(f_k)}{\mathbf{v}^H(f_k)\mathbf{R}_x^{-1}\mathbf{v}(f_k)}.$$

The power of the signal, i.e., $\mathbb{E}[|y_k(n)|^2]$, is the minimum-variance spectral estimate

$$\hat{R}_M^{\mathsf{mv}}(e^{i2\pi f_k}) = \mathbb{E}[|y_k(n)|^2] = \frac{M}{\mathbf{v}^H(f_k)\mathbf{R}_x^{-1}\mathbf{v}(f_k)}.$$

Note that in order to compute the minimum-variance spectral estimate, we need to find the inverse of the correlation matrix, which is a Toeplitz matrix since $x(n)$ is stationary (e.g. can be calculated via Levinson recursion or other efficient algorithms).

8.2 Harmonic Models and Frequency Estimation

Often we can model a signal as an LTI system that is excited by white noise. However, often, signals of interest are complex exponentials embedded in white noise for which a *sinusoidal* or *harmonic model* is more appropriate. Signals consisting of complex exponentials are found in such settings as radar and array signal processing as moving targets and spatially propagating signals, respectively.

For complex exponentials found in noise, the parameters of interest are the frequencies of the signals. Thus, we want to estimate these frequencies from the data. We could estimate them with nonparametric methods, as described previous, and then the frequency estimates of the complex exponentials are the frequencies at which peaks occur in the spectrum. However, while this method works reasonably well, it does not take into account for the underlying model of complex exponentials in noise. Use of the appropriate model is desirable from both an intuitive point of view and in terms of performance. With the appropriate model we can resolve complex exponentials more closely spaced in frequency, the techniques associated with this are dubbed *superresolution* methods.

8.2.1 Harmonic Model

Consider the signal model that consists of P complex exponentials in noise, i.e.,

$$x(n) = \boldsymbol{\alpha}^T \boldsymbol{v}(n) + w(n)$$

where $\boldsymbol{v}$ is a vector of P unit-length complex exponentials at frequencies f_p for $p = 1, \ldots, P$; $\boldsymbol{\alpha}$ are some coefficients $\alpha_p \in \mathbb{C}$; and $w(n)$ is white noise. The normalized, discrete-time frequency of the p^{th} component is

$$f_p = \frac{\omega_p}{2\pi} = \frac{F_p}{F_s}$$

where ω_p is the discrete-time frequency in radians, F_p is the actual frequency of the p^{th} complex exponential, and F_s is the sampling frequency. The complex exponentials may occur either individually or in complex conjugate pairs (i.e., (z, z^*) for $z \in \mathbb{C}$), as in the case of real signals[1]. In general, we want to estimate the frequencies and potentially the amplitudes of these signals. Note that the phase of each complex exponential is contained in the coefficient of the complex exponential, that is,

$$\alpha_p = |\alpha_p| e^{i\psi_p}$$

[1] $e^{i\omega} = \sin(\omega) + i\cos(\omega); (e^{i\omega})^* = e^{-i\omega} = \cos(\omega) - i\sin(\omega); e^{i\omega} + (e^{i\omega})^* = 2\cos(\omega) \in \mathbb{R}$

where the phases ψ_p are uncorrelated random variables uniformly distributed over $[0, 2\pi]$. The magnitude $|\alpha_p|$ and frequency f_p are deterministic quantities.

If we consider the spectrum of a harmonic process[1], we note that it consists of a set of impules with a constant level at the power of the white noise $\sigma_w^2 = \mathbb{E}[|w(n)|^2]$. As a result, the power spectrum of complex exponentials is commonly refered to as a *line spectrum*[2].

Now let's characterize the signal model in the form of a vector over time, i.e.,

$$\mathbf{x}(n) = [x(n)\ x(n+1)\ \cdots\ x(n+M-1)]^T = (\boldsymbol{\alpha}^T \boldsymbol{\nu}(n))\mathbf{v}(f_p) + \mathbf{w}(n) = \mathbf{s}(n) + \mathbf{w}(n) \tag{8}$$

where $\mathbf{w}(n) = [w(n)\ w(n+1)\ \cdots\ w(n+M-1)]^T$ is the time-window vector of white noise and

$$\mathbf{v}(f) = [1\ e^{i2\pi f}\ \cdots\ e^{i2\pi(M-1)}f]^T$$

is the time-window frequency vector[3]. $\mathbf{s}(n)$ is the signal, and $\mathbf{w}(n)$ is the noise.

The autocorrelation matrix of this model can be written as the sum of signal and noise autocorrelation matrices, i.e.,

$$\begin{aligned}\mathbf{R}_x &= \mathbb{E}[\mathbf{x}(n)\mathbf{x}^H(n)] = \mathbf{R}_s + \mathbf{R}_w \\ &= \mathbf{V}\mathbf{A}\mathbf{V}^H + \sigma_w^2\mathbf{I}\end{aligned}$$

where

$$\mathbf{V} = [\mathbf{v}(f_1)\ v(f_2)\ \cdots\ \mathbf{v}(f_p)$$

is an $M \times P$ matrix whose columns are the time-window frequency vectors at frequencies f_p of the complex exponentials and

$$\mathbf{A} = \begin{bmatrix} |\alpha_1|^2 & 0 & \cdots & 0 \\ 0 & |\alpha_2|^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & |\alpha_P|^2 \end{bmatrix}$$

is a diagonal matrix of the powers of each of the respective complex exponentials.

The autocorrelation matrix of the white noise is

$$\mathbf{R}_w = \sigma_w^2\mathbf{I}$$

which is full rank, as opposed to $\mathbf{R}_s$, which is rank-deficient for $P < M$. Thus it is good practice to use $M > P$.

The autocorrelation matrix can also be written in terms of its eigendecomposition

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

[1] A *harmonic process* is defined by $x(n) = \sum_{k=1}^{M} A_k \cos(\omega_k n + \phi_k)$ and $\omega_k \neq 0$, where M, $\{A_k\}_1^M$, and $\{\omega_k\}_1^M$ are constants and $\{\phi_k\}_1^M$ are pairwise independent random variables uniformly distributed in the interval $[0, 2\pi]$. $x(n)$ is a stationary process with mean $\mathbb{E}[x(n)] = 0$ for all n and autocorellation $r_x(l) = \frac{1}{2}\sum_{k=1}^{N} A_k^2 \cos(\omega_k l)$ for $-\infty < l < \infty$.

[2] Doesn't really make sense, but sure. I suppose "line with spikes spectrum" didn't have the same ring.

[3] this is terrible notation to time-shift the results so that it lines up with equation 8. Unclear and terrible, but I'm following [1] here so blame them.

where $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues in descending order, and $\mathbf{Q}$ is made up of the corresponding eigenvectors. The eigenvalues due to the signals can be written as the sum of the signal power in the time window and the noise:

$$\lambda_m = M|\alpha_m|^2 + \sigma_w^2 \quad \text{for} \quad m \leq P. \tag{9}$$

The remaining eigenvalues are due to the noise only, i.e.,

$$\lambda_m = \sigma_w^2 \quad \text{for} \quad m > P.$$

Thus the P largest eigenvalues correspond to the signal made up of complex exponentials and the remaining eigenvalues have equal value and correspond to the noise. Thus we can partition the correlation matrix into portions due to the signal and noise eigenvectors, i.e.,

$$\mathbf{R}_x = \mathbf{Q}_s\mathbf{\Lambda}_s\mathbf{Q}_s^H + \sigma_w^2\mathbf{Q}_w\mathbf{Q}_w^H \tag{10}$$

where

$$\mathbf{Q}_s = [\mathbf{q}_1 \; \cdots \; \mathbf{q}_P] \qquad \mathbf{Q}_w = [\mathbf{q}_{P+1} \; \cdots \; \mathbf{q}_M]$$

are matrices whose columns consist of the signal and noise eigenvectors, respectively. The matrix $\mathbf{\Lambda}_s$ is a $P \times P$ diagonal matrix containing the eigenvalues in equation 9. Thus the M-dimensional subspace that contains the observations of the time-window signal vector from equation 8 can be split into two subspaces spanned by the signal and noise eigenvectors, respectively. These two subspaces, known as the *signal subspace* and the *noise subspace*, are orthogonal to each other since the correlation matrix is Hermitian symmetric[1].

Note that the projection matrix from an M-dimensional space onto and L-dimensional subspace, where $L < M$, spanned by a set of vectors $\mathbf{Z} = [\mathbf{z}_1 \; \mathbf{z}_2 \; \cdots \; \mathbf{z}_L]$ is

$$\mathbf{P} = \mathbf{Z}(\mathbf{Z}^H\mathbf{Z})^{-1}\mathbf{Z}^H.$$

Thus, we can write the matrices that project an arbitrary vector onto the signal and noise subspaces as

$$\mathbf{P}_s = \mathbf{Q}_s\mathbf{Q}_s^H \qquad \mathbf{P}_w = \mathbf{Q}_w\mathbf{Q}_w^H$$

since the eigenvectors of the correlation matrix are orthonormal ($\mathbf{Q}_{s,w}^H\mathbf{Q}_{s,w} = \mathbf{I}$). Since the two subspaces are orthogonal

$$\mathbf{P}_w\mathbf{Q}_s = \mathbf{0} \qquad \mathbf{P}_s\mathbf{Q}_w = \mathbf{0}$$

then all the time-window frequency vectors $\mathbf{v}(f)$ must lie completely in the signal subspace, i.e.,

$$\mathbf{P}_s\mathbf{v}(f_p) = \mathbf{v}(f_p) \qquad \mathbf{P}_w\mathbf{v}(f_p) = \mathbf{0}.$$

These concepts are central to the following subspace-based frequency estimation methods.

