Stefan Trampitsch

Complex-Valued Data Estimation
Second-Order Statistics and Widely Linear Estimators

Masterarbeit

zur Erlangung des akademischen Grades
Diplom-Ingenieur

Studium Information Technology

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften

Begutachter: Prof. Mario Huemer
Betreuer: Prof. Mario Huemer

Institut für Vernetzte und Eingebettete Systeme

Klagenfurt, im April 2013
Ehrenwörtliche Erklärung

Ich declare ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiter, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben.


Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

(Unterschrift)                                                 (Ort, Datum)
Abstract

This thesis presents the fundamental concepts of complex-valued data estimation. Theory as well as a number of applications are presented. With the introduction of complex-valued entities in digital signal processing, complex-valued estimation these has become an interesting research topic. The main focus is on second-order statistics and widely linear Bayesian estimators. By maintaining the link to real-valued estimation theory, well known knowledge can be transferred to the complex case. The work in this thesis is intended to provide an intuitive approach on the most important techniques required for complex-valued data estimation. A quite sophisticated mathematical framework allows access to all statistical properties directly in the complex domain. Widely linear estimators are quite new in the field of statistical signal processing. Advantages and disadvantages of strictly linear and widely linear estimators are highlighted by the helpf of simulations for a range of application examples.
# Contents

1 Introduction ................................. 1
   1.1 Motivation and Aims .................... 2
   1.2 Overview ................................ .. 2

2 Basic Concepts ............................. 4
   2.1 Complex Analysis and CR-Calculus ..... 4
      2.1.1 Complex Derivatives ............... 5
      2.1.2 CR-Calculus .......................... 7
      2.1.3 Complex Integrals - Green’s Theorem . 10
   2.2 Augmented Representation ............... 10
      2.2.1 Alternative Representations ......... 10
      2.2.2 Inner Products and Quadratic Forms . 11
   2.3 Widely Linear (WL) Transformations ..... 12
   2.4 Complex FIR Filter ...................... 14
   2.5 Hilbert Space ........................... 15

3 Complex Statistics .......................... 17
   3.1 Complex pdf ............................ 18
   3.2 Complex Moments ....................... 21
   3.3 Second-Order Statistics ................. 23
      3.3.1 Properties of the Covariance Matrices . 27
      3.3.2 Properness and Circularity .......... 29
      3.3.3 Complex Correlation Coefficient .... 35
   3.4 Complex Gaussian pdf ................... 35
      3.4.1 Important Properties of the Proper Complex Multivariate Gaussian pdf .......... 37
      3.4.2 Effects of Improperness on Complex Gaussian pdf ................. 38
      3.4.3 Conditional Gaussian pdf ............. 40
   3.5 Complex Random Processes ................ 42

4 Estimators ................................. 49
   4.1 Bayesian Estimators .................... 49
      4.1.1 Complex LMMSE ....................... 52
      4.1.2 WLMMSE .............................. 56
      4.1.3 Comparison of LMMSE and WLMMSE .... 67
   4.2 Classical Estimators for Proper Signals .... 71
4.2.1 Minimum Variance Unbiased (MVU) Estimator ........................................ 71
4.2.2 Complex Linear Model and BLUE ..................................................... 72
4.2.3 Least Squares (LS) Estimation ......................................................... 73

5 Applications for Complex-Valued Data Estimation ........................................ 75
5.1 Estimation of a Constant Complex Envelope ........................................... 75
5.2 Advantages of WLMMSE for CP-OFDM and UW-OFDM ............................. 78
  5.2.1 CP-OFDM ......................................................................................... 82
  5.2.2 UW-OFDM ......................................................................................... 85
  5.2.3 Conclusion ......................................................................................... 85
5.3 Channel Estimation with LMS Algorithm ................................................... 88

A Derivations .................................................................................................. 99
A.1 Derivation of CR Gradients ........................................................................ 99
  A.1.1 Gradient of a Linear Form .................................................................. 99
  A.1.2 Gradient of Hermitian Form ............................................................... 100
A.2 Derivation of $H$ ...................................................................................... 101
A.3 Inner Products .......................................................................................... 102
A.4 ACF and CCF .......................................................................................... 103
A.5 Derivation of the WLMMSE for Proper Signals via Orthogonality Principle .. 104

B MATLAB scripts ....................................................................................... 106
B.1 Complex-Valued Gaussian Random Vector .............................................. 106
B.2 WLMMSE vs. LMMSE ............................................................................. 107
B.3 MATLAB script for CP-OFDM ............................................................... 109
B.4 MATLAB script for UW-OFDM ............................................................... 114

Bibliography ................................................................................................. 121
Chapter 1

Introduction

Complex-valued signals or variables occur in many areas of engineering. In signal processing, narrowband signals are analytically described in the equivalent complex baseband representation [17]. As the name already implies, it introduces complex entities. Figure 1.1 illustrates the frequency representation of a passband and a baseband signal. Digital processors benefit from working with the latter one, as a low sampling frequency is sufficient. Further, models are quite easy to implement in hardware. Thus it saves a huge amount of energy and calculation performance. Estimation of those complex signals has become a significant part of digital signal processing [19, 16, 2]. One of the advantages of using complex signals is the combination of two real-valued variables to one complex-valued, while still keeping a rather simple analytical model to work with. This approach provides insights, which would not be visible at first look. A complex random variable is denoted by \( x = x_r + j x_i \), where \( x_r \) and \( x_i \) represent the real and imaginary part. As \( x_r \) and \( x_i \) are real-valued random variables, their statistical behavior can be completely described by a real probability density function (pdf). The extension to complex estimation theory requires complex pdfs. Although one calls it complex pdf, it is always a real-valued function. With it, it is possible to derive characteristics like the mean or the variance for complex-valued variables. In the past view decades, complex signals have been assumed to be proper. With this, simple closed form solutions can be derived, which are very similar to their real-valued relatives. Nevertheless, there are cases where proper signals are very poor models. Properness is a characteristic which is determined by second-order statistics. Only if the pseudo-variance vanishes (\( x \) uncorrelated to \( x^* \)), a variable/vector is called proper. Otherwise, it is called improper. Unfortunately, signals are rarely proper in real world applications. Recent publications [28, 26] have altered the knowledge of complex statistics to a new level. By keeping the link between real-valued and complex-valued random variables, new types of statistics in \( \mathbb{C} \) can be accessed. To achieve these, the augmented representation [29] is introduced. It can be advantageously used to yield even better estimation results. These ones are optimized for improper signal constellations. It is because of the relation between real and complex domain which yields to widely linear transformations. There, the transformed variable is linear in the original one and in its conjugate, i.e. \( y = a x + b x^* \). Estimators, which exploit this concept are called widely linear estimators. These have key benefits regarding estimation performance, but drawbacks in operation expense [27]. Anyhow, if used in digital communication systems, like OFDM, realizations of equalizers can be significantly improved. Possibilities are not just limited
CHAPTER 1. INTRODUCTION

Figure 1.1: Frequency domain representation of a real-valued bandpass and a complex-valued baseband signal.

(a) Bandpass. (b) Baseband.

to signal estimation only, but also to well known control system applications [5, 4] or even to neural network modeling [22].

Not only statistical properties, but also the fundamental concept of deriving estimators are of great interest. It is often the case that the best one is given by the solution of an extrema (usually minimum) of a specific cost function. Basic mathematical textbooks like [20] only work with complex analytic signals. However, the cost function of a linear minimum mean square error (LMMSE) estimator is a real-valued function, $J = E(|e|^2) \in \mathbb{R}$. From definition [20], it is therefore nonanalytic. By the revolutionary work of Wilhelm Wirtinger [34] the Wirtinger calculus or CR-Calculus provides the mathematical framework to find minima of nonanalytic functions in $\mathbb{C}$. The concepts have been especially used on complex random variables by van den Bos and Schreier [26, 2], resulting in closed form solutions for estimators in the complex domain. The algorithms derived may then be implemented and used in real world applications.

1.1 Motivation and Aims

The focus of this thesis is on establishing a detailed knowledge in complex estimation. Most basic engineering books only provide insights in estimation in the real domain $\mathbb{R}$. Whereas, in digital signal processing the focus is shifted towards complex-valued signals. Estimation of these signals is not always a straightforward approach. A detailed background is required to achieve a performance gain when estimating in $\mathbb{C}$. Utilizing this knowledge can be used to transform complicated real-valued algorithms in quite simple complex ones. Beside all theoretical aspects, it is also of great interest to find real world applications to apply the acquired results. Digital signal processing applications offer many possibilities to test the derived algorithms. Simulations in this thesis shall provide a good feeling of how the estimators work in a real world environment.

1.2 Overview

This thesis is organized as follows. In Chapter 2 the theoretical basics are introduced. The presented CR-Calculus provides the mathematical framework to find minima of complex functions, allowing derivations of estimators directly in $\mathbb{C}$. Different representations of a complex variable are introduced, offering an intuitive view on the problem of complex-
valued estimation. Also, the link between this representations is shown. A linear transformations in $\mathbb{R}$ results in a widely linear transformation in $\mathbb{C}$. Chapter 3 is the first which deals with random variables. Most results and properties are first derived for real-valued random variables and then altered to complex ones. The focus is on second-order statistics, which are needed to characterize a signal as proper or improper. These properties greatly influence the derivation and complexity of estimators. It is shown which conditions must hold for a random variable/vector to be called proper. Further, the Gaussian pdf is derived in detail. Figures and diagrams are used to intuitively show the behavior of a complex pdf. Also the effects of improperness on the Gaussian pdf are illustrated. At last, complex random processes are introduced. In Chapter 4 complex estimators are established. The chapter splits in Bayesian and classical estimation approaches. The focus of the latter one is on proper signals only. The Bayesian way is limited to the (widely) linear case. The introduced widely linear estimators are the extension of the strictly linear ones, which may already be known from basic estimation theory. These yield superior outcomes if signals are characterized as improper. Chapter 5 uses the derived theoretical results on real world applications, where the focus is on widely linear vs. strictly linear estimators. Finally, detailed derivations and MATLAB scripts are part of appendix A and B.

Notation:

- $\mathbf{x}$ ... vector
- $\mathbf{X}$ ... matrix
- $\hat{x}$ ... estimate
- $\mathbf{x}$ ... augmented representation
- $(\cdot)^T$ ... transpose
- $(\cdot)^H$ ... Hermitian transpose / conjugate transpose
- $\mathbb{R}$ ... set of real numbers
- $\mathbb{C}$ ... set of complex numbers
Chapter 2

Basic Concepts

This chapter is intended to give the reader a comprehensive introduction to the fundamentals, which are required to understand the concepts of complex signal estimation covered by this thesis. Often, the task in estimation theory is to find the extrema of a specific function in order to derive optimal estimators. Analytically, these extrema can be found by differentiating with respect to the parameter(s) which shall minimize/maximize the function. However, when the parameter or the function is no longer part of $\mathbb{R}$, it becomes difficult. Thus it is important to understand the rules and ways of complex differentiation. Section 2.1 gives an overview of the basics of complex analysis. Then, a deeper look on the complex derivatives is presented, which may be known from (classical) mathematical textbooks like [20]. These concepts require holomorphic functions. As cost functions are usually not holomorphic, a different approach must be used. Therefore the concepts of the CR-Calculus are introduced in Section 2.1.2. It allows derivation of estimators directly in the complex domain. Apart from complex analysis, an additional quite necessary theory is introduced in Section 2.2. The so called augmented representation explains how all required statistical properties can be described in $\mathbb{C}$. This section is crucial to understand analysis of complex random variables. Beside mathematical theory of complex signals and vectors, also one specific signal processing application is extended to the complex domain. Section 2.4 shows the underlying model of a complex FIR filter. It points out, that analytical operations of complex-valued signals must be realized with real-valued operations.