Note that the correlation matrix is not known and must be estimated from the measured data samples. If we have a time-window vector as in equation 8, then we can form the data matrix by stacking the rows with measurements of the time-window data vector at a time n, i.e.,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(0) \\ \mathbf{x}^T(1) \\ \vdots \\ \mathbf{x}^T(n) \\ \vdots \\ \mathbf{x}^T(N-2) \\ \mathbf{x}^T(N-1) \end{bmatrix} = \begin{bmatrix} x(0) & x(1) & \cdots & x(M-1) \\ x(1) & x(2) & \cdots & x(M) \\ \vdots & \vdots & \vdots & \vdots \\ x(n) & x(n+1) & \cdots & x(n+M-1) \\ \vdots & \vdots & \vdots & \vdots \\ x(N-2) & x(N-1) & \cdots & x(N+M-3) \\ x(N-1) & x(N) & \cdots & x(N+M-2) \end{bmatrix}$$

[1] the eigenvectors of a Hermitian symmetric matrix are orthogonal

which has dimensions of $N \times M$, where N is the number of data records and M is the time-window length. From this we can form an estimate of the correlation matrix, called the sample correlation matrix

$$\hat{\mathbf{R}}_x = \frac{1}{N}\mathbf{X}^H\mathbf{X}. \tag{11}$$

In this case, the noise eigenvalues are no longer equal because of the finite number of samples used to compute $\hat{\mathbf{R}}$. Thus, there is not a *clean* threshold between signal and noise eigenvalues, as described in equation 9, so P must be estimated via some other technique. Note that there is some performance degradation compared to the true correlation matrix; however, this is the best we have to work with in reality.

8.3 Pisarenko Harmonic Decomposition

The *Pisarenko harmonic decomposition* (PHD) is stated here since it motivates the MUSIC and ESPRIT subspace methods, however it is too sensitive to noise for practical use. The PHD method uses the eigenvector associated witht he smallest eigenvalue to estimate the frequencies of the complex exponentials.

Consider the model of complex exponentials contained in noise proposed in equation 8 and the eigendecomposition of its correlation matrix in equation 10. The eigenvector corresponding to the minimum eigenvalue must be orthogonal to all the eigenvectors in the signal space. Thus we choose the time window to be of length $M = P + 1$, i.e., 1 greater than the number of complex exponentials. Therefore the noise subspace consists of a single eigenvector

$$\mathbf{Q}_w = \mathbf{q}_M$$

corresponding to the minimum eigenvalue λ_M. Since the signal and noise subspaces are orthogonal, each of the P complex exponentials in the time-window signal vector model is orthogonal to this eigenvector, i.e.,

$$\mathbf{v}^H(f_p)\mathbf{q}_M = 0 \quad \text{for} \quad p \leq P.$$

Thus we can compute

$$\bar{R}_{\text{phd}}(e^{i2\pi f}) = \frac{1}{|\mathbf{v}^H(f)\mathbf{q}_M|^2} = \frac{1}{|Q_M(e^{i2\pi f})|^2}$$

which is commonly referred to as the *pseudospectrum* (since it does not contain information about the powers of the complex exponentials or the background level noise).

The frequencies are then estimated by observing the P peaks in $\bar{R}_{\text{phd}}$. Alternately, the frequencies of the complex exponentials can be found by computing the zeros of the Fourier transform of the M^{th} eigenvector.

8.4 MUSIC Algorithm

The *multiple signal classification* (MUSIC) frequency estimation method extends the PHD method by allowing $M > P + 1$. Therefore the noise subspace has a dimension greater than 1 which allows for averaging over the noise subspace, which provides a more robust frequency estimation method compared to the PHD method.

Note that for each eigenvector ($P < m \leq M$), we have

$$\mathbf{v}^H(f_p)\mathbf{q}_m = 0 \quad \text{for} \quad p \leq P.$$

Thus the pseudospectrum for each noise eigenvector is computed as

$$\bar{R}_m(e^{i2\pi f}) = \frac{1}{|\mathbf{v}^H(f)\mathbf{q}_m|^2} = \frac{1}{|Q_m(e^{i2\pi f})|^2}.$$

The polynomial $Q_m(e^{i2\pi f})$ has $M-1$ roots, P of which correspond to the frequencies of the complex exponentials. These roots produce P peaks in the pseudospectrum. Note that the pseudospectra of all $M-P$ noise eigenvectors occur at different frequencies. Since there are no constraints on the location of these roots, some may be close to the unit circle and produce extra peaks in the pseudospectrum. A means of reducing the levels of these spurious peaks in the pseudospectrum is to average the $M-P$ pseudospectra of the individual noise eigenvectors, i.e.,

$$\bar{R}_{\text{music}}(e^{i2\pi f}) = \frac{1}{\sum_{m=P+1}^{M} |Q_m(e^{i2\pi f})|^2} \tag{12}$$

which is known as the MUSIC pseudospectrum.

The peaks in the MUSIC pseudospectrum correspond to the frequencies at which the denominator in equation 12 approaches zero.

8.5 ESPRIT Algorithm

The *estimation of signal parameters via rotational invariance techniques* (ESPRIT) algorithm estimates the signal subspace from the data matrix $\mathbf{X}$ instead of the estimated correlation matrix $\hat{\mathbf{R}}_x$. The core principle of ESPRIT lies in the rotational property between staggered (in time) subspaces that is invoked to produce the frequency estimates.

Remark. ESPRIT can be extended to a spatial array of sensors, i.e., array processing. More on that in the next section.

Consider a single complex exponential $s_0(n) = \alpha e^{i2\pi f n}$ with complex amplitude α and frequency f. This signal has the property

$$s_0(n+1) = \alpha e^{i2\pi f(n+1)} = s_0(n)e^{i2\pi f}$$

that is, the next sample is a phase-shifted version of the current value, which is just a rotation on the unit-circle in the complex plane.

Recall the time-window vector model from equation 8, we can rewrite that as so

$$\mathbf{x}(n) = \mathbf{V}\mathbf{\Phi}^n\boldsymbol{\alpha} + \mathbf{w}(n)$$

where the P columns of matrix $\mathbf{V}$ are length-M time-window frequency vectors of complex exponentials, i.e.,

$$\mathbf{V} = [\mathbf{v}(f_1)\ \mathbf{v}(f_2)\ \cdots\ \mathbf{v}(f_P)].$$

The vector $\boldsymbol{\alpha}$ consists of the amplitudes of the complex exponentials α_p. The matrix $\mathbf{\Phi}$ is the diagonal matrix of phase-shifts between neighboring time samples of the individual complex exponential components of $\mathbf{s}(n)$, i.e.,

$$\mathbf{\Phi} = \text{diag}(\phi_1, \phi_2, \ldots, \phi_P)$$

where $\phi_p = e^{i2\pi f_p}$ for $p = 1, 2, \ldots, P$. Note that this is a rotation matrix (since each component causes a rotation) and the frequencies of the complex exponentials f_p completely describe the rotations. Thus the frequency estimates can be obtained by finding $\mathbf{\Phi}$.

Consider two overlapping subwindows of length $M - 1$ within the length M time-window vector, and consider the signal consisting of the sum of complex exponentials

$$\mathbf{s}(n) = \begin{bmatrix} \mathbf{s}_{M-1}(n) \\ s(n+M-1) \end{bmatrix} = \begin{bmatrix} s(n) \\ \mathbf{s}_{M-1}(n+1) \end{bmatrix}$$

where $\mathbf{s}_{M-1}(n)$ is the length-$(M-1)$ subwindow of $\mathbf{s}(n)$, i.e.,

$$\mathbf{s}_{M-1}(n) = \mathbf{V}_{M-1}\mathbf{\Phi}^n\boldsymbol{\alpha} \tag{13}$$

Matrix $\mathbf{V}_{M-1}$ is constructed in the same manner as $\mathbf{V}$ except its time-window frequency vectors are length $M - 1$. Recall that $s(n)$ is the scalar signal made up of the sume of complex exponentials at time n. Using equation 13, we can define the matrices

$$\mathbf{V}_1 = \mathbf{V}_{M-1}\mathbf{\Phi}^n\mathbf{V}_2 = \mathbf{V}_{M-1}\mathbf{\Phi}^{n+1}$$

where $\mathbf{V}_1$ and $\mathbf{V}_2$ correspond to the unstaggered and staggered windows. Then

$$\mathbf{V}_2 = \mathbf{V}_1\mathbf{\Phi}. \tag{14}$$

Note that each of these two matrices spans a different, though related, $(M-1)$-dimensional subspace.