2.1 Complex Analysis and CR-Calculus

The complex plane $\mathbb{C}$ is an extension to the real line $\mathbb{R}$. A complex number is defined as $x = x_r + j x_i$, where $j = \sqrt{-1}$ is the imaginary unit and $x_r, x_i$ are the real and imaginary part, respectively. In most mathematical references the symbol $j$ is replaced with $i$. Representing a complex number with its real and imaginary part is called the Cartesian representation, where it represents a vector or point in a Cartesian coordinate system. It is also possible to use the magnitude and the angle to represent the complex number. This is called the polar representation. Although nothing changes, it is much easier to work with the polar representation when applying multiplications and divisions.
CHAPTER 2. BASIC CONCEPTS

Magnitude and angle (or argument) of a complex number are given as:

\[ |x| = \sqrt{x_r^2 + x_i^2} \in \mathbb{R}, \quad (2.1) \]
\[ \theta = \arctan \left( \frac{x_i}{x_r} \right) \in \mathbb{R}, \quad (2.2) \]

Thus \( x = x_r + j x_i \Leftrightarrow x = |x| \exp(j\theta) \). Complex numbers are not ordered like real numbers, which can be represented as points on an infinite straight line \([20]\). Thus \( x_1 > x_2 \) is not possible. Instead, the magnitude \( |x_1| > |x_2| \) and angle \( \theta_1 < \theta_2 \) can be used. Beside the different representations, it is of great interest to calculate real and imaginary part of \( x \). This can be achieved by using complex conjugates, which are defined as \( x^* = x_r - j x_i \).

The conjugate operation \((\cdot)^*\) mirrors \( x \) about the real axis. Using the conjugate allows one to calculate the real and imaginary part as follows

\[ x_r = \frac{1}{2} (x + x^*) \quad (2.3) \]
\[ x_i = \frac{1}{2j} (x - x^*) \quad (2.4) \]

For convenience, \( \Re(x) \) and \( \Im(x) \) are further used as short form of (2.3) and (2.4). Following list shows a very short review of common complex operations, where \( x = x_r + j x_i \) and \( y = y_r + j y_i \):

- \( j^2 = -1 \)
- \( x \pm y = x_r \pm y_r + j (x_i \pm y_i) \)
- \( x \cdot y = x_r y_r - x_i y_i + j (x_r y_i + x_i y_r) = |x| |y| \exp [j(\theta_x + \theta_y)] \)

2.1.1 Complex Derivatives

The field of complex analysis deals with complex functions and their operations and properties. A complex function is defined as:

\[ y = f(x) = y_r(x_r, x_i) + j y_i(x_r, x_i); \in \mathbb{C}, \quad (2.5) \]

where \( y_r(x_r, x_i), y_i(x_r, x_i) \) are real functions, \( y_r, y_i : \mathbb{R}^2 \to \mathbb{R} \). Functions like (2.5) are in general complex, but may be real-valued in special cases, e.g.: squared error cost function \( J(|e|^2) \). An important field of complex analysis is the theory of limits. A limit is the value which a function \( f(x) \) approaches as \( x \) approaches a certain value \( x_0 \). The definition in \( \mathbb{C} \) is the very same as in the real case. A complex function is said to have a limit \( l \) as \( x \) approaches a point \( x_0 \)

\[ \lim_{x \to x_0} f(x) = l, \quad (2.6) \]

if \( f(x) \) is defined in a neighborhood of \( x_0 \) and if values of \( f(x) \) are close to \( l \) for all \( x \) close to \( x_0 \) \([20]\). A neighborhood is the area inside a disk, defined as \( |x - x_0| < \mu, \) where \( \mu > 0, \in \mathbb{R} \). A neighborhood is also called an open circular disk. For more details on complex limits see \([20]\). A very intuitive explanation of this definition is shown in Figure 2.1.
The derivative of a complex function \( f(x) \) at point \( x_0 \) is defined as
\[
f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}.
\] (2.7)
If that limit exists, then \( f(x) \) is differentiable at point \( x_0 \). Using the relation \( \Delta x = x - x_0 \) the derivative can be rewritten as
\[
f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}.
\] (2.8)

It is important to remember that, as already explained above, \( f(x) \) has to be defined in a neighborhood of \( x_0 \). So \( x \) can approach \( x_0 \) from any direction. Differentiability at \( x_0 \) means that, along whatever path \( x \) approaches \( x_0 \), the quotient of (2.8) always approaches a certain value and all these values are equal. Or in other words, (2.7) requires that \( f'(x) \) must be independent of the direction \( \Delta x \) approaches 0 [2]. A function that is complex-differentiable on its entire domain is called an analytic or holomorphic function. Following example shows the differentiation of \( f(x) = x^2 \):
\[
f'(x) = \lim_{\Delta x \to 0} \frac{(x + \Delta x)^2 - x^2}{\Delta x}
= \lim_{\Delta x \to 0} (2x + \Delta x)
= 2x
\] (2.9)

The Cauchy-Riemann equations can be used to prove if \( f(x) \) is holomorphic, and hence differentiable. The Cauchy-Riemann equations are
\[
\frac{\partial y_r}{\partial x_r} = \frac{\partial y_i}{\partial x_i},
\] (2.10)
\[
\frac{\partial y_r}{\partial x_i} = -\frac{\partial y_i}{\partial x_r}.
\] (2.11)
These conditions are necessary for \( f(x) \) to be complex-differentiable. Further it is required for \( y_r(x_r, x_i) \) and \( y_i(x_r, x_i) \) to be differentiable with respect to \( x_r \) and \( x_i \). Following example shows that \( f(x) = x^2 \) is a holomorphic function.

**Example 2.1.1.** Let \( f(x) = x^2 \) be a complex function with \( x = x_r + jx_i \). Then

\[
f(x) = x^2 = x_r^2 - x_i^2 + 2x_r x_i
\]

Applying the differentiations

\[
\frac{\partial y_r}{\partial x_r} = 2x_r = \frac{\partial y_i}{\partial x_i} = 2x_r
\]
\[
\frac{\partial y_r}{\partial x_i} = -2x_i = -\left( \frac{\partial y_i}{\partial x_r} = 2x_i \right)
\]

shows that the Cauchy-Riemann equations hold and hence \( f(x) = x^2 \) is a holomorphic function.

If the Cauchy-Riemann equations hold, the same differentiation rules can be used as in real calculus. The theory of differentiating complex analytic functions is straightforward, but there are also non-analytic functions of interest. A simple example of a non-analytic function is \( y = f(x) = x x^* = |x|^2 \). This one is real-valued and therefore \( y_i = 0 \). Consequently the Cauchy-Riemann equations do not hold and \( f(x) \) is not differentiable with the definitions described above.

### 2.1.2 \( \text{CR-} \) Calculus

The \( \text{CR-} \) Calculus, also known as Wirtinger calculus, provides a way to differentiate non-analytic complex functions. It was introduced by Wilhelm Wirtinger in 1927 [34]. The \( \text{CR-} \) Calculus only requires, that a function, expressed as \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \), is differentiable. Such a function \( f(\cdot) \) is then called real-differentiable. Real-differentiability of a function \( f(x) = y_r(x_r, x_i) + j y_i(x_r, x_i) \) is possible if and only if \( y_r(x_r, x_i) \) and \( y_i(x_r, x_i) \) have continuous partial derivatives with respect to \( x_r \) and \( x_i \). The general \( \text{CR} \) derivatives are defined as

\[
\frac{\partial f(x)}{\partial x} = \frac{1}{2} \left( \frac{\partial f(x)}{\partial x_r} - j \frac{\partial f(x)}{\partial x_i} \right) \tag{2.12}
\]
\[
\frac{\partial f(x)}{\partial x^*} = \frac{1}{2} \left( \frac{\partial f(x)}{\partial x_r} + j \frac{\partial f(x)}{\partial x_i} \right) \tag{2.13}
\]

where \( \frac{\partial f(x)}{\partial x_r} \) and \( \frac{\partial f(x)}{\partial x_i} \) are real derivatives. For real-valued functions, following condition holds

\[
\left( \frac{\partial f(x)}{\partial x} \right)^* = \frac{\partial f(x)}{\partial x^*}, \tag{2.14}
\]

because as \( f(x) \in \mathbb{R}, f(x)^* = f(x) \). A complex function generally depends on the real and imaginary part of \( x \), or \( f(x) = f(x_r, x_i) \). When using (2.3) and (2.4) real and imaginary part can be expressed with \( x \) and its conjugate \( x^* \). One can therefore regard \( f \) as a function of \( x \) and \( x^* \), i.e. \( f(x_r, x_i) \equiv f(x, x^*) \). Note, \( x_r \) and \( x_i \) are not substituted by \( x \) and \( x^* \),
but expressed by (2.3) and (2.4). This relation is very important when working with the \(\CR\)-Calculus, because it can be shown that the derivatives in (2.12) and (2.13) can be formally implemented by treating \(x\) and \(x^\ast\) as independent from each other. As a result from this, when taking the \(\CR\) derivative of \(f(x, x^\ast)\) with respect to \(x\), then \(x^\ast\) is treated as a constant, and vice versa

\[
\frac{\partial x}{\partial x} = 1; \quad \frac{\partial x}{\partial x^\ast} = 0; \quad \frac{\partial x^\ast}{\partial x} = 0; \quad \frac{\partial x^\ast}{\partial x^\ast} = 1; \quad \tag{2.15}
\]

**Example 2.1.2.**

\[
f(x, x^\ast) = x^2 x^\ast
\]

\[
\frac{\partial f(x, x^\ast)}{\partial x} = 2xx^\ast
\]

\[
\frac{\partial f(x, x^\ast)}{\partial x^\ast} = x^2
\]

One can show that the Cauchy-Riemann equations can simply be stated as

\[
\frac{\partial f(x, x^\ast)}{\partial x^\ast} = 0.
\]

Because of that, an analytic function cannot depend on \(x^\ast\).

**Multivariate \(\CR\)-Calculus**

The \(\CR\) differentiation rules written above are only suited for the scalar case. It is often of great interest to differentiate functions a vector in a complex vector space \(\mathbb{C}^N\). One example is the cost function of the least squares (LS) estimator for more than one parameter. The \(\CR\)-Calculus can be extended to the vector case, where the parameters to be differentiated are vectors. This is called Multivariate \(\CR\)-Calculus. The basic rules of the \(\CR\) remain the same. Given the complex vector \(x \in \mathbb{C}^N\), the gradient of a complex function is given as

\[
\nabla_x f(x) = f'(x) = \frac{\partial f(x)}{\partial x} = \left[ \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_N} \right], \quad \tag{2.16}
\]

where each derivative is (2.12). Obviously, the gradient of a scalar complex function produces a vector field [20]. The gradient yields the direction of the largest increase of function \(f(x)\). Additionally, the magnitude of the resulting vectors of \(f'(x)\) yield the degree of increase at each point. As in the scalar case, also the multivariate \(\CR\) treats \(x\) and \(x^\ast\) as independent.