Now suppose that we have a data matrix $\mathbf{X}$ with N data records of the length-M time-window vector signal $\mathbf{x}(n)$. We can use SVD (see A.6 for more details) to write the data matrix as

$$\mathbf{X} = \mathbf{L}\mathbf{\Sigma}\mathbf{U}^H$$

where $\mathbf{L}$ is an $N \times N$ matrix of left singular vectors and $\mathbf{U}$ is an $M \times M$ matrix of right singular vectors. Both of these matrices are unitary. Matrix $\mathbf{\Sigma}$ has dimensions $N \times M$ consisting of singular values on the main diagonal ordered in descending magnitude.

The squared magnitudes of the signlar values are equal to the eigenvalues of $\hat{\mathbf{R}}$ scaled by a factor of N from equation 11, and the columns of $\mathbf{U}$ are their corresponding eigenvectors. Thus $\mathbf{U}$ forms an orthonormal basis for the underlying M-dimensional vector space. This subspace can be partitioned into signal and noise subspaces as

$$\mathbf{U} = [\mathbf{U}_s \mid \mathbf{U}_n]$$

where $\mathbf{U}_s$ is the matrix of right-hand singular vectors corresponding to the singular values with the P largest magnitudes. Note that since the signal portion consists of the sum of complex exponentials modeled as time-window frequency vectors $\mathbf{v}(f)$, all these frequency vectors for $f = f_1, f_2, \ldots, f_P$, must also lie in the signal subspace.

Therefore, there exists an invertible transformation $\mathbf{T}$ that maps $\mathbf{U}_s$ into $\mathbf{V}$, i.e.,

$$\mathbf{V} = \mathbf{U}_s\mathbf{T}.$$

The transformation $\mathbf{T}$ is not solved for in this derivation; it is just a map between the two linear transformations within the signal subspace.

Proceeding as we did with the matrix $\mathbf{V}$, we can also partition the signal subspace into two smaller $(M-1)$-dimensional subspaces; call these $\mathbf{U}_1$ and $\mathbf{U}_2$, which correspond to the unstaggered and staggered subspaces, respectively. Since $\mathbf{V}_1$ and $\mathbf{V}_2$ correspond to the same subspaces, then

$$\mathbf{V}_1 = \mathbf{U}_1\mathbf{T} \qquad \mathbf{V}_2 = \mathbf{U}_2\mathbf{T}. \tag{15}$$

The staggered and unstaggered components of the matrix $\mathbf{V}$ are related through the subspace rotation $\mathbf{\Phi}$. Since the matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ also span these respective, related subspaces, a similar—though different—rotation must exist that relates (rotates) $\mathbf{U}_1$ to $\mathbf{U}_2$, i.e.,

$$\mathbf{U}_2 = \mathbf{U}_1\mathbf{\Psi} \tag{16}$$

where $\mathbf{\Psi}$ is this rotation matrix.

Recall that frequency estimation comes down to solving for the subspace rotation matrix $\mathbf{\Phi}$. We can estimate $\mathbf{\Phi}$ by making use of the relations in equations 15, 14, and 16.

In this process, matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ are known from the SVD of $\mathbf{X}$. First we solve for $\mathbf{\Psi}$ using least-squares (LS)

$$\mathbf{\Psi} = (\mathbf{U}_1^H\mathbf{U}_1)^{-1}\mathbf{U}_1^H\mathbf{U}_2,$$

and it follows from equation 16 that

$$\mathbf{V}_2 = \mathbf{U}_2\mathbf{T} = \mathbf{U}_1\mathbf{\Psi}\mathbf{T}.$$

Similarly, $\mathbf{V}_2$ can be solved from equation 14 as

$$\mathbf{V}_2 = \mathbf{V}_1\mathbf{\Phi} = \mathbf{U}_1\mathbf{T}\mathbf{\Phi}.$$

Thus, we get the following relation between the two subspace rotations

$$\mathbf{\Psi} = \mathbf{T}\mathbf{\Phi}\mathbf{T}^{-1}.$$

Notice that this is a relationship between eigenvectors and eigenvalues of the matrix $\mathbf{\Psi}$. Therefore, the diagonal elements of $\mathbf{\Phi}$, ϕ_p for $p = 1, 2, \ldots, P$, are simply the eigenvalues of $\mathbf{\Psi}$.

Therefore, the estimates of the frequencies are

$$\hat{f}_p = \frac{\measuredangle \phi_p}{2\pi}.$$

Remark. The previous derivation of ESPRIT used LS; however, the preferred version uses total least-squares. The details of the TLS version are not much different, so they are not included.

9 Array Processing Fundamentals

Array processing concerns the extration of information from signals collected with an array of sensors. An array of sensors can focus on signals from a particular direction, i.e., serve as a spatial filter. To filter in space, the sensor array signals are combined in such a way that a particular direction is emphasized. Note that since the direction in which the array is focused is almost independent of the orientation of the array, we can emphasize multiple different directions.

9.1 Sensor Arrays

Definition 9.1. *Spatial signals* are signals that propagate through space. Since space is three-dimensional, a spatial signal at a point specified by the vector $\mathbf{r}$ can be represented either in Cartesian coordinates (x, y, z) or, more usually, in spherical coordinates $(R, \phi_{\text{az}}, \theta_{\text{el}})$, where $R = \|\mathbf{r}\|$, and ϕ_{az} and θ_{el} are the azimuth and elevation angles, respectively.

The propagation of a spatial signal is governed by the solution to the wave equation; for electromagnetic propagating signals, the wave equation can be deduced from Maxwell's equations. In any case, the solution for a propagating wave emanating from a source located at $\mathbf{r}_0$, is a single-frequency wave given by

$$s(t, \mathbf{r}) = \frac{A}{\|\mathbf{r} - \mathbf{r}_0\|^2} e^{i2\pi F_c\left(t - \frac{\|\mathbf{r}-\mathbf{r}_0\|}{c}\right)}$$

where A is the complex amplitude, F_c is the carrier frequency, and c is the speed of propagation of the wave. In this paper, we ignore the singularity at $\mathbf{r}_0$. Further, we assume that the wave has the frequency F_c at any point in space, the wave travels at constant speed, and that the medium does not attenuate the propagating signal further than predicted by the wave equation. Additionally, we assume the signal is produced from a point source, and that we are in the far-field (i.e., working with plane waves).

Recall the wavelength of a spatial signal is defined as

$$\lambda = \frac{c}{F_c}.$$

Consider placing a linear array—uniformly spaced—in three-dimensional space in order to sense spropagating waves, such an array is known as a *uniform linear array* (ULA). In this paper, we choose the ULA to be placed on the x axis in three-dimensional space. Thus the source of a plane wave that impinges on the ULA can be described by its distance from the origin $\|\mathbf{r}\|$ and its azimuth and elevation angles ϕ_{az} and θ_{el}, respectively. If the distance between elements of the ULA is d, then the difference between neighboring elements for a plane wave arriving from an azimuth ϕ_{az} and elevation θ_{el} is

$$d_x = \|\mathbf{r}\| \sin\phi_{\text{az}} \cos\theta_{\text{el}}.$$

These differences in the propagation distance that the plane wave must travel to each of the sensors are a function of a general *angle of arrival* (AoA) with respect to the ULA ϕ. In three-dimensional space, the delays are an equivalence class with respect to a cone surrounding the ULA, i.e., a signal arriving at the ULA on a cone surface has the same set of relative delays between the elements. Thus, the angle of incidence to a linear array is commonly referred to as a cone angle, ϕ_{cone}.

Note that the cone angle is a function of azimuth and elevation defined by

$$\sin\phi = \sin\phi_{\text{az}} \cos\theta_{\text{el}}$$

where $\phi = 90^\circ - \phi_{\text{cone}}$. Thus we calculate the cone angle given an azimuth and elevation pair.

9.1.1 Array Signal Model

First let's define the signal model. Let $\tilde{s}_0(t) = s_0(t)\cos 2\pi F_c t$ be the modulated signal. Let ϕ_s be the angle of the signal received by the ULA. Then the demodulated signal is

$$\tilde{x}_m(t) = \tilde{s}_0(t - \tau_m) * h_m(t, \phi_s) + \tilde{w}_m(t),$$

where $h_m(t, \phi_s)$ is the impulse response of the m^{th} sensor, $\tilde{w}_m(t)$ is the noise on the m^{th} sensor, and τ_m is the signal delay on the m^{th} sensor. Note that τ_m is a function of ϕ_s. This demodulated signal is then split into two digital signals via an ADC, the final signal is denoted

$$x_m(n) = x_m^{(I)}(n) + i x_m^{(Q)}(n)$$

where $x_m^{(I)}(n)$ and $x_m^{(Q)}(n)$ are the in-phase and quadrature components of the demodulated signal, respectively.

The discrete-time signals from a ULA may be written as a vector containing the individual sensor signals, i.e.,

$$\mathbf{x}(n) = [x_1(n)\ x_2(n)\ \cdots\ x_M(n)]^T$$

where M is the total number of sensors.

Assumptions of this model is that the signal $s_0(t)$ has a deterministic amplitude and random, uniformly distributed phase. The ˜ is used to indicate that the signal is a passband or carrier-modulated signal.

We can also express the signal in the frequency domain as

$$\tilde{X}_m(f) = H_m(f, \phi_s)\tilde{S}_0(f)e^{-i2\pi f \tau_m} + \tilde{W}_m(f).$$

After using a low-pass filter, we get the spectrum of the signal as

$$X_m(f) = H_m(f + F_c, \phi_s)S_0(f)e^{i2\pi(f+F_c)\tau_m} + W_m(f) \tag{17}$$

and $X_m(f) = X_m^{(I)} + iX_m^{(Q)}$.

From here we assume that the bandwidth of $s_0(t)$ is small compared to the carrier frequency, which is known as the *narrowband assumption*. This assumption allows us to approximate the propagation delays of a particular signal between sensor elements with a phase shift, since the phase difference between the upper and lower band edges for propagation across the entire array is small.

There is no strict definition of what consitutes a narrowband assumption, but, in general, it holds for cases in which the signal bandwidth is less than some small percentage of the carrier freqeuncy, e.g., less than 1 percent. The ratio of the signal bandwidth to the carrier frequency is referred to as the *fractional bandwidth*. Note that the fractional bandwidth for which the narrowband assumption holds is strongly dependent on the length of the array and the strength of the received signals. Thus it is convenient to consider the *time-bandwidth product* (TBWP), which is the maximum amount of time for a spatial signal to propagate across the entire array ($\phi_s = \pm 90°$). If TBWP $\ll 1$, then the narrow bandassumption is valid.

In addition to the narrowband assumption, we assume that the response of the sensor is constant across the receiver, i.e., $H_m(f + F_c, \phi_s) = H_m(F_c, \phi_s)$ for $|f| < B/2$. Thus, the spectrum in equation 17 is simplified to

$$X_m(f) = H_m(F_c, \phi_s) S_0(f) e^{-i2\pi F_c \tau_m} + W_m(f)$$

and the discrete-time signal model is obtained by sampling the inverse Fourier transform of the above, i.e.,

$$x_m(n) = H_m(F_c, \phi_s) s_0(n) e^{-i2\pi F_c \tau_m} + w_m(n).$$

Note that we assume $w_m(n)$ has a flat PSD across the bandwidth of the receiver, i.e., the discrete-time noise samples are uncorrelated. Also, the noise in all sensors is mutually uncorrelated. If we further assume that each of the sensors in the array has an equal, omnidirectional response at frequency F_c, i.e., $H_m(F_c, \phi_s) = H(F_c, \phi_s) =$ constant, for $1 \leq m \leq M$, then the constant sensor responses can be absorbed into the signal term

$$s(n) = H(F_c) s_0(n).$$

We can then, and will for the remainder of the chapter, write the full-array discrete-time signal model as

$$\mathbf{x}(n) = \sqrt{M}\, \mathbf{v}(\phi_s) s(n) + \mathbf{w}(n) \tag{18}$$

where

$$\mathbf{v}(\phi) = \frac{1}{\sqrt{M}} [1 \; e^{i2\pi F_c \tau_2(\phi)} \; \cdots \; e^{-i2\pi F_c \tau_M(\phi)}]^T$$

is the *array response vector*. We have chosen to measure all delays relative to the first sensor ($\tau_1(\phi) = 0$) and are now indicating the dependence of these delays on ϕ. $\frac{1}{M}$ is used such that $\mathbf{v}$ has unit norm. Another critical assumption at this point is to note that we have perfect knowledge of the array sensor locations.

Note that the array signal model we have proposed holds for arbitrary arrays. However, we will focus on a ULA structure with uniform spacing d between sensor elements. Then for a signal that impinges on the ULA from an angle ϕ, we have the time delay between successive sensors equal to

$$\tau(\phi) = \frac{d \, \sin(\phi)}{c}$$

where c is the rate of propagation of the signal. As a result, the delay to the m^{th} sensor with respect to the first is defined as

$$\tau_m(\phi) = (m-1) \frac{d \, \sin(\phi)}{c}$$

Then we see the array response vector for a ULA is

$$\mathbf{v}(\phi) = \frac{1}{\sqrt{M}} [1 \; e^{-i2\pi[(d \, \sin\phi)/\lambda]} \; \cdots \; e^{-i2\pi[(d \, \sin\phi)/\lambda](M-1)}]^T \tag{19}$$

since $F_c = c/\lambda$.

9.1.2 Spatial Sampling

We can interpret a sensor array as a mechanism to spatially sample wavefronts propagating at a certain carrier frequency. Note that sampling frequency must be high enough so as not to create spatial ambiguities.

In an arbitrary, sampling is done in multiple dimensions and along a nonuniform grid so that it is difficult to compare to discrete-time sampling. However, a ULA has a direct correspondence to uniform, regular temporal sampling, since it samples uniformly in space on one axis. Thus, for a ULA, we can talk about *spatial sampling frequency* U_s defined by

$$U_s = \frac{1}{d}$$

where the spatial sampling period is determined by the interelement spacing d and is measured in cycles per unit of length (meters).

Recall that consecutive samples of the same signal differ only by a phase shift of $e^{i2\pi f}$, where f is the frequency. In the case of a spatially propagating signal, this frequency is given by

$$U = \frac{\sin\phi}{\lambda}$$

which can be thought of as the *spatial frequency*. The *normalized spatial frequency* is then defined by

$$u := \frac{U}{U_s} = \frac{d\ \sin\phi}{\lambda}.$$

Therefore, we can rewrite the array response vector from equation 19 in terms of the normalized spatial frequency,

$$\mathbf{v}(\phi) = \mathbf{v}(u) = \frac{1}{\sqrt{M}}[1\ e^{-i2\pi u}\ \cdots\ e^{-i2\pi u(M-1)}]^T$$

which is a *Vandermonde* vector, i.e., a vector whose elements are successive integer powers of the same number, in this case $e^{-i2\pi u}$.

The interelement spacing d is simply the spatial sampling interval, which is the inverse of the sampling frequency. Therefore, similar to Shannon's (or Nyquist) theorem for discrete-time sampling, there are certain requirements on the spatial sampling frequency to avoid aliasing. Since normalized frequencies are unambiguous for $-\frac{1}{2} \leq u < \frac{1}{2}$ and the full range of possible unambiguous angles is $-90^\circ \leq \phi \leq 90^\circ$, the sensor spacing must be

$$d \leq \frac{\lambda}{2}$$

to prevent spatial ambiguities. Since lowering the array spacing further than this only provides redundant information, we generally set $d = \lambda/2$.

9.2 Beamforming

We often want to extract the information of a spatially propagating signal from a sensor array in a certain direction. Thus, we want to linearly combine the signals from all the sensors so as to examine signals arriving from a specific angle. This is known as *beamforming*, since the weighting process emphasizes signals from a particular direction while attenuating those from other directions. Thus a beamformer is a spatial filter, and in the case of a ULA, it has a direct analogy to an FIR frequency-selective filter for temporal signals.

Beamforming is commonly referred to as "electronic" steering since the weights are applied using electronic circuitry following the reception of the signal for the purpose of steering the array in a particular direction (in contrast to mechanical steering).

In the most general form, a beamformer produces its output by forming a weighted combination of signals from the M elements of the sensor array, i.e.,

$$y(n) = \mathbf{c}^H \mathbf{x}(n)$$

where

$$\mathbf{c} = [c_1 \; c_2 \; \cdots \; c_M]^T$$

is the column vector of beamforming weights.

9.2.1 Beam Response

To analyze the performance of a beamformer, we can look at the response of a given weight vector $\mathbf{c}$ as a function of angle ϕ, known as the *beamresponse*, i.e.,

$$C(\phi) = \mathbf{c}^H \mathbf{v}(\phi)$$

for $\phi \in [-90°, 90°)$. We look at $|C(\phi)|^2$ for analysis, which is known as the *beampattern*.

This differs from the *steered response*, which is the response of the array to a certain set of spatial signals impinging on the array as we steer the array to all possible angles, this is better defined as the spatial power spectrum, i.e.,

$$R(\phi) = \mathbb{E}[|\mathbf{c}^H \mathbf{v}(\phi) \mathbf{x}(n)|^2].$$

9.2.2 Output Signal-to-Noise Ratio

Now we determine the improvement in SNR with respect to each element, known as the *beamforming gain*. Let us use the signal model in equation 18, then the beamformer $\mathbf{c}$ is applied to $\mathbf{x}(n)$ as

$$y(n) = \mathbf{c}^H \mathbf{x}(n) = \sqrt{M}\, \mathbf{c}^H \mathbf{v}(\phi_s) s(n) + \bar{w}(n)$$

where $\bar{w} = \mathbf{c}^H \mathbf{w}(n)$ is the noise at the beamformer output and is also temporally uncorrelated.