**Example 2.1.3.** Let a complex function \(f(x)\) be given as

\[
f(x, x^\ast) = x_1 x_2^\ast + x_1^\ast x_2
\]

then the complex gradients with respect to \(x\) and \(x^\ast\) are

\[
\frac{\partial f(x)}{\partial x} = \left[ \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2} \right] = [x_2^\ast x_1^\ast] \quad \tag{2.18}
\]
\[
\frac{\partial f(x)}{\partial x^\ast} = \left[ \frac{\partial f(x)}{\partial x_1^\ast}, \frac{\partial f(x)}{\partial x_2^\ast} \right] = [x_2 x_1] \quad \tag{2.19}
\]
CHAPTER 2. BASIC CONCEPTS

If given a complex function \( f(x, x^*) \in \mathbb{C}^M \) with complex vectors \( x, x^* \in \mathbb{C}^N \), the gradient is the Jacobi matrix

\[
J_f(x, x^*) = \frac{\partial f(x, x^*)}{\partial x} = \begin{bmatrix}
\frac{\partial f_1(x, x^*)}{\partial x_1} & \frac{\partial f_1(x, x^*)}{\partial x_2} & \ldots & \frac{\partial f_1(x, x^*)}{\partial x_N} \\
\frac{\partial f_2(x, x^*)}{\partial x_1} & \frac{\partial f_2(x, x^*)}{\partial x_2} & \ldots & \frac{\partial f_2(x, x^*)}{\partial x_N} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_M(x, x^*)}{\partial x_1} & \frac{\partial f_M(x, x^*)}{\partial x_2} & \ldots & \frac{\partial f_M(x, x^*)}{\partial x_N}
\end{bmatrix},
\]

and

\[
J^C_f(x, x^*) = \frac{\partial f(x, x^*)}{\partial x^*} = \begin{bmatrix}
\frac{\partial f_1(x, x^*)}{\partial x^*_1} & \frac{\partial f_1(x, x^*)}{\partial x^*_2} & \ldots & \frac{\partial f_1(x, x^*)}{\partial x^*_N} \\
\frac{\partial f_2(x, x^*)}{\partial x^*_1} & \frac{\partial f_2(x, x^*)}{\partial x^*_2} & \ldots & \frac{\partial f_2(x, x^*)}{\partial x^*_N} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_M(x, x^*)}{\partial x^*_1} & \frac{\partial f_M(x, x^*)}{\partial x^*_2} & \ldots & \frac{\partial f_M(x, x^*)}{\partial x^*_N}
\end{bmatrix}.
\]

\( J_f(x, x^*) \neq J^C_f(x, x^*) \) in general. Only if \( f(x, x^*) \) is a real-valued function, then \( J_f(x, x^*)^* = J^C_f(x, x^*) \), because

\[
J_f(x, x^*)^* = \left( \frac{\partial f(x, x^*)}{\partial x} \right)^* = \frac{\partial f(x, x^*)}{\partial x^*} = J^C_f(x, x^*)
\]

The following list shows the most important gradients used in this thesis for linear forms \( a = [a_1 \cdots a_N]^T \in \mathbb{C}^N \) and Hermitian forms \( A = A^H \in \mathbb{C}^{N \times N} \). The differentiations are with respect to the complex vector \( x = [x_1 \cdots x_N]^T \).

\[
\frac{\partial a^H x}{\partial x} = x^T a^* = a^*
\]
\[
\frac{\partial x^H a}{\partial x} = a^T x^* = 0
\]
\[
\frac{\partial a^T x}{\partial x} = x^T a = a
\]
\[
\frac{\partial x^H a}{\partial x} = a^T x^* = a
\]
\[
\frac{\partial a^H x}{\partial x^*} = x^T a^* = 0
\]
\[
\frac{\partial x^H a}{\partial x^*} = a^T x = 0
\]
\[
\frac{\partial x^H A x}{\partial x} = (A x)^* = A^T x^*
\]
\[
\frac{\partial x^H A x}{\partial x^*} = A x
\]
\[
\frac{\partial x^H x}{\partial x} = x^*
\]

To get a feeling how to work with complex gradients, detailed derivations of (2.22) and (2.28) are given in appendix A.1. A very good example demonstrating the benefits of the CR-Calculus is the derivation of the LS estimator, shown in Section 4.2.3.
2.1.3 Complex Integrals - Green's Theorem

In estimation theory it is often required to evaluate integrals. These must be handled differently in the complex domain. It can be shown that the integral of a function \( f(x_r, x_i) \) in a region \( R \) possesses the form \( \int \int_R f(x_r, x_i) \, dx_r \, dx_i \). By using Green’s Theorem or Stoke’s Theorem \([20, 1]\) the real-valued integral can be written as a contour integral in the complex domain

\[
\int \int_R f(x_r, x_i) \, dx_r \, dx_i = \frac{1}{2} j \oint_C F(x, x^*) \, dx,
\]

(2.31)

where

\[
\frac{\partial F(x, x^*)}{\partial x^*} = f(x, x^*).
\]

(2.32)

It is required that \( f(x_r, x_i) \) is continuous through the simply connected region \( R \), where \( C \) describes its contour \([28]\).

2.2 Augmented Representation

2.2.1 Alternative Representations

A complex vector \( \mathbf{x} \) can be described by its real and imaginary part. It can also be represented by creating an equivalent real vector \( \mathbf{x}_R \). Thus, a relation is introduced between the complex and the real vector:

\[
\mathbf{x} = x_r + j x_i \in \mathbb{C}^N \iff \mathbf{x}_R = \begin{bmatrix} x_r \\ x_i \end{bmatrix} \in \mathbb{R}^{2N}.
\]

(2.33)

It is obvious that the real vector has twice the dimension compared to the complex one. Intuitively, a complex variable can be represented by a point/vector in a Cartesian coordinate system, as already explained in 2.1. This technique is often used in signal processing applications, e.g.: modulation techniques. As an example, in hardware two real data streams (I- and Q-path) are analytically represented as one complex data stream. Hence, it is possible to represent functions, which depend on \( \mathbf{x} \), also as functions depending on \( x_r \) and \( x_i \), i.e. \( f(\mathbf{x}) \equiv f(x_r, x_i) = f(\mathbf{x}_R) \). Note, \( f(\mathbf{x}) \) to \( f(x_r, x_i) \) is not a parameter substitution. In fact, \( f(x_r, x_i) \) is another function and should be denoted like \( g(x_r, x_i) \). But as \( f(\mathbf{x}) \) and \( f(x_r, x_i) \) yield the very same result this will be neglected further on.

Example 2.2.1.

\[
f(x) = |x|^2 = x_r^2 + x_i^2 = [x_r x_i] \begin{bmatrix} x_r \\ x_i \end{bmatrix} = \mathbf{x}_R^T \mathbf{x}_R = f(\mathbf{x}_R)
\]

It will be shown that the augmented representation is required to describe linear transformations on \( \mathbf{x}_R \) in \( \mathbb{C} \) and to provide full access to second-order statistics of complex random variables. The augmented vector is defined as

\[
\mathbf{x} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}
\]

(2.34)

Although (2.34) has twice the dimension as \( \mathbf{x} \), it only spans a subspace of \( \mathbb{C}^{2N} \) \([2]\). The subspace spanned by \( \mathbf{x} \) will be denoted as \( \mathbb{C}_2^{2N} \), the \( 2N \) is used for better readability of the
dimensions of augmented vectors and matrices. The relation between the real vector \( x_R \) and the augmented vector \( x \) can be described by the linear real-to-complex transformation

\[
x = T_N x_R.
\]  

(2.35)

\( T_N \) is a linear transformation matrix, defined as

\[
T_N = \begin{bmatrix} I_N & jI_N \\ I_N & -jI_N \end{bmatrix} \in \mathbb{C}^{2N \times 2N},
\]  

(2.36)

where \( I_N \) is the \( N \)-dimensional identity matrix. \( T_N \) transforms the coordinates from the real to the augmented space without loss of information, \( T_N : \mathbb{R}^{2N} \rightarrow \mathbb{C}_2^{2N} \). The transformation matrix was initially introduced by van den Bos [29, 30] and formalized as real-to-complex transformation by Schreier [2]. If dimensions of \( T \) are obvious from context, the \( N \) sign will be dropped. As there is a bivariate relation between \( x_R \) and \( x \) the transformation can also be done the other way around \( x_R = T^{-1} x \). For that, the inverse of \( T \) is needed. By applying the matrix inversion lemma [20]

\[
T^{-1} = \frac{1}{2} T^H.
\]  

(2.37)

The transformation matrix is unitary up to a factor of two, which means that \( T T^H = 2I \). A square matrix is unitary if \( U U^H = I \). This provides an alternative approach to show that the inverse of \( T = \frac{1}{2} T^H \):

\[
T T^H = 2I \\
T^{-1} T T^H = T^{-1} 2I \\
T^H = 2 T^{-1} \\
T^{-1} = \frac{1}{2} T^H
\]

Using (2.37) it is easy to show that

\[
x_R = \frac{1}{2} T^H x.
\]  

(2.38)

To conclude, the augmented vector \( x \) provides a way to represent the real vector \( x_R \) in the complex domain. Of course, \( x \) yields some redundancy, but it is a convenient method to work within \( \mathbb{C} \).

\subsection*{2.2.2 Inner Products and Quadratic Forms}

With all kind of representations, it is of great interest to see how inner products and quadratic forms behave in \( \mathbb{R}, \mathbb{C} \) and \( \mathbb{C}_2 \). An inner product (dot product, scalar product) on \( \mathbb{R}^N \) is defined as [20]

\[
c = a \cdot b = \sum_{n=1}^{N} a_n b_n = a^T b,
\]  

(2.39)
where \( c \) is a real-valued scalar and \( \mathbf{a} \) and \( \mathbf{b} \) are \( N \)-dimensional real-valued vectors. Geometrically interpreted the dot product, which is then called an inner product, can be rewritten as \( c = |\mathbf{a}| |\mathbf{b}| \cos(\theta) \), where \(| \cdot |\) is the magnitude of a vector and \( \theta \) the angle spanned by \( \mathbf{a} \) and \( \mathbf{b} \). The inner products for all representations are related as follows

\[
\mathbf{x}^T \mathbf{R} \mathbf{y} = \Re(\mathbf{x}^H \mathbf{y}) = \frac{1}{2} \mathbf{x}^H \mathbf{y},
\]

where \( \mathbf{x}^T \mathbf{R} \mathbf{y} \) is the inner product on \( \mathbb{R}^{2N} \), \( \mathbf{x}^H \mathbf{y} \) the inner product on \( \mathbb{C}^N \) and \( \mathbf{x}^H \mathbf{y} \) the inner product on \( \mathbb{C}^{2N} \). Proofs of (2.40) are shown in appendix A.3.