The beamformer output power is

$$P_y = \mathbb{E}[|y(n)|^2] = \mathbf{c}^H \mathbf{R}_x \mathbf{c}$$

where R_x is the correlation matrix for $\mathbf{x}$.

Recall that the signal for the m^{th} element is given by

$$x_m(n) = e^{-i2\pi(m-1)u_s} s(n) + w_m(n)$$

where u_s is the normalized spatial frequency of the array signal produced by $s(n)$. The signal $s(n)$ is the signal of interest within a single sensor including the sensor response $H_m(F_c)$. Therefore the signal-to-noise ratio in each element is given by

$$\text{SNR}_{\text{elem}} := \frac{\sigma_s^2}{\sigma_w^2} = \frac{|e^{-i2\pi(m-1)u_s}s(n)|^2}{\mathbb{E}[|w_m(n)|^2]}$$

where $\sigma_s^2 = \mathbb{E}[|s(n)|^2]$ and $\sigma_w^2 = \mathbb{E}[|w_m(n)|^2]$ are the element level signal and noise powers, respectively. Recall that $s(n)$ has deterministic amplitude and random phase, and we assume that all the elements have equal noise power σ_w^2 so that the SNR does not vary from element to element. This SNR_{elem} is commonly referred to as the *element level SNR*.

Now if we consider the signals at the output of the beamformer, the signal and noise powers are given by

$$P_s = \mathbb{E}[|\sqrt{M}\,[\mathbf{c}^H\mathbf{v}(\phi_s)]s(n)|^2] = M\sigma_s^2|\mathbf{c}^H\mathbf{v}(\phi_s)|^2$$
$$P_n = \mathbb{E}[|\mathbf{c}^H\mathbf{w}(n)|^2] = \mathbf{c}^H\mathbf{R}_n\mathbf{c} = \|\mathbf{c}\|^2\,\sigma_w^2$$

because $\mathbf{R}_n = \sigma_w^2\mathbf{I}$. Therefore the resulting SNR at the beamformer output known as the *array SNR*, is

$$\text{SNR}_{\text{array}} = \frac{P_s}{P_n} = \frac{M|\mathbf{c}^H\mathbf{v}(\phi_s)|^2}{\|\mathbf{c}\|^2}\frac{\sigma_s^2}{\sigma_w^2} = \frac{|\mathbf{c}^H\mathbf{v}(\phi_s)|^2}{\|\mathbf{c}\|^2}M\;\text{SNR}_{\text{elem}}$$

which is simply the product of the beamforming gain and the element level SNR. Thus, the *beamforming gain* is given by

$$G_{bf} := \frac{\text{SNR}_{\text{array}}}{\text{SNR}_{\text{elem}}} = \frac{|\mathbf{c}^H\mathbf{v}(\phi_s)|^2}{\|\mathbf{c}\|^2}M.$$

The beamforming gain is strictly a function of the angle of arrival ϕ_s of the desired signal, the beamforming weight vector $\mathbf{c}$, and the number of sensors M.

9.2.3 Spatial Matched Filter

The beamforming weight vector that phase-aligns a signal from direction ϕ_s at the different array elements is the *steering vector*, which is just the array response vector in that direction, i.e.,

$$\mathbf{c}_{mf}(\phi_s) = \mathbf{v}(\phi_s).$$

The steering vector beamformer is also known as the *spatial matched filter* since the steering vector is matched to the array response of signals impinging on the array from an angle ϕ_s, which is known as the *look direction*. This is commonly referred to as *conventional beamforming*.

In this case, the beamforming gain turns out to be equal to the number of sensors, and is also called the *array gain*. The spatial matched filter maximizes the SNR because the individual sensor signals are coherently aligned prior to their combination. However, other sources of interference that have spatial correlation require other types of adaptive beamformers that maximize the signal-to-interference-plus-noise ratio (SINR).

The beampattern of the spatial matched filter has a large lobe centered on ϕ_s, known as the *mainbeam*, and the remaining smaller peaks are known as sidelobes.

Another attribute is the *beamwidth*, which is the angular span of the mainbeam. The smaller the beamwidth, the greater the angular resolution. The beamwidth is commonly measured from half-power (-3-dB) points $\Delta\phi_{\text{3dB}}$ or from null to null of the mainlobe $\Delta\phi_{nn}$.

9.2.4 Spacing and Aperture

Previously, we stated that element spacing must be $d \leq \lambda/2$ to prevent spatial aliasing. However, there are ways to use an array with element spacing $d > \lambda/2$ which is commonly referred to as a *thinned array*. I won't discuss that more here.

The *aperture* is the distance between the first and last element in a ULA; in an arbitrary array, the aperture can be defined as the sensors furthest apart (it is not well defined since it can change based on the direction of the impinging signal). In general, we want the largest aperature possible, since resolution increases with the size of the aperature. Thus we can see closely spaced sources and have better angle estimation capabilities.

The angular resultion of a sensor array is measured in beamwidth $\Delta\phi$. In general, the -3-dB beamwidth for an array with an aperture length of L is quoted in radians as

$$\Delta\phi_{\text{3dB}} \approx \frac{\lambda}{L}.$$

For funsies, the actual -3-dB points of a spatial matched filter yield a resolution of $\Delta\phi_{\text{3dB}} = 0.89\lambda/L$.

9.2.5 Tapered Beamforming

The spatial matched filter would be perfect if there were only one signal present, but that is most often not the case. If these signals are in the operating frequency of the array and are not of interest, then we call them *interference*. There are ways to deal with interference with both adaptive and nonadaptive methods. Let's first talk about nonadaptive methods, which in this case will be with the use of a taper on a spatial matched filter.

Briefly, a taper is just a window (i.e., Hann, Hamming, Chebyshev, etc.) applied to the beamforming coefficients which reduces the sidelobes at the expense of resolution and peak power. See Algorithm 1 for an idea of what is going on.

9.3 Optimal Array Processing

Now let's base the beamforming weights on the array data instead of a priori knowledge of direction. This leads to an *adaptive array* and the operation is known as *adaptive beamforming*. Ideally, the beamforming weights are adapted in such a way as to optimize the spatial response of the array to a certain criterion, i.e., enhance desired signal while rejecting unwanted signals.

To create such a system, we make use of a priori known statistics of the data to derive the beamforming weights. In this section, we will use the term *adaptive* to refer to beamformers that use an estimated correlation matrix computed from array snapshots, while reserving the term *optimal* for beamformers that optimize a certain criterion based on knowledge of the array data statistics.

Here are some more definitions. They are pretty straight-forward, but let's make sure we are all on the same page.

Definition 9.2. *Detection* is the determination of the presense of signals of interest.

Definition 9.3. The inference of parameters from signals of interest is called *estimation*.

Note that we seek to maximize the *visibility* of the desired signal at the array output, i.e., the ratio of the signal power to that of the interference plus noise to facilitate the detection process (in this paper at least).

We assume that interference has spatial correlation according to the angles of the contributing interferers. Further, in this paper, we assume that the signal of interest, the interference, and the noise are all mutually uncorrelated.

9.3.1 Optimal Beamforming

The goal of the adaptive beamformer is to combine the sensor signals in such a way that the interference signal is reduced to the level of the thermal noise while the desired signal is preserved, i.e., maximize the SINR.

The SINR at each sensor is given by

$$\text{SINR}_{\text{elem}} = \frac{\sigma_s^2}{\sigma_i^2 + \sigma_w^2}$$

where σ_s^2, σ_i^2, and σ_w^2 are the signal, interference, and thermal noise powers in each individual element. The SINR at the beamformer output, following the application of the beamforming weight vector $\mathbf{c}$ is

$$\text{SINR}_{\text{out}} = \frac{|\mathbf{c}^H \mathbf{s}(n)|^2}{\mathbb{E}[|\mathbf{c}^H \mathbf{x}_{i+n}(n)|^2} = \frac{M\sigma_s^2 |\mathbf{c}^H \mathbf{v}(\phi_s)|^2}{\mathbf{c}^H \mathbf{R}_{i+n} \mathbf{c}}.$$

We wish to maximize this quantity.

It turns out that the maximum SINR is

$$\text{SINR}_{\text{out}}^{\text{max}} = M\sigma_s^2 [\mathbf{v}^H(\phi_s) \mathbf{R}_{i+n}^{-1} \mathbf{v}(\phi_s)]$$

and it follows from this that the optimal coefficients are

$$\mathbf{c}_o = \alpha \mathbf{R}_{i+n}^{-1} \mathbf{v}(\phi_s)$$

for arbitrary constant α. However, we want unity gain in the look direction in this paper, and the resulting beamformer is given by

$$\mathbf{c}_o = \frac{\mathbf{R}_{i+n}^{-1} \mathbf{v}(\phi_s)}{\mathbf{v}^H(\phi_s) \mathbf{R}_{i+n}^{-1} \mathbf{v}(\phi_s)}.$$

We can also find the optimal beamformer with the following optimization formulation

$$\mathbf{c}_o = \arg\min_{\mathbf{c}} \mathbb{E}[|\mathbf{c}^H \mathbf{x}_{i+n}|^2] = \arg\min_{\mathbf{c}} \mathbf{c}^H \mathbf{R}_{i+n} \mathbf{c} \quad \text{subject to} \quad \mathbf{c}^H \mathbf{v}(\phi_s) = 1.$$

This formulation has led the optimal beamformer to be called the *minimum-variance distortionless response (MVDR) beamformer*.

We will want to gauge the performance of the optimal beamformer relative to the interference-free case. To do this we normalize the SINR by the hypothetical array output SNR with no interference $(\mathsf{SNR}_0 = M\sigma_s^2/\sigma_w^2)$, which is known as the *SINR loss*,

$$L_{\mathsf{sinr}}(\phi_s) := \frac{\mathsf{SINR}_{\mathsf{out}}(\phi_s)}{\mathsf{SNR}_0} = \sigma_w^2 \mathbf{v}^H(\phi_s)\mathbf{R}_{i+n}^{-1}\mathbf{v}(\phi_s).$$

Note that the SINR loss is always between 0 and 1, and takes on the value 1 when the performance is equal to the interference-free case. Also notice that L_{sinr} is the reciprocal of the minimum-variance power spectrum of the interference plus noise.

Recall, we can apply a taper (window) to the coefficients to lower sidelobe levels in the optimal beamformer (at the expense of “optimality”).

Note that the optimal beamformer serves as the upper bound on the performance of any adaptive method. Two major factors that affect the performance of the optimal beamformer are

1. Mismatch of the actual signal to the assumed signal model used by the optimal beamformer, called *signal mismatch*
2. Violation of the narrowband assumption on the signal.

We won’t discuss these problems here, but know that they affect performance and there are analytical methods to determine the performance degradation. See Chapter 11, Section 4 in [1] for more.

Before moving on, I will provide some definitions to improve your beamforming vocabulary bigly:

Definition 9.4.

1. The array response vector $\mathbf{v}(\phi)$ is also referred to as the *array manifold vector*. Note that the manifold vector for a particular direction contains all the information about the geometry involved when a wave is incident on the array from that direction. By recording the locus of the manifold vectors as a function of direction, a “continuum” (i.e., a geometrical object such as a curve or surface) is formed lying in an N-dimensional space. The geometrical object is known as the *array manifold*. The array manifold completely characterizes any array and provides a representation of the real array into N-dimensional complex space. [6]
2. The *distortionless constraint* is the constraint that requires $\mathbf{c}^H\mathbf{v}(\phi) = 1$ which guarantees that any signal propagating along the direction of the signal will pass through the filter undistorted.
3. The *quiescent response* of a beamformer is defined as

$$C_q(\phi) = \mathbf{v}^H(\phi_s)\mathbf{v}(\phi).$$

4. The *eigenbeam* is the beam response of the m^{th} eigenvector defined as

$$Q_m(\phi) = \mathbf{q}_m^H\mathbf{v}(\phi).$$

9.4 Adaptive Beamforming

Note that the optimal beamformer can only be achieved because we assumed prior knowledge of the second order moments of the interference at the array, i.e., the interference-plus-noise correlation matrix $\mathbf{R}_{i+n}$. However, we would like to use adaptive methods that are only based on collected data from which the correlation matrix is estimated.

9.4.1 Sample Matrix Inversion

In practice, we must estimate the correlation matrix from the data. The maximum-likelihood (ML) estimate of the correlation matrix is given by the average of outer products of the array snapshots

$$\hat{\mathbf{R}}_{i+n} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}_{i+n}(n_k)\mathbf{x}_{i+n}^H(n_k),$$

where the indices n_k define the K samples of $\mathbf{x}_{i+n}(n)$ for $1 \leq n \leq N$ that make up the *training set*.

The ML estimate of the correlation matrix implies that as $K \to \infty$, then $\hat{\mathbf{R}}_{i+n} \to \mathbf{R}_{i+n}$; $\hat{\mathbf{R}}_{i_n}$ is known as the *sample correlation matrix*. The total number of snapshots K used to compute the sample correlation matrix is referred to as the *sample support*. The larger the sample support, the better the estimate $\hat{\mathbf{R}}_{i+n}$ of the correlation matrix for stationary data.

Substituting the sample correlation matrix into the optimum beamformer weight computation results in the adaptive beamformer

$$\mathbf{c}_{\text{smi}} = \frac{\hat{\mathbf{R}}_{i+n}^{-1}\mathbf{v}(\phi_s)}{\mathbf{v}^H(\phi_s)\hat{\mathbf{R}}_{i+n}^{-1}\mathbf{v}(\phi_s)}$$

known as the *sample matrix inversion* (SMI) adaptive beamformer.

We can apply tapers and analyze the beamformer as before. Just recall that the more snapshots that train the sample correlation matrix, the better the response of the beamformer.

To implement the SMI beamformer, we must estimate the interference-plus-noise correlation matrix, which requires no desired signals $\mathbf{s}(n)$ be present. In many applications, we can turn off the desired signal and train in *listen-only* mode. We can also use a *split window* technique, that uses data only from before and after a signal is present.

To increase robustness of the SMI beamformer when there is limited training data, we can use *diagonal loading* on the sample correlation matrix, i.e.,

$$\hat{\mathbf{R}}_l = \hat{\mathbf{R}}_{i+n} + \sigma_l^2\mathbf{I}.$$

Diagonal loading adds bias (reduces variance) which reduces output SINR, but gives an increased quality of the adaptive beampattern in limited training data scenarios.

9.4.2 Other Adaptive Methods

We can compute the beamforming weights on a sample-by-sample basis; referred to as *sample-by-sample adaptive*. They solve an unconstrained least-squares (LS) problem, whereas what we have discussed so far has solved a constrained[1] LS problem. This is solved via recursive least squares or gradient descent. Note that since these algorithms take time to converge, sample-by-sample methods are not always practical.

We can also design a beamformer to reject energy from multiple directions while passing energy from multiple directions. A way to solve this problem is to formulate the problem as an optimization problem; called the *linearly constrained minimum-variance* (LCMV) beamformer

$$\mathbf{c}_{\text{lcmv}} = \arg\min_{\mathbf{c}} \mathbf{c}^H \mathbf{R}_{i+n} \mathbf{c} \quad \text{subject to} \quad \mathbf{C}^H \mathbf{c} = \delta$$

where $\mathbf{C}$ is the constraint matrix and δ is the constraint response vector. We can do a similar operation with quadratic constraints as well (with a different formluation).

Other topics that are of interest are partially adaptive arrays, subarray partially adaptive arrays, and beamspace partially adaptive arrays. These methods reduce the computational complexity of processing the data collected by an array by reducing the *degrees of freedom* of the array (number of places we can put a null or, equivalently, place a beam).

9.5 Angle Estimation

Now we will discuss determining an angle of arrival ϕ_s for a signal $\mathbf{s}(n)$.

9.5.1 Maximum-Likelihood Angle Estimation

Consider a spatially propagating signal of interest

$$\mathbf{s} = \sqrt{M}\,\sigma_s \mathbf{v}(\phi_s)$$

where M is the number of sensors in the ULA, σ_s is the complex amplitude of the signal, and ϕ_s is the angle of arrival of the signal relative to the sensor array. The signal received by the ULA has both interference $\mathbf{i}$ and spatially uncorrelated thermal noise $\mathbf{w}$, i.e.,

$$\mathbf{x} = \mathbf{s} + \mathbf{i} + \mathbf{w} = \mathbf{s} + \mathbf{x}_{i+n}.$$

We are not using the discrete-time index n since we are assuming the signal is present and we are interested in a single snapshot only. The interference-plus-noise correlation matrix of $\mathbf{x}$ is given by

$$\mathbf{R}_{i+n} = \mathbb{E}[\mathbf{x}_{i+n}\mathbf{x}_{i+n}^H] = \mathbf{R}_i + \sigma_w^2 \mathbf{I}.$$

We also assume that the interference-plus-noise signal $\mathbf{x}_{i+n}$ has a complex Gaussian density function with zero mean. Thus, the probability density function of the snapshot is a complex Gaussian function with a mean determined by the signal of interest, i.e.,

$$f(\mathbf{x}; \sigma_s, \phi_s) = \frac{1}{\pi^M \det(\mathbf{R}_{i+n})} \exp\left(-[\mathbf{x} - \mathbf{s}]^H \mathbf{R}_{i+n}^{-1} [\mathbf{x} - \mathbf{s}]\right).$$

[1] distortionless constraint

The maximum value in f corresponds to the mean given by the signal of interest $\mathbf{s}$, which is the "most likely" event. Thus, the maximum-likelihood (ML) angle estimate is given by

$$\hat{\phi}_s = \arg\max_{\phi} f(\mathbf{x}; \sigma_s, \phi_s).$$

This makes the ML estimator of ϕ_s equal to

$$\hat{\phi}_s = \arg\max_{\phi} \frac{|\mathbf{v}^H(\phi)\mathbf{R}_{i+n}^{-1}\mathbf{x}|^2}{\mathbf{v}^H(\phi)\mathbf{R}_{i+n}^{-1}\mathbf{v}(\phi)}.$$

9.5.2 Cramér-Rao Lower Bound on Angle Accuracy

The *Cramér-Rao lower bound* (CRLB) places a lower bound on the performance of an unbiased estimator (where lower is better). We provide a sketch of the derivation of the CRLB for angle accuracy. The CRLB provides the minimum variance of an unbiased estimator. If an estimator can achieve the CRLB, then it is the maximum-likelihood estimator.