A further, rather important mathematical form is the quadratic form. For real-valued vectors it is defined as

\[
q = \mathbf{a}^T \mathbf{B} \mathbf{a} = \sum_{n=1}^{N} \sum_{m=1}^{N} b_{nm} a_n a_m,
\]

where \( \mathbf{a} \in \mathbb{R}^N \) and \( \mathbf{B} \in \mathbb{R}^{N \times N} \). By definition [20], a quadratic form \( q \) in the components \( a_0, a_1, ..., a_{N-1} \) of vector \( \mathbf{a} \) is a sum of \( N^2 \)-terms, with \( \mathbf{B} \) the so called coefficient matrix. A quadratic form in the real-valued vector \( \mathbf{x}_\mathbb{R} \) is of course a real valued scalar. When adopting this concept to the augmented complex domain, one can show that it can be represented by a widely quadratic form [2]

\[
q = \mathbf{x}_\mathbb{R}^T \mathbf{M} \mathbf{x}_\mathbb{R} = \frac{1}{2} \left( \mathbf{x}_\mathbb{R}^T \mathbf{T}^H \right) \left( \frac{1}{2} \mathbf{T} \mathbf{M} \mathbf{T}^H \right) \left( \mathbf{T} \mathbf{x}_\mathbb{R} \right)
\]

with \( \mathbf{T} \) as in (2.36).

### 2.3 Widely Linear (WL) Transformations

This section offers a detailed view on widely linear (WL) transformations, which are very important when dealing with complex-valued random signals. To work directly in the complex domain, without performing the intermediate step in the real domain, one must know how linear transformations in \( \mathbb{R} \) can be described in \( \mathbb{C} \). The correct mathematical description is called a WL transformation. A WL transformation follows when a linear transformation is performed on \( \mathbf{x}_\mathbb{R} = [\mathbf{x}_r^T \mathbf{x}_i^T]^T \). For a linear transformations \( f : \mathbb{R}^N \rightarrow \mathbb{R}^M \) the following conditions must be fulfilled:

\[
f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})
\]

\[
f(c \mathbf{a}) = c f(\mathbf{a}),
\]

where \( \mathbf{a}, \mathbf{b} \) are real-valued vectors in \( \mathbb{R}^N \) and \( c \) is a real-valued scalar. A linear transformation can also be expressed by a vector matrix product [20], i.e. \( \mathbf{y}_\mathbb{R} = \mathbf{M} \mathbf{x}_\mathbb{R} \). Thusmathematically expressed this is \( \mathbf{M} : \mathbb{R}^{2N} \) to \( \mathbb{R}^{2M} \). Assume a real-valued matrix \( \mathbf{M} \in \mathbb{R}^{2M \times 2N} \), which is multiplied with real-valued vector \( \mathbf{x}_\mathbb{R} \in \mathbb{R}^{2N} \), then the resultant vector is

\[
\mathbf{y}_\mathbb{R} = \begin{bmatrix} \mathbf{y}_r \\ \mathbf{y}_i \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix} = \mathbf{M} \mathbf{x}_\mathbb{R}.
\]
Rewriting (2.45) yields the two components
\[ y_r = M_{11} x_r + M_{12} x_i, \]
\[ y_i = M_{21} x_r + M_{22} x_i, \]
where \( M_{mn} \in \mathbb{R}^{M \times N} \). \( x_r \) and \( x_i \) influence both, real and imaginary parts of \( y_R \). \( y_R \) is transformed to \( y \) by using (2.36) but with different dimensions. As a result from this, the augmented transformation is given as
\[ y = H x \in \mathbb{C}^{2M} \]

**Proof.**

\[ y = T_M y_R \]
\[ = T_M M x_R \]
\[ = \left( \frac{1}{2} T_M M T_N^H \right) x \]
\[ y = H x \]

\( \square \)

The matrix \( H \) is called an augmented matrix. It satisfies a specific block pattern, where the south-east block is the conjugate of the north-west block and the south-west the conjugate of the north-east, see appendix A.2:
\[ H = \begin{bmatrix} H_1 & H_2 \\ H_2^* & H_1^* \end{bmatrix} \in \mathbb{C}^{2M \times 2N}. \]

All augmented matrices used in this thesis possess this special block pattern. When inserting the linear transformation matrix \( M \) in \( H = \frac{1}{2} T_M M T_N^H \), the block matrices \( H_1 \) and \( H_2 \) are determined by
\[ H_1 = \frac{1}{2} [M_{11} + M_{22} + j(M_{21} - M_{12})] \in \mathbb{C}^{M \times N} \]
\[ H_2 = \frac{1}{2} [M_{11} - M_{22} + j(M_{21} + M_{12})] \in \mathbb{C}^{M \times N}. \]

See appendix A.2 for the detailed derivation. Finally, a WL transformation is defined as

\[ y = H_1 x + H_2 x^*. \]

It is important to notice that \( y \) depends on \( x \) and \( x^* \). A transformation on the real vector \( x_R \) is only linear in \( x \) if and only if the real-valued transformation matrix \( M \) possess the following block pattern
\[ M = \begin{bmatrix} M_{11} & M_{12} \\ -M_{12} & M_{11} \end{bmatrix}. \]

Then \( H_2 = 0 \), and the WL transformation reduces to a *strictly linear* transformation
\[ y = H_1 x. \]

A strictly linear transformation does not depend on the conjugate of \( x \).

Concluding, a linear transformation on the vector \( x_R \) yields a WL transformation on the complex-valued vector \( x \). Only if the real transformation matrix satisfies the specific block pattern (2.53), the transformation is strictly linear on \( x \).
2.4 Complex FIR Filter

In this section, a short review of filters with finite impulse response (FIR) is given. The concept of FIR filters is then altered to complex ones. This case is needed for complex and widely linear Wiener-Filters. An FIR filter can be modeled as a linear time invariant (LTI) system without feedback. The output equation is given by [17]

\[ y[n] = \sum_{k=0}^{N-1} h_k x[n-k]. \]  

(2.55)

where \( x \) is the input signal, \( y \) the output signal, and constants \( h_k \) are the filter coefficients. Equation (2.55) represents the discrete convolution and can also be written as \( y[n] = h_n * x[n] \), which is linear and time invariant. A basic block model of an FIR filter is shown in Figure 2.2. In digital signal processing, it is usual to represent the coefficients of the filter by the discrete time series, \( h[n] \). This is called the impulse response of the filter. The output is given by

\[ y[n] = h[n] * x[n]. \]  

(2.56)

A convolution can also be written in vector notation as

\[ y = Hx, \]  

(2.57)

where \( y \) is the output vector, \( x \) the input vector, and \( H \) the so-called convolution matrix [20]. The filter order is given by the number of the delay elements \( z^{-1} \). When the length of the impulse response is \( N \), then the order is \( (N - 1) \). Using the discrete-time Fourier transformation (DTFT) [17]

\[ X(f) = \sum_{k=-\infty}^{\infty} x[n] \exp(-j2\pi T_s f n), \]  

(2.58)

or short \( X(f) = \text{DTFT}(x[n]) \), the output equation can be described by

\[ Y(f) = H(f) \cdot X(f). \]  

(2.59)

With that, the impulse response can be calculated with \( h[n] = i\text{DTFT}\left(\frac{Y(f)}{X(f)}\right) \), where \( i\text{DTFT}(\cdot) \) is the inverse discrete-time Fourier transform defined by

\[ x[n] = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} X(f) \exp(j2\pi T_s f n) df. \]  

(2.60)

The concept of real FIR filters can be extended to the complex case, where either the input signal and/or the coefficients of the filter are complex.

\[ x[n] = x_r[n] + jx_i[n]; \quad x_r[n], x_i[n] \in \mathbb{R}. \]  

(2.61)

\[ h[n] = h_r[n] + jh_i[n]; \quad h_r[n], h_i[n] \in \mathbb{R}. \]  

(2.62)
The output is still a convolution of the input signal with the impulse response and thus also complex. Therefore, $*$ can also be used for complex signals, but the operation underneath changes. The convolution in the complex case is defined as

\[
y[n] = h[n] * x[n] \\
= \sum_{k=0}^{N-1} (h_r[k] + j h_i[k]) (x_r[n-k] + j x_i[n-k]) \\
= \sum_{k=0}^{N-1} h_r[k] x_r[n-k] - \sum_{k=0}^{N-1} h_i[k] x_i[n-k] + \]

\[
\sum_{k=0}^{N-1} j (h_r[k] x_i[n-k] + h_i[k] x_r[n-k]),
\]

which can be rewritten as

\[
y[n] = h_r[n] * x_r[n] - h_i[n] * x_i[n] + j (h_r[n] * x_i[n] + h_i[n] * x_r[n]) \quad (2.63)
\]

With (2.63), one complex convolution requires four real convolutions. A detailed block diagram of a complex FIR filters is shown in Figure 2.3.

### 2.5 Hilbert Space

This section gives a brief overview of the concepts of the Hilbert space. It extends the known methods from the Euclidean space, which is limited to $\mathbb{R}^3$, to spaces with finite or infinite
dimensions [20]. In digital signal processing, data and signals are very often represented by multidimensional vectors, i.e. \( x \in \mathbb{C}^N \), where usually \( N >> 3 \). Consequently the properties of Hilbert space show to be very useful. Not only data samples, but also random variables can be modeled as vectors. The set of zero-mean SO random variables, where \( E(|x|^2) < \infty \), forms a Hilbert space with respect to the inner product \( <x,y>=E(xy^*) \), where \( E(xy^*) \) is the correlation between \( x \) and \( y \) [2]. With the inner product \( <x,y> \), it is possible to define norm, distance and orthogonality of random variables. The property \( <x,y>=E(xy^*) \) is exploited in Chapter 4.1. The Hilbert space for complex random variables \( x \) and \( y \) implies the following properties

\[
\text{norm: } ||x|| = \sqrt{<x,x>} = \sqrt{E(|x|^2)} = \sqrt{\sigma_x^2} = \sigma_x \tag{2.64}
\]

\[
\text{distance: } ||x-y|| = \sqrt{<x-y,x-y>} = \sqrt{E(|x-y|^2)} \tag{2.65}
\]

\[
\text{angle: } \cos^2(\alpha) = \frac{|<x,y>|^2}{||x||^2 ||y||^2} = \frac{|E(xy^*)|^2}{E(|x|^2)E(|y|^2)} \tag{2.66}
\]

\( x \) and \( y \) are orthogonal if: \( <x,y>=E(xy^*) = 0 \) \( \tag{2.67} \)

Cauchy-Schwarz inequality: \( |<x-y>| \leq ||x|| ||y|| \)
\[
\iff |E(xy^*)| \leq \sqrt{E(|x|^2)E(|y|^2)} \tag{2.68}
\]

The proceeding can easily be generalized for random vectors.