We first redefine the beamformer for a ULA from the spatial matched filter that has its phase center moved from the first element to the center of the array

$$\begin{aligned}\mathbf{v}_\Sigma(\phi) &= e^{-i2\pi\frac{M-1}{2}\frac{d}{\lambda}\sin\phi}\mathbf{v}(\phi) \\ &= \frac{1}{\sqrt{M}}\begin{bmatrix} e^{-i2\pi\frac{M-1}{2}\frac{d}{\lambda}\sin\phi} & e^{-i2\pi\frac{M-3}{2}\frac{d}{\lambda}\sin\phi} & \dots & e^{i2\pi\frac{M-1}{2}\frac{d}{\lambda}\sin\phi}\end{bmatrix}^T\end{aligned}$$

which we will refer to as the *sum beamformer*. This choice of a phase center provides the tightest bound on accuracy. We can define a second beamformer based on the derivative of $\mathbf{v}_\Sigma(\phi)$ given by

$$\mathbf{v}_\Delta(\phi) = i\delta \odot \mathbf{v}_\Sigma(\phi)$$

where

$$\delta = \begin{bmatrix} -\frac{M-1}{2} & -\frac{M-3}{2} & \dots & \frac{M-1}{2}\end{bmatrix}^T$$

which can be thought of as a difference taper. The steering vector $\mathbf{v}_\Delta(\phi)$ provides a difference pattern beamformer steered to the angle ϕ, and we correspondingly call this the *difference beamformer*. In relation to the sum beamformer, we get

$$\mathbf{v}_\Delta^T(\phi)\mathbf{v}_\Sigma(\phi) = 0$$

or, in other words, they are orthogonal. Since the two beamformers are orthogonal to each other, then—in terms of the signal $\mathbf{s}$—the two beamformers can make two independent measurements of the signal. These independent measurements allow for the discrimination of angle.

Using these two steering vectors $\mathbf{v}_\Delta(\phi)$ and $\mathbf{v}_\Sigma(\phi)$, we can form an adaptive sum beamformer

$$\mathbf{c}_\Sigma(\phi) = \mathbf{R}_{i+n}^{-1}\mathbf{v}_\Sigma(\phi)$$

and an adaptive difference beamformer

$$\mathbf{c}_\Delta(\phi) = \mathbf{R}_{i+n}^{-1}\mathbf{v}_\Delta(\phi)$$

which both have not been normalized to satisfy any particular criteria.

Proceeding, we can compute the power of the interference-plus-noise output of these two beamformers

$$P_\Sigma = \mathbf{c}_\Sigma^H \mathbf{R}_{i+n} \mathbf{c}_\Sigma \qquad P_\Delta = \mathbf{c}_\Delta^H \mathbf{R}_{i+n} \mathbf{c}_\Delta.$$

Similarly, we can measure the normalized cross-correlation $\rho^2_{\Sigma\Delta}$ of the interference-plus-noise outputs of these adaptive sum and difference beamformers $\mathbf{R}_{i+n}$

$$\rho^2_{\Sigma\Delta} = \frac{|\mathbf{c}_\Sigma^H \mathbf{R}_{i+n} \mathbf{c}_\Delta|}{P_\Sigma P_\Delta}.$$

Then, using the power of the interference-plus-noise output of the two beamformers and the normalized cross-correlation, we get the CRB on angle estimation for a ULA:

$$\sigma^2_\phi \geq \frac{1}{2\pi^2 \cdot \text{SNR}_0 \cdot P_\Delta (1 - \rho^2_{\Sigma\Delta}) \cos^2 \phi}$$

where SNR_0 is the SNR for a spatial matched filter in the absence of interference, i.e., only noise, which is given by

$$\text{SNR}_0 = M \frac{\sigma_s^2}{\sigma_w^2}$$

where M is the number of elements in the array and the $\sigma^2_{s,w}$ are the second order moments of the signal and noise, respectively.

First, notice that as the signal power increases, SNR_0 increases; as a result, angle accuracy improves. Likewise, the term $\cos^2 \phi$ represents the increase in beamwidth of the ULA as we steer away from the broadside $(\phi = 0°)$. P_Δ provides a measure of the received power aligned with the adaptive difference beamformer. Ideally, $\rho_{\Sigma\Delta}$ is zero, since $\mathbf{c}_\Sigma$ and $\mathbf{c}_\Delta$ beamformers are derived from $\mathbf{v}_\Sigma$ and $\mathbf{v}_\Delta$, respectively, which are orthogonal to each other. In the case of the two adaptive beamformers, the adaptiation will remove this orthogonality, but the beamformers should be different enough that $\rho_{\Sigma\Delta} \ll 1$. Otherwise, angle accuracy will suffer.

Okay, so the takeaway of this section is that SNR improves angle estimation, and angle estimation accuracy improves along the side with the most antennas.

9.5.3 Beamsplitting Algorithms

Let's consider the scenario with a single beamformer steered to an angle ϕ_0 with our signal of interest at angle ϕ_s. The beamformer passes all signals within its beamwidth with only slight attenuation of signals that are not directly at the center of the beam steered to ϕ_0. As a side effect, this single beamformer cannot discriminate between signals received within its beamwidth. However, we want a resolution finer than the beamwidth in our angle estimate for a signal of interest. Algorithms that acheive such resolution are often called *beamsplitting algorithms*.

To find an angle estimate, we obtain different measurements of the signal of interest. These measurements allow an angle estimation algorithm to discriminate between returns that arrive at the array from different angles. To this end, we use a set of beamformers steered in the general direction of the signal of interest but with different beampatterns, i.e., two beams with offset mainlobes

$$\phi_1 = \phi_0 - \varepsilon \qquad \phi_2 = \phi_0 + \varepsilon$$

where ε is a fraction of the beamwidth (e.g., half a beamwidth). Let the weight vectors for these two beamformers be $\mathbf{c}_1$ and $\mathbf{c}_2$, respectively. These two beamformers can be either nonadaptive or adaptive. Ideally, a pair of adaptive beamformers is used for applications in which interference is encountered. Since the two beamformers are slightly offset from angle ϕ_0, they can be thought of as “left” and “right” beamformers. Using the beamformer weight vectors, we form the ratio

$$\gamma_x = \frac{\mathbf{c}_1^H \mathbf{x}}{\mathbf{c}_2^H \mathbf{x}} \tag{20}$$

where $\mathbf{x}$ is the snapshot under consideration that contains the signal of interest $\mathbf{s} = \sqrt{M}\sigma_s \mathbf{v}(\phi_s)$. Similarly, we can hypothesize this ratio for any angle ϕ to form a discrimination function

$$\gamma(\phi) = \frac{\mathbf{c}_1^H \mathbf{v}(\phi)}{\mathbf{c}_2^H \mathbf{v}(\phi)}. \tag{21}$$

Comparing the value of the measured ratio in equation 20 for the snapshot $\mathbf{x}$ to the angular discrimination function in equation 21, we obtain an estimate of the angle of interest ϕ_s. For this to work, $\gamma(\phi)$ must be bijective.

References

[1] D. Manolakis, V. Ingle and S. Kogon, Statistical and Adaptive Signal Processing, 1st ed. Boston: Artech House, 2005.

[2] M. Sullivan, Practical Array Processing, 1st ed. New York: McGraw-Hill, 2009.

[3] C. Therrien, Discrete Random Signals and Statistical Signal Processing, 1st ed. Englewood Cliffs: Prentice-Hall, 1992.

[4] G. Žitković, 'Theory of Probability,' The University of Texas at Austin, 2016.

[5] H. Van Trees, Optimum Array Processing, 1st ed. New York: Wiley-Interscience, 2002.

[6] A. Manikas, Differential Geometry in Array Processing, 1st ed. London: Imperial College Press, 2004.

A Linear Algebra

Definition A.1. A *Hermitian matrix* is a square matrix equal to its conjugate transpose.

Definition A.2. An $n \times n$ Hermitian M is said to be *positive definite* if the scalar z^*Mz is real and positive for all non-zero column vectors z on n complex numbers, i.e., $z^*Mz > 0$. (z^* denotes the conjugate transpose).

Positive semi-definite implies $z^*Mz \geq 0$. Guess the definitions for negative definite and semi-definite.

Definition A.3. The *eigenvalues* and *eigenvectors* corresponding to an $n \times n$ matrix $\mathbf{M}$ are column vectors $\mathbf{q}$ of length n that satisfy

$$\mathbf{M}\mathbf{q} = \lambda\mathbf{q}.$$

The eigenvectors are the set of $\mathbf{q}$ that satisfy the following and the eigenvalues are the corresponding λ scalar values. Note that the direction of an eigenvector doesn't change under the linear transformation $\mathbf{M}$.

Property A.1 (Eigenvalues and Eigenvectors)**.**

1. Every square matrix over an algebraically closed field has eigenvectors (e.g., $\mathbb{C}$ but not $\mathbb{R}$).
2. Unique eigenvalues correspond to linearly independent eigenvectors.
3. Unique eigenvalues of a Hermitian matrix correspond to orthogonal eigenvectors.

Definition A.4. For a Hermitian matrix M and nonzero vector x, the *Rayleigh quotient* is defined as

$$R(M,x) := \frac{x^*Mx}{x^*x}$$

where x^* denotes the conjugate transpose.

Theorem A.1 (Spectral theorem)**.** *We can decompose any symmetric matrix $A \in S^n$ with the symmetric eigenvalue decomposition*

$$A = U\Lambda U^T, \quad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n).$$

where the matrix of U contains the eigenvectors of A.

Property A.2. The trace of a square matrix is the sum of all eigenvalues (if they exist).

Property A.3. The determinant of a square matrix is equal to the product of all eigenvalues (if they exist).

Definition A.5. A matrix is normal if it commutes with its conjugate transpose, i.e., $A^HA = AA^H$.

Definition A.6 (Singular value decomposition)**.** *Singular value decomposition* (SVD) is a factorization of a real of complex matrix. It is the generalization of the eigendecomposition of a positive semi-definite normal matrix to any $m \times n$ matrix.

Let M be a $m \times n$ matrix whose entries come from $K \in \{\mathbb{R}, \mathbb{C}\}$. Then there exists a factorization, called a singular value decomposition of M, of the form

$$M = U\Sigma V^*$$

where

- U is a $m \times m$ unitary matrix (inverse equal to conjugate transpose).
- Σ is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal.
- V^* is a $n \times n$ unitary matrix over K. V^* is the conjugate transpose of the $n \times n$ unitary matrix V.

The diagonal entries Σ, denoted σ_i, are known as the singular values of M. The columns of U are the left singular vectors, and the columns of V are the right singular vectors. V diagonalizes A^*A and U diagonalizes AA^*, the columns of both are the eigenvectors of A^*A and AA^* respectively.

Diagonalizing a matrix in this way is very convenient and useful in many applications, e.g., we can use the first k singular vectors to approximate a rank r matrix where $k < r$. That is good, believe you me.

B Frequentist and Bayesian Statistics

Frequentist inference is a type of statistical inference that draws conclusions from observed data, and makes assumptions based on the frequency or proportion of the data. This contrasts to Bayesian inference, where we incorporate our beliefs into our prediction then update the parameters of our hypothesis conditioned on new observations.

Definition B.1. In Bayesian statistics, the *prior* probability is our belief of what the parameters of a distribution are before observation, i.e., the prior probability of the parameters θ for some distribution of observations is denoted by $\mathbb{P}[\theta]$. The *posterior* probability is the probability of the observations X conditioned on our prior beliefs, using the previous example, it would be

$$\mathbb{P}[\theta|X] = \frac{\mathbb{P}[X|\theta]\,\mathbb{P}[\theta]}{\mathbb{P}[X]}.$$

Definition B.2. The *likelihood function* is a function of the parameters of a statistical model given data. The *likelihood* of a set of parameter values θ given outcomes X is equal to the probability of those observed outcomes given those parameter values, i.e.,

$$L(\theta|X) = \mathbb{P}[X|\theta].$$

Remark. Note that the posterior is proportional to the likelihood times the prior. The denominator is just a (but the right) normalizing term such that the posterior is still a probability measure (posterior $\in [0,1]$).

Definition B.3. The *maximum likelihood estimation* (MLE) is a method of estimating the parameters of a statistical model, given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters. In MLE, we seek a point value for θ which maximizes the likelihood function. Let's denote this $\hat{\theta}$. Note that since $\hat{\theta}$ is a point estimate, it is not a random variable. Thus, we cannot inject our prior beliefs about the likely values for θ in the estimation calculation. Thus MLE is associated with frequentist statistics (although it is heavily used in Bayesian as well).

It turns out that MLE is really just counting the number of positive (or whatever) occurances in a data set and normalizing by the dataset and assigning that as the probability of that positive (or whatever) occurance happening again. To see this, notice that clearly this maximizes the likelihood of the parameters given the observations. Thus this is just a fancy name for a very simple concept.

As an equation the MLE can be described as

$$\hat{\theta}_{\mathsf{ML}}(X) = \arg\max_{\theta} f(X|\theta)$$

where f is the sampling distribution (density) of X and $f(X|\theta)$ is the probability of x when the underlying population parameter is θ. Thus $\theta \mapsto f(X|\theta)$ is the likelihood function, as described in Definition B.2.

Definition B.4. The *maximum a posteriori estimation* (MAP) is an estimate of an unknown quantity, that equals the mode of the posterior distribution. It is similar to the MLE but incorporates a prior distribution over the quantity one wants to estimate.

Assume that a prior density g over θ exists. Then we can treat θ as a random variable (i.e. we are in the realm of Bayesian statistics). Then we can calculate the posterior distribution of θ using Bayes' theorem

$$\theta \mapsto f(\theta|X) = \frac{f(X|\theta)g(\theta)}{\int_{\vartheta\in\Theta} f(x|\vartheta)g(\vartheta)\,d\vartheta}$$

where Θ is the domain of g.

The method of MAP estimation then estimates θ as the mode of the posterior distribution of this random variable

$$\hat{\theta}_{\text{MAP}}(X) = \arg\max_{\theta} f(\theta|X) = \arg\max_{\theta} \frac{f(X|\theta)g(\theta)}{\int_{\vartheta} f(X|\vartheta)g(\vartheta)\,d\vartheta} = \arg\max_{\theta} f(X|\theta)g(\theta).$$

(the denominator is a constant which is why we can remove it from the optimization).

Remark. MLE often overfits data (variance of parameter estimates is high), to combat this we can *regularize* the MLE (introduce bias to the estimate) through MAP (assume the parameters are a random variable, of which we have prior beliefs). Note that MAP is a generalization of MLE and the two are equal when the prior g is uniform.

Definition B.5. The *expectation-maximization algorithm* (EM algorithm) is an iterative statistical estimation procedure to address the incomplete or missing data or parameter estimation problem. Given some observation data X, latent (unobserved) variables λ, and a model parameterized by θ, the EM algorithm solves the following optimization problem

$$\arg\max_{\theta}\ \mathbb{E}[\log L(\theta|X,\lambda)] \quad \text{for} \quad (X,\lambda)|\hat{\theta}^{(n-1)}$$

where $L(\theta|X,\lambda) = \mathbb{P}(X,\lambda|\theta)$. The $\log$ operator is applied to make this an easier, and more numerically stable, maximization ($\log$ makes many distributions concave, i.e., maximization will result in the global maximum).

Definition B.6 (Principal component analysis)**.** *Principal component analysis* (PCA) is often used to reduce the dimensionality of data while preserving most of the variation in the dataset (e.g., save space, make less computationally expensive algorithms). This is sometimes (mostly in signal processing) called the *Karhunen-Loéve transform*.

To compute PCA:

1. Organize data into an $m \times n$ matrix, where m is the number of measurement types and n is the number of samples.
2. Subtract the mean from each measurement type.
3. Calculate the SVD or the eigenvectors of the covariance.

Note that the singular values produced from SVD are the square root of the eigenvalues of the covariance matrix and the right singular vector matrix (denoted V in A.6) are the principal components.

We can reduce the dimension of X by only using the first k principal components of X (eigenvectors of $\text{Cov}(X)$) corresponding to the first k largest eigenvalues.

C More Math

Definition C.1. A *topology* on a set X is a family τ of subsets of S which contain $\varnothing$ and X and is closed under finite intersection and arbitrary union. The elements of τ are often called the *open sets*. A set X on which a topology is chosen, i.e., (X, τ) is called a topological space.

In other words, a topology just a way to define open sets on a space (with some additional structure).

Definition C.2. A *random field* is a generalization of a stochastic process, i.e., a stochastic process which is indexed over a parameter space of dimension greater than (or equal to) one.

Theorem C.1 (Wiener-Khinchin Theorem)**.** *For a wide-sense stationary stochastic process, the Fourier transform of the autocorrelation is equal to the power spectrum.*

Theorem C.2 (Taylor's Theorem)**.** *Let $k \in \mathbb{Z}_{\geq 1}$ and let the map $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ such that*

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

and $\lim_{x \to a} h_k(x) = 0$. (This is called the Peano form of the remainder*).*