Figure 2.3: Detailed block diagram of a complex FIR filter.
Chapter 3

Complex Statistics

This chapter introduces the reader to the definitions of moments for complex-valued random variables, with special focus on second-order statistics. These statistical properties are needed to derive and evaluate estimators for random variables/signals. A simple example would be to measure the DC voltage of a battery. One would expect, if the manufacturer specifies the battery voltage at 5V, the measurement equipment will show 5 V. This is not the case in reality. There are now two reasons why this is rather impossible. At first, due the variability of the battery components it is impossible to build a battery which possesses exactly a voltage of $5.000000000\ldots$ V. The second reason is the inaccuracy of the measurement equipment. There, noise disturbs the measured signal. As a direct consequence of this, it is not enough to just measure signals. Of course, in this simple case one could round the displayed voltage of 5.015 V to 5 V and live with it. Estimation theory requires knowledge of the statistical properties, such as the mean value or the variance, of signals or the disturbing noise. Thus a random variable $x$ can be described its statistical properties. A complete description of $x$ is given by its pdf or cumulative distribution function (cdf). Apparently, the pdf is not always available or cannot be derived. If not known prior to estimation, one could repeat the experiment and remember the outcome. A far more applied approach is to simulate the underlying process and approximate the pdf. All necessary statistical properties needed for deriving and understanding estimators introduced in this thesis are described in the following sections. The reader may already be familiar with real-valued random variables. The relation to the real case plays a great part to simplify the transition to the complex case. This may help to get a better feeling for complex estimation theory. Reading this chapter also shows that, with some exceptions, conclusions from real-valued random variables can also be used for complex ones.

The chapter first gives a short overview of the basics of probability densities and distributions, which are then altered to the complex domain in Section 3.1. Next, complex mean and variance are introduced in 3.2. Section 3.3 focuses on second-order statistics, which is the main part of this chapter. All necessary characteristics, which are needed to describe a random variable in the real domain, are extended to the complex domain. It is further pointed out why the so called pseudo-characteristics, and hence the augmented representation, are so important in complex estimation. Supplementary the terms proper and improper are explained. These two characteristics have a very high impact on estimators and their performance. In Chapter 4 it is shown why properness is a desired attribute, as it significantly simplifies estimation in the complex domain. The most important properties
of the different covariance matrices are listed in 3.3.1. The complex correlation coefficient in Section 3.3.3 can be used as a measure for the degree of impropri ness. After a brief review of the real-valued Gaussian pdf in Section 3.4, the function is altered to the complex case. Using the augmented representation allows a general mathematical description of the pdf. A significant mathematical simplification is achieved by assumed proper ness. In 3.4.2 the effects of impropri ness on the Gaussian pdf are derived. Some visual examples help the reader to intuitively understand the quite sophisticated mathematical descriptions. The conditional Gaussian pdf, described in Section 3.4.3, is later on required for one particular kind of estimators. At last, Section 3.5 deals with random processes. It is the first section which points out that real world signals can be treated as random processes. With the introduction of time dependencies, the autocorrelation function and the power spectral density (PSD) are welcome tools to work with.

3.1 Complex pdf

A pdf describes the probability density of a random variable over the complete sample space. The sample space \( \Omega \) is the space spanned by all possible outcomes \( x \) of a specific experiment. Its mathematical notation is \( p_X(x) \), where \( 0 \leq p_X(x) < 1 \) for \( -\infty < x < \infty \). Note that the subscript of \( p \) is a capital \( X \). In literature, e.g.: [18], capital letters are used to describe random variable. Whereas lowercase letters describe the realization of the random variable. Until now these two definitions have not been separated. In the next view lines \( X \) marks a random variable and \( x \) its realization. Imagine a simple experiment like throwing a dice. The experiment is described by \( X \) and \( x \) stands for the outcome of one throw. Anyhow, the outcome can only be 1, 2, 3, 4, 5, or 6. It follows that the sample space is completely described by \( \Omega = \{ 1, 2, 3, 4, 5, 6 \} \). It is not possible to get a result like 7 and thus the probability of getting 7 is zero. Mathematically expressed this is \( \Pr(X = 7) = 0 \). One can be completely sure that the outcome is part of the sample space \( \Omega \). Mathematically expressed the probability of getting 1, 2, 3, 4, 5, or 6 is \( \Pr(X = u) = \frac{1}{6}, \forall u \in \Omega \). A dice yields discrete results. In the discrete case, the pdf is called probability mass function (pmf) [18], see Figure 3.1b. For continuous \( x \), an infinite number of outcomes is possible. Thus also the pdf is a continuous function, illustrated by Figure 3.1a. Another way of describing the properties of a random variable is the cdf \( F_X(x) \). It is defined as the area covered by the pdf:

\[
F_X(x) = \Pr(X \leq x) = \int_{-\infty}^{x} p_X(u) \, du \tag{3.1}
\]

Intuitively spoken, the cdf is nothing else but the sum of all probabilities. Integrating over all possible values yields one, \( F_X(\infty) = \int_{-\infty}^{\infty} p_X(x) \, dx = 1 \). The pdf and cdf of an uniform distribution is illustrated in Figure 3.2. Differentiating the cdf yields the pdf

\[
p_X(x) = \frac{dF_X(x)}{dx} \tag{3.2}
\]

The probability of an interval is defined as

\[
\Pr(a \leq X \leq b) = F_X(b) - F_X(a) = \int_{a}^{b} p_X(x) \, dx. \tag{3.3}
\]
Figure 3.1: pdf and pmf of an uniform distribution for continuous and discrete random variables.

In contrast, the probability of getting exactly one value on a continuous random variable is equal to zero.

**Proof.** Let $x$ be a continuous random variable with $\Delta x$ going to zero. Then the probability of getting exactly $x_0$ is

$$
\lim_{\Delta x \to 0} \Pr(x_0 \leq X \leq x_0 + \Delta x) = \lim_{\Delta x \to 0} F_X(x_0 + \Delta x) - F_X(x_0) \\
= F_X(x_0) - F_X(x_0) \\
= 0
$$
A pdf may also be assigned to more random variables. It is then a multidimensional function. This can be visualized by throwing two dices, where the outcome of dice one is marked as random variable $x$ and the second as $y$. With that, the joint pdf is a two-dimensional one and mathematically expressed as $p_{XY}(x, y)$. See Figure 3.3 for a 2-dimensional Gaussian pdf. Note, the semicolon marks that the pdf is now related to $x$ and $y$. However, the covered area is still one, \[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{XY}(x, y) \, dx \, dy = 1. \] A multidimensional pdf can also be described via vector notation. Mathematically, with $z = [x \ y]^T$, the pdf writes as $p_Z(z)$. It is important to know, that if $x$ and $y$ are statistically independent of each other, then the joint pdf is a product of the marginals $p_X(x)$ and $p_Y(y)$, i.e. $p_{XY}(x, y) = p_X(x) p_Y(y)$. From now on the notation of a random variable and its realization will no be distinguished. It should be clear from context if a random variable or its realization is meant. Further, if clear from context, the indices of the pdfs will be dropped for convenience.

![Figure 3.3: MATLAB plot of a 2-dimensional Gaussian pdf.](image)

The pdf can be altered to complex random variables, but it remains real-valued. However, as it is related to a complex random variable the term complex pdf is used in most literature, [19, 28]. A complex-valued random variable is defined as $x = x_r + j x_i$, where $x_r$ and $x_i$ are real-valued random variables. Therefore a complete statistical description requires the joint pdf of $x_r$ and $x_i$. The real vector $x_R = [x_r \ x_i]^T$, see Section 2.2, is assumed to possess a joint pdf $p(x_r, x_i)$. The cdf of the pair $(x_r, x_i)$ is given by $F(x_{r_0}, x_{i_0}) = \Pr (x_r \leq x_{r_0}, x_i \leq x_{i_0})$, which is a real-valued 2-dimensional cdf. The pdf is then the derivative with respect to $x_r$ and $x_i$, $p(x_r, x_i) = \frac{\partial}{\partial x_r} \frac{\partial}{\partial x_i} F(x_r, x_i)$. The complex pdf is defined as

$$p(x) = p(x_r + j x_i) = p(x_r, x_i).$$

(3.4)

It corresponds to the joint pdf of the real-valued random variables [2]. Another used expression is $p(x, x^*)$, which is needed to fully represent real and imaginary part. However, this may also cause confusion. One may interpret $p(x, x^*)$ as the joint pdf of $x$ and its conjugate $x^*$ and hence treat them as independent variables. This is of course not the
case, since \( x^* \) is fully determined by \( x \). As a direct consequence it is not possible that different probability densities are assigned. Only when using the \( \mathbb{C} \mathbb{R} \)-Calculus, \( x^* \) is treated independently of \( x \). A visual example of a complex pdf is illustrated by Figure 3.3, where \( z = x + jy \) and \( p(x, y) \equiv p(z) \).

### 3.2 Complex Moments

This sections introduces the definitions of complex moments. Random variables are completely characterized by their pdf. In practice the complete pdf function is sometimes not known. Therefore it is desirable to summarize some of the key aspects of a density function by using a view numbers instead of the complete function. These numbers are called statistical averages or moments. In this case, complex moments. In the following chapters of this thesis the term complex moments will be reduced to moments. It should be clear from context if the variable is real or complex. For \( \mathbb{R} \), details and examples are described in [18]. As already mentioned, a complex-valued random variable \( x \) consists of two different real-valued random variables. The equivalent real random vector \( \mathbf{x}_\mathbb{R} = [x_r, x_i]^T \) is just a different approach to look on \( x \). Hence, all statistical properties and operations performed on one must also be true for the other. As the reader may know from the real case, the first-order moment is defined as

\[
E(u) = \int_{-\infty}^{\infty} u p(u) \, du,
\]

where \( u \) is a real-valued scalar. Another mathematical notation is \( \mu_u = E(u) \). Other labellings for the first-order moment are mean or expected value. Each possible outcome is multiplied with its probability and added with all other. In the discrete case the integral is replaced by a sum

\[
E(u) = \mu_u = \sum_{n=-\infty}^{\infty} u_n \, p(u_n),
\]

and the pdf becomes a pmf, see Section 3.1. The expectation operator of a function \( g : D \to \mathbb{C}^N \) whose domain \( D \) includes the range of \( x \) is defined as [2]

\[
E [g(x)] = E \{ \Re [g(x)] \} + j \, E \{ \Im [g(x)] \}
\]

\[
= \int_{\mathbb{R}^N} g(x_r + j \, x_i) \, p(x_r + j \, x_i) \, dx_r \, dx_i.
\]

As a consequence the expected value of \( x \) is defined as

\[
E(x) = E(x_r) + j \, E(x_i),
\]

where \( E(\cdot) \) is the expected value with respect to the marginal pdfs \( p(x_r) \) and \( p(x_i) \) of the real-valued random variables. The expected value of a complex random variable is also complex in general. Figure 3.4 shows the simulated mean value of different setups.
Figure 3.4: Mean values (x) of simulated complex-valued random variables, using 100 realizations.

(a) $x \sim \mathcal{N}(0, 1)$, $y \sim \mathcal{N}(0, 1)$, $\rho = 0$

(b) $x \sim \mathcal{N}(2, 0.3^2)$, $y \sim \mathcal{N}(1, 1.2^2)$, $\rho = 0.967$

(c) $x \sim \mathcal{U}(0, 1)$, $y \sim \mathcal{U}(0, 1)$, uncorrelated

(d) $x \sim \mathcal{U}(0, 1)$, $y \sim \mathcal{U}(0, 1)$, correlated
3.3 Second-Order Statistics

The mean value is the expected outcome of the random experiment, but does not hold any information about the distribution. Therefore, higher-order moments are needed. The first one is called second-order (SO) moment and is given by

\[ E(|x|^2) = E(x_r^2) + E(x_i^2). \]  

The SO moment is defined by the expectation of the squared magnitude of the complex-valued random variable. Instead of \( E(|x|^2) \), one could also write \( E(xx^*) \). Compared to the real case, the complex random variable is multiplied with its complex conjugate and not just with itself. The definition of the second order moments varies in literature. For instance in [19], the conjugate is on the first entry, \( E(x^*x) \). In contrast, in [2], the conjugate is on the second, \( E(xx^*) \). In the scalar case, this is irrelevant, but it gets critical when using vectors. In this thesis, the approach of [2] is used further on. This concept is the same for all higher-order moments, which yield real numbers only. (3.11) is the sum of the SO moments of the real and imaginary part.

\[ \text{Proof.} \] Let \( x \) be a scalar complex-valued random variable. Then the SO moment is a real-valued scalar, derived as

\[
E(|x|^2) = E \left( \sqrt{x_r^2 + x_i^2} \right) \\
= E(x_r^2 + x_i^2) \\
= E(x_r^2) + E(x_i^2)
\]

When subtracting the mean, the SO moment becomes the variance

\[
\text{var}(x) = \sigma_x^2 = E \left( |x - E(x)|^2 \right) \tag{3.12}
\]

\[
\downarrow \quad x = x_r + jx_i \\
= E(x_r^2 + x_i^2) - (E(x_r)^2 + E(x_i)^2) \tag{3.13}
\]

\[
\quad = E(|x|^2) + |E(x)|^2 \tag{3.14}
\]

The variance is a real-valued scalar, \( \sigma_x^2 \geq 0 \in \mathbb{R} \). Another denotation is SO central moment. If \( \sigma_x^2 = 0 \), then the variable is not random at all. There would be no spread on the outcomes of the random experiment. An often used characteristic is the standard deviation, which is the square root of the variance \( \sqrt{\text{var}(x)} = \sqrt{\sigma_x^2} = \sigma_x \).

One further important statistical property is the cross-correlation between two (or more) random variables. The complex cross-correlation is defined by

\[
r_{xy} = E(xy^*) \\
= E[(x_r + jx_i)(y_r - jy_i)] \\
= [E(x_r y_r) + E(x_i y_i)] + j[E(x_i y_r) - E(x_r y_i)], \tag{3.15}
\]
which is seen to involve all possible real cross-moments. For further details see [19], but note that Kay uses the conjugate on the first random variable. The covariance as the central moment is defined by

$$\text{cov}(x, y) = E[(x - E(x))(y - E(y))^*], \quad (3.16)$$

which, can be reduced to

$$\text{cov}(x, y) = E(x y^*) - E(x) E(y^*). \quad (3.17)$$

If $x$ is independent of $y$ then $\text{cov}(x, y) = 0$. This is the very same as in the real case. Note that the covariance reduces to the variance if $x = y$, $\text{cov}(x, x) = \text{var}(x)$. However, in the vector case $x \in C^N$, the covariance is a $N \times N$-dimensional matrix. From (3.17), if $E(x) = 0$ or $E(y) = 0$, the covariance is equivalent to the correlation (3.15). Because of the conjugate in (3.16), $\text{cov}(x, y) \neq \text{cov}(x, y)$ [19]. Due to $E(x y^*) = E(x^* y)^*$ the relation $\text{cov}(x, y) = \text{cov}(y, x)^*$ holds for $x \neq y$.

For SO statistics, a closer look on the equivalent real vector $x_R = [x_r, x_i]^T$ is required. As already explained in Section 2.2, it is just a different approach to $x = x_r + jx_i$. Thus all definitions from the real vector must also apply to the complex one. The real covariance matrix $C_{x_R x_R}$ for $x_r, x_i \in \mathbb{R}$ is given by

$$C_{x_R x_R} = E \left[ (x_R - E(x_R))(x_R - E(x_R))^T \right] = \begin{bmatrix} \sigma_{x_r}^2 & \text{cov}(x_r, x_i) \\ \text{cov}(x_i, x_r) & \sigma_{x_i}^2 \end{bmatrix} \in \mathbb{R}^{2 \times 2}. \quad (3.18)$$

The real covariance matrix involves information about the variance of real and imaginary parts and of their covariances. This means that $x_r$ and $x_i$ could be statistically dependent. Imagine an IQ-receiver [17], which converts a received passband (bandpass) signal into an equivalent baseband signal. In theory, there is no interaction between the inphase and quadrature (IQ-) paths, because perfect modulators and filters are assumed. In the real world, the implementation is never perfect and thus at some point, it may happen that parts of the I-path influence the Q-path and vice versa. If that happens, $\text{cov}(x_r, x_i) \neq 0$. Normally, the influence is very small and may be neglected, but for a complete SO description of $x$, these dependencies must be covered. From (2.3) and (2.4) it can be seen that $x$ and $x^*$ are required to get $x_r$ and $x_i$. Consequently, this shall also be used for SO statistics. In Section 2.2 the augmented vector $\tilde{x}$ was introduced, consisting of $x$ and $x^*$. With $\tilde{x}$, it is possible to get a complete SO description in the complex plane by using the so-called augmented covariance matrix

$$C_{\tilde{x} \tilde{x}} = E \left[ (\tilde{x} - E(\tilde{x}))(\tilde{x} - E(\tilde{x}))^H \right] = \begin{bmatrix} \sigma_{x}^2 & \tilde{\sigma}_{x}^2 \\
\tilde{\sigma}_{x}^2 & \sigma_{x}^2 \end{bmatrix} \in \mathbb{C}^{2 \times 2}, \quad (3.19)$$

where $\sigma_{x}^2$ is the complex variance and $\tilde{\sigma}_{x}^2$ the pseudo-variance (pseudo-covariance) [16]. In the field of complex estimation theory, various definitions of $\tilde{\sigma}_{x}^2$ can be found. In some textbooks, e.g.: Schreier [2], the pseudo-variance is denoted as complementary variance or complementary covariance. Others, like Picinbono in [26], name it relation function. This
thesis will go on with the denotation pseudo-variance for scalars and pseudo-covariance for matrices. Note, matrix (3.19) is an augmented matrix, refer to Section 2.3. In the following lines it is shown why the pseudo-variance is an important property. Its definition is
\[ \tilde{\sigma}_x^2 = E \left[ (x - E(x))^2 \right] \in \mathbb{C}. \] \hspace{1cm} (3.20)

In contrast to (3.12) the square and not the squared magnitude is used. Hence, the pseudo-variance is defined by the covariance of \( x \) and its conjugate \( x^* \), i.e. \( \tilde{\sigma}_x^2 = \text{cov}(x, x^*) \). It also considers the correlation between real and imaginary part of \( x \), and it will be immediately be shown that \( \tilde{\sigma}_x^2 = \text{cov}(x, x^*) = \sigma_{x_r}^2 - \sigma_{x_i}^2 + 2j \text{cov}(x_r, x_i) \)

**Proof.** Let \( x \) be a complex random variable defined as \( x = x_r + j x_i \). Then the pseudo-variance derives as
\[ \sigma_x^2 = \text{cov}(x, x^*) = E \left[ (x - E(x)) (x^* - E(x^*))^* \right] \]
\[ = E \left[ (x - E(x))^2 \right] \]
\[ \downarrow x = x_r + j x_i \]
\[ = E \left[ (x_r + j x_i)(x_r + j x_i) \right] - E(x_r + j x_i)E(x_r + j x_i) \]
\[ = E(x_r^2) - E(x_r)^2 - \left[ E(x_i^2) - E(x_i)^2 \right] + 2j \left[ E(x_r x_i) - E(x_r)E(x_i) \right] \]
\[ = \sigma_{x_r}^2 - \sigma_{x_i}^2 + 2j \text{cov}(x_r, x_i) \] \hspace{1cm} (3.21)

The real covariance in (3.21) can also be expressed in dependence of the real correlation coefficient \( \rho_R = \frac{\text{cov}(x_r, x_i)}{\sqrt{\sigma_{x_r}^2 \sigma_{x_i}^2}} \) [18],
\[ \tilde{\sigma}_x^2 = \sigma_{x_r}^2 - \sigma_{x_i}^2 + 2j \rho_R \sigma_{x_r} \sigma_{x_i}. \] \hspace{1cm} (3.22)

This provides a more intuitive view how the correlation between real and imaginary components affect \( \tilde{\sigma}_x^2 \). With (3.12) and (3.22) all properties from (3.18) are covered in the augmented covariance matrix (3.19). This can be shown by defining \( \sigma_{x_r}^2, \sigma_{x_i}^2 \) and \( \text{cov}(x_r, x_i) \) in dependence of \( \sigma_x^2 \) and \( \tilde{\sigma}_x^2 \).
\[ \sigma_{x_r}^2 = \frac{1}{2} \left( \sigma_x^2 + \Re(\tilde{\sigma}_x^2) \right) \]
\[ \sigma_{x_i}^2 = \frac{1}{2} \left( \sigma_x^2 - \Re(\tilde{\sigma}_x^2) \right) \]
\[ \text{cov}(x_r, x_i) = \frac{1}{2} \Im(\tilde{\sigma}_x^2) \]

These properties can be extended to a vector of complex-valued random variables, i.e. \( \mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^T \). Consequently, with \( \bar{x} \) twice the dimension of \( x \), the augmented covariance matrix becomes
\[ \mathbf{C}_{xx} = E \left[ (\mathbf{x} - E(\mathbf{x})) (\mathbf{x} - E(\mathbf{x}))^H \right] \]
\[ = \begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{x\bar{x}} \\ \mathbf{C}_{\bar{x}x} & \mathbf{C}_{\bar{x}\bar{x}} \end{bmatrix} \in \mathbb{C}^{2N \times 2N}. \] \hspace{1cm} (3.23)
\( C_{xx} \) is the Hermitian covariance matrix, defined as
\[
C_{xx} = E \left[ (x - E(x)) (x - E(x))^H \right] \in \mathbb{C}^N. \tag{3.24}
\]
\( \tilde{C}_{xx} \) is the pseudo-covariance matrix and given by
\[
\tilde{C}_{xx} = E \left[ (x - E(x)) (x - E(x))^T \right] \in \mathbb{C}^N. \tag{3.25}
\]
The real covariance matrix becomes
\[
C_{xR}x_{R} = \begin{bmatrix} C_{xx} & C_{xR}x_{i} \\ C_{xR}x_{i}^T & C_{xi}x_{i} \end{bmatrix} \in \mathbb{R}^{2N \times 2N}. \tag{3.26}
\]
If the mean is not subtracted then the matrices are the correlation matrices:
\[
R_{xx} = E(x x^H) \tag{3.27}
\]
\[
R_{xx} = E(x x^T) \tag{3.28}
\]
\[
R_{xx} = E(x x^T) \tag{3.29}
\]
If \( E(x) = 0 \) the covariance matrices are equal to the correlation matrices. Covariance matrices may sometimes be reduced to correlation matrices for convenience. This can simply be done by subtracting the mean before deriving the required result \( x' = x - E(x) \). This has no influence on the SO statistics, but it must be kept in mind that the mean has been subtracted. \( E(x) \) must then be added again to the final solution. The definitions are also naturally extended for cross-covariance and cross-correlation matrices. For two different complex-valued random vectors \( x \) and \( y \), covariance and pseudo-covariance matrices must be considered. In other words, the augmented cross-covariance matrix \( C_{xy} \) is required for a complete SO description in \( \mathbb{C} \).

Conflicts arise, when trying to adopt the properties for real random variables to the complex plane. \( (u, v) \in \mathbb{R}^N \) are called uncorrelated if
\[
C_{uv} = E(u v^T) - E(u) E(v)^T = 0 \tag{3.30}
\]
and orthogonal if
\[
R_{uv} = E(u v^T) = 0. \tag{3.31}
\]
It can be seen that uncorrelatedness is equivalent to orthogonality if at least one random vector has zero-mean. Independent random variables are always uncorrelated, but must not be orthogonal. Without loss of generality, zero-mean random vectors are assumed further on. Thus uncorrelatedness \( \Leftrightarrow \) orthogonality and \( C_{uv} = R_{uv} \). Details can be found in [18]. However, there are different interpretations in literature how these properties are used on complex-valued random variables. One approach is to denote \( x \) uncorrelated to \( y \) if
\[
C_{xy} = E(x y^H) = 0. \tag{3.32}
\]
This condition implies that \( x \perp y \), see Section 2.5. Nevertheless, as mentioned above, to access complete SO description the augmented vectors must be used. So \( x \) and \( y \) are treated as the complex representative of \( x_R = [x_r \ x_i]^T \) and \( y_R = [y_r \ y_i]^T \)
\[
x = x_r + j x_i \Leftrightarrow x = T x_R
\]
\[
y = y_r + j y_i \Leftrightarrow y = T y_R.
\]
Then, to be equivalent to the condition in (3.30), the following must hold

\[
\mathbf{C}_{xy} = \begin{bmatrix} \mathbf{C}_{xy} & \tilde{\mathbf{C}}_{xy} \\ \mathbf{C}^*_{xy} & \mathbf{C}^*_{xy} \end{bmatrix} = \mathbf{0}.
\] (3.33)

Interestingly, this shows that not just the Hermitian covariance matrix, but also the pseudo-covariance one must be zero, \( \mathbf{C}_{xy} = \mathbf{E}(\mathbf{x}\mathbf{y}^H) = \mathbf{C}_{xy} = \mathbf{E}(\mathbf{x}\mathbf{y}^T) = 0 \). It follows that \( \mathbf{x} \perp \mathbf{y} \) and \( \mathbf{x} \perp \mathbf{y}^* \), refer to Section 2.5 for details. Same is true for orthogonality. Two complex-valued random variables are called orthogonal if

\[
\mathbf{R}_{xy} = \begin{bmatrix} \mathbf{R}_{xy} & \tilde{\mathbf{R}}_{xy} \\ \mathbf{R}^*_{xy} & \mathbf{R}^*_{xy} \end{bmatrix} = \mathbf{0}
\] (3.34)

In this thesis, the terms uncorrelatedness and orthogonality are only valid if the structure of the augmented or equivalent real covariance matrices satisfy the specific block pattern of (3.33) and (3.34).

### 3.3.1 Properties of the Covariance Matrices

Here, not only the derivations are shown, but also how the covariance matrices are connected to each other. Assume the \( N \)-dimensional complex-valued random vector \( \mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i \) with its equivalent real vector \( \mathbf{x}_\mathbb{R} = [\mathbf{x}_r^T \mathbf{x}_i^T]^T \) and the augmented vector \( \mathbf{x} = [\mathbf{x}^T \mathbf{x}^H]^T \). The individual block matrices of the real covariance matrix \( \mathbf{C}_{\mathbf{x}_\mathbb{R}\mathbf{x}_\mathbb{R}} \) (3.26) are given by

\[
\mathbf{C}_{\mathbf{x}_r\mathbf{x}_r} = \mathbf{E} \left[ (\mathbf{x}_r - \mathbf{E}(\mathbf{x}_r)) (\mathbf{x}_r - \mathbf{E}(\mathbf{x}_r))^T \right]
\] (3.35)

\[
\mathbf{C}_{\mathbf{x}_i\mathbf{x}_i} = \mathbf{E} \left[ (\mathbf{x}_i - \mathbf{E}(\mathbf{x}_i)) (\mathbf{x}_i - \mathbf{E}(\mathbf{x}_i))^T \right]
\] (3.36)

\[
\mathbf{C}_{\mathbf{x}_r\mathbf{x}_i} = \mathbf{E} \left[ (\mathbf{x}_r - \mathbf{E}(\mathbf{x}_r)) (\mathbf{x}_i - \mathbf{E}(\mathbf{x}_i))^T \right] = \mathbf{C}_{\mathbf{x}_i\mathbf{x}_r}^T
\] (3.37)

For real-valued random variables, \( \text{cov}(\mathbf{x}_r, \mathbf{x}_i) = \text{cov}(\mathbf{x}_i, \mathbf{x}_r) \) holds [18]. Thus \( \mathbf{C}_{\mathbf{x}_\mathbb{R}\mathbf{x}_\mathbb{R}} = \mathbf{C}_{\mathbf{x}_\mathbb{R}\mathbf{x}_\mathbb{R}}^T \) and it follows that \( \mathbf{C}_{\mathbf{x}_r\mathbf{x}_i} = \mathbf{C}_{\mathbf{x}_i\mathbf{x}_r}^T \). \( \mathbf{C}_{\mathbf{x}_r\mathbf{x}_r} \) and \( \mathbf{C}_{\mathbf{x}_i\mathbf{x}_i} \) are symmetric, too. \( \mathbf{C}_{\mathbf{x}_r\mathbf{x}_i} \) and \( \mathbf{C}_{\mathbf{x}_i\mathbf{x}_r} \) are skew-symmetric. Beside, the real covariance matrix is positive semi-definite, i.e. \( \mathbf{a}^T \mathbf{C}_{\mathbf{x}_\mathbb{R}\mathbf{x}_\mathbb{R}} \mathbf{a} \geq 0 \), \( \mathbf{a} \in \mathbb{R}^{2N} \).

The Hermitian covariance matrix (3.24) is derived as follows

\[
\mathbf{C}_{\mathbf{x}\mathbf{x}} = \mathbf{E} \left[ (\mathbf{x} - \mathbf{E}(\mathbf{x})) (\mathbf{x} - \mathbf{E}(\mathbf{x}))^H \right]
\]

\[
= \mathbf{E} \begin{bmatrix} x_1 - E(x_1) \\ x_2 - E(x_2) \\ \vdots \\ x_N - E(x_N) \end{bmatrix} \begin{bmatrix} x_1^* - E(x_1)^* & x_2^* - E(x_2)^* & \cdots & x_N^* - E(x_N)^* \end{bmatrix}^T
\]

\[
= \begin{bmatrix} \text{var}(x_1) & \text{cov}(x_1, x_2) & \cdots & \text{cov}(x_1, x_N) \\ \text{cov}(x_2, x_1) & \text{var}(x_2) & \cdots & \text{cov}(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(x_N, x_1) & \text{cov}(x_N, x_2) & \cdots & \text{var}(x_N) \end{bmatrix}
\] (3.38)
In contrast, in [19] an additional conjugate $C_{xx}^{Kay} = [\cdot]^*$ is used to achieve the same form as in (3.38). This is because of the different definition of the covariances. Recall, in [19] the conjugate is on the first entry in the covariance definition. Whereas in the vector case, the covariance matrix is defined equivalent to (3.24), where the $(\cdot)^H$ is on the latter vector. A serious weakness with the argument is visible if using the covariance matrix definition and reducing the vector to a scalar. Then $C_{xx} = E[(x - E(x))(x - E(x))^*]$ is different from $\text{cov}(x, x) = E[(x - E(x))(x - E(x))^*]$, while meaning the very same. So, though $C_{xx}^{Kay}$ uses the very same definition as in (3.24), without the additional conjugate the covariance matrix would be the transpose of (3.38):

$$
C_{xx}^{Kay} = \begin{bmatrix}
\text{var}(x_1) & \text{cov}(x_2, x_1) & \cdots & \text{cov}(x_N, x_1) \\
\text{cov}(x_1, x_2) & \text{var}(x_2) & \cdots & \text{cov}(x_N, x_2) \\
\vdots & \vdots & \ddots & \vdots \\
\text{cov}(x_1, x_N) & \text{cov}(x_2, x_N) & \cdots & \text{var}(x_N)
\end{bmatrix} = C_{xx}^T \tag{3.39}
$$

The reason why $C_{xx}^{Kay*} = C_{xx}$ is because of $\text{cov}(x_n, x_m) = \text{cov}(x_m, x_n)^*$. This follows from the properties of the expectation operator $E(\cdot)$. A further property is the Hermitian form of (3.24), $C_{xx} = C_{xx}^H$. The entries in the main diagonal are real-valued, and the off-diagonal elements are conjugates of each other. That is why it is called Hermitian covariance matrix [2]. Similar to the real covariance matrix, the Hermitian one can be shown to be positive semi-definite, $a^H C_{xx} a \geq 0$ [19].

However, the property $\text{cov}(x, y^*) = \text{cov}(y, x^*)$ holds for the pseudo-covariance.

**Proof.** Let $x$ and $y$ be complex random scalars with $\text{cov}(x, y^*) = E[(x - E(x))(y - E(y))]$. Then the pseudo-covariance of the reverse entries is

$$
\text{cov}(y, x^*) = E[(y - E(y))(x^* - E(x)^*)] = E[(y - E(y))(x - E(x))] \nonumber \\
\downarrow \nonumber \\
x, y \in \mathbb{C}^1 \\
= E[(x - E(x))(y - E(y))] = \text{cov}(x, y^*)
$$

It follows that the pseudo-covariance matrix possesses a symmetric form $\tilde{C}_{xx} = \tilde{C}_{xx}^T$. The entries in its main diagonal are the pseudo-variances and thus generally complex.

The augmented covariance matrix (3.23) is derived by using the augmented vector. When inserting $T x_R$ instead of $x$ the derivation shows

$$
C_{xx} = E\left[(x - E(x))(x - E(x))^H\right] = E\left[(T x_R - E(T x_R))(T x_R - E(T x_R))^H\right] = E\left[(T x_R - T E(x_R))(x_R^H T^H - E(x_R)^H T^H)\right] = T E \left[(x_R - E(x_R))(x_R - E(x_R))^T\right] T^H = T C_{x_R x_R} T^H \tag{3.40}
$$
T is as in (2.36) and $x_R^H = x_R^T$ holds as $x_R \in \mathbb{R}^{2N}$. With $T^{-1} = \frac{1}{2}T^H$ (2.37), the real covariance matrix is

$$C_{x_R x_R} = \frac{1}{4}T^H C_{xx} T$$

(3.41)

Further, the inverse of $C_{x_R x_R}$ can be simply calculated by

$$C^{-1}_{x_R x_R} = \left(\frac{1}{4}T^H C_{xx} T\right)^{-1}$$

$$= 4T^{-1} C^{-1}_{xx} T^{-H}$$

$$= 4 \frac{1}{2}T^H C^{-1}_{xx} \frac{1}{2}T$$

$$= T^H C^{-1}_{xx} T,$$

(3.42)

The relation of the determinants is [2]

$$\det (C_{xx}) = 2^{2N} \det (C_{x_R x_R}),$$

(3.43)

where $N$ is the dimension of $x$. From (3.40) the Hermitian and pseudo-covariance matrix are

$$C_{xx} = C_{x_r, x_r} + C_{x_i, x_i} + j \left(C_{x_r, x_i}^T - C_{x_r, x_i}\right)$$

(3.44)

$$\tilde{C}_{xx} = C_{x_r, x_r} - C_{x_i, x_i} + j \left(C_{x_r, x_i}^T + C_{x_r, x_i}\right)$$

(3.45)

From the Hermitian form of $C_{xx}$ and the symmetric form of $\tilde{C}_{xx}$ it follows that the augmented covariance matrix is a Hermitian matrix $C_{xx} = C_{xx}^H$.

### 3.3.2 Properness and Circularity

This section covers the conditions and properties of proper and circular complex-valued random variables/vectors. Properness is a constraint, which simplifies the closed form solution of estimators to a very similar structure than in the real case. Further, it provides a simple way to attain $C_{xx}$ from $C_{x_R x_R}$. Another characteristic is called circularity, which introduces even stronger constraints on the statistics of $x$.

A complex-valued random variable is called proper if and only if its pseudo-variance is zero, $\tilde{\sigma}_x^2 = 0$. This results in:

$$\sigma_{x_r}^2 = \sigma_{x_i}^2$$

(3.46)

$$\text{cov}(x_r, x_i) = 0.$$  

(3.47)

In words, a complex-valued random variable $x$ is called proper if and only if real and imaginary parts possess the same variance and are uncorrelated. In contrast to other definitions, which mean exactly the same thing, it may be the most intuitive one. Properness may also be defined if $x$ is uncorrelated to its conjugate $x^*$ [28]. Although this definitions may sound different, but when looking on the mathematical definition (3.20) it is the very same. Similar, a complex-valued random vector $x = [x_1 \ x_2 \ \cdots \ x_N]^T$ with $x_n = x_{rn} + jx_{in}$ is
called proper if the pseudo-covariance matrix vanishes, \( \tilde{C}_{xx} = 0 \). Therefore the equivalent real-valued vector \( x_R = [x_r^T \ x_i^T]^T \) must fulfill following conditions:

\[
\begin{align*}
\text{cov}(x_r^n, x_r^m) &= \text{cov}(x_i^n, x_i^m) \quad (3.48) \\
\text{cov}(x_r^n, x_i^m) &= -\text{cov}(x_i^n, x_r^m). \quad (3.49)
\end{align*}
\]

The covariance between the real parts of \( x_n \) and \( x_m \) must be the same as between the imaginary parts (3.48). Further it is required that the covariance between the real part of \( x_n \) and imaginary part of \( x_m \) is the negative as between the imaginary part of \( x_n \) and the real part of \( x_m \) (3.49). For the real covariance matrices it follows

\[
\begin{align*}
C_{x_r x_r} &= C_{x_i x_i} \quad (3.50) \\
C_{x_r x_i} &= -C_{x_i x_r} \quad (3.51)
\end{align*}
\]

With that, the equivalent real covariance matrix for proper constellations is given by

\[
C_{x_R x_R} = \begin{bmatrix} C_{x_r x_r} & C_{x_r x_i} \\ -C_{x_r x_i} & C_{x_i x_r} \end{bmatrix}
\]

(3.52)

Looking at (3.45) it becomes clear, why the pseudo-covariance matrix vanishes if (3.50) and (3.51) hold. If \( x \) is proper, then the augmented covariance matrix is a block-diagonal one

\[
C_{xx} = \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{xx}^{*} \end{bmatrix}
\]

(3.53)

As a result the determinant and the inverse of \( C_{xx} \) simply becomes

\[
\det (C_{xx}) = |\det (C_{xx})|^2 \]

(3.54)

\[
C_{xx}^{-1} = \begin{bmatrix} C_{xx}^{-1} & 0 \\ 0 & C_{xx}^{-*} \end{bmatrix}
\]

(3.55)

However, the complex covariance matrix (3.38) reduces to

\[
C_{xx} = 2 \left( C_{x_r x_r} + j C_{x_r x_i}^T \right) \]

(3.56)

If (3.46) and (3.47), or (3.50) and (3.51), respectively, are violated, then \( x/x \) is called improper. Improper random variables/vectors do not imply any limitations on covariance matrices or on the form of the pdf. This lack of constraints greatly increases the calculation effort for estimators. Unfortunately, most real world signals are improper [2]. The task rises to choose between the lower computation effort of proper estimators or the better results of the improper ones. Detailed results are shown in Section 4.1.3.

As already mentioned, an even stronger limitation on the characteristics of \( x \) is the circularity condition. A complex-valued random vector/variable is called circular if and only if its pdf is rotationally invariant [28], i.e. if \( x' = x \ exp (j \ \alpha) \) have the same probability distribution for any given real \( \alpha \). Therefore, \( E(x) = 0 \) in order that \( x \) can be circular. Non centralized pdfs are not rotationally invariant. Figure 3.5a shows the contour of a circular complex Gaussian distributed random variable. As one can see, rotation does not have any influence on the form of the pdf. Whereas in Figure 3.5b, the contour still has a circle shape, but changes when rotated. Circularity does not imply any conditions on the Hermitian covariance matrix.
CHAPTER 3. COMPLEX STATISTICS

Proof.

\[ C'_{xx} = E(x'x'^H) \\
= E(x \exp(j\alpha)x^H \exp(-j\alpha)) \\
= E(xx^H \exp(+j\alpha - j\alpha)) \\
= E(xx^H) \\
= C_{xx} \]

However, a necessary condition for circularity is \( \tilde{C}_{xx} = 0 \).

Proof.

\[ \tilde{C}'_{xx} = E(x'x'^T) \\
= E(x \exp(j\alpha)x^T \exp(j\alpha)) \\
= E(xx^H) \\
= E(xx^H) \exp(+2j\alpha) \]

\( \tilde{C}_{xx} = \tilde{C}'_{xx} \) can be true for arbitrary \( \alpha \) only if \( \tilde{C}_{xx} = 0 \).

To conclude, a circular complex random variable/vector is also proper. This is not generally true the other way. Hence, a proper random variable must not be circular. The relation is a result of different constraints. While the conditions for properness are only limited to SO statistics, circularity requires that the distribution, and thus all moments are rotationally invariant. Figure 3.5 shows contours of improper, proper, and circular complex Gaussian pdfs. Chapter 3.1 provides deeper insights in the complex Gaussian distribution. In the field of complex estimation, researchers use the terms proper and circular interchangeably. While Schreier [2] denotes it as proper, others like Picinbono [26] call it circular. Proper may also be denoted as second-order circular. In this thesis the term proper is used if \( \tilde{C}_{xx} = 0 \) and circular if the pdf is rotationally invariant.

Now the question may arise where this is useful in real world. A good example are modulation techniques, like QAM, BPSK, or PSK. Consequently as the data signal is the outcome of a random process, the modulated symbols are then random, too. In receivers, like for ODFM systems, algorithms are used to estimate the transmitted data out of the distorted received signal. But not the data itself, it is the modulated signal, which is estimated. However, some types of these receivers use Wiener-Filters [19]. These algorithms require prior knowledge of the first and SO statistics of the transmitted signal. The modulated signal is transmitted and estimated. Thus the statistics required are related to the modulation scheme, which is treated as the pmf [18]. Like for the complex Gaussian random variable in Figure 3.5 the symbol distribution may be characterized as proper, improper, or circular. Figure 3.6 shows the scatter plot of four typical modulation techniques. For BPSK 3.6a and ASK 3.6b the quadrature part (imaginary part) is zero and hence \( \sigma_{x_i} = 0 \). With that, the condition (3.48) is violated. A signal of BPSK modulated symbols is always improper. Thus correct estimation requires full SO statistics. The QAM
(a) Circular and proper.

(b) Non centralized. Proper but not circular.

(c) Improper, real and imaginary parts possess different variances.

(d) Improper, real and imaginary parts are correlated.

Figure 3.5: Results of properness, circularity, and improperness on the contour of a complex distributed Gaussian random variable $x$. 
scatter plot 3.6d shows that the real and imaginary part are equally distributed. So (3.48) and (3.49) are fulfilled and thus a QAM modulated signal is always proper. Further, also PSK modulations are proper, see Figure 3.6c. The scatterplot may look circular, but with the condition of circularity the pmf is circular only if it is rotationally invariant for any given real \( \alpha \). Thus if \( \alpha \neq k \frac{2\pi}{16} \) for \( k = 1, 2, 3, \ldots \) the pmf is not circular. Chapter 5 shows simulation results for ASK modulated signals in OFDM systems.

Anyhow, it is important to know that propriety is preserved by strictly linear transformations [2]. A linear transformation is called strictly linear if and only if \( \mathbf{H}_2 = 0 \), i.e. \( y = \mathbf{H}_1 \mathbf{x} \). Thus, in the real case \( \mathbf{y}_R = \mathbf{M} \mathbf{x}_R \), \( \mathbf{M} \) must be given as in (2.53). For \( \mathbf{y}_R \) the conditions of (3.50) and (3.51) must hold. Therefore, \( \mathbf{y} \) possess the same form as in (3.52).

**Proof.** The proof is restricted to the scalar case. Assume \( x, y \) are complex scalars and thus \( x_R, y_R \in \mathbb{R}^2 \). Further, the mean is given by \( E(x_R) = [E(x_r) \ E(x_i)]^T \) and the real covariance matrix is as in (3.52):

\[
\mathbf{C}_{x_Rx_R} = \begin{bmatrix}
\sigma^2 & 0 \\
0 & \sigma^2
\end{bmatrix}.
\]

After the linear transformation the mean is

\[
E(y_R) = \mathbf{M} E(x_R) = \begin{bmatrix} M_2 & -M_1 \\
-M_2 & M_1 \end{bmatrix} \begin{bmatrix} x_r \\
x_i \end{bmatrix} = \begin{bmatrix} M_1 E(x_r) + M_2 E(x_i) \\
M_1 E(x_i) - M_2 E(x_r) \end{bmatrix}.
\]

And the covariance matrix becomes

\[
\mathbf{C}_{y_Ry_R} = E \left( (y_R - E(y_R)) (y_R - E(y_R))^T \right) = M \mathbf{E} \left( (x_R - E(x_R)) (x_R - E(x_R))^T \right) \mathbf{M}^T
\]

\[
= \mathbf{M} \begin{bmatrix}
\sigma_1^2 & 0 \\
0 & \sigma_1^2
\end{bmatrix} \begin{bmatrix} M_1 & M_2 \\
-M_2 & M_1 \end{bmatrix} = \sigma_1^2 \begin{bmatrix}
M_1 & M_2 \\
-M_2 & M_1
\end{bmatrix} \begin{bmatrix} M_1 & M_2 \\
-M_2 & M_1 \end{bmatrix}
\]

\[
= \sigma_1^2 \left( M_1^2 + M_2^2 \right) \begin{bmatrix} M_1 & M_2 \\
-M_2 & M_1 \end{bmatrix} \left( M_1^2 + M_2^2 \right) = 0
\]

\[
\mathbf{M} \mathbf{C}_{x_Rx_R} \mathbf{M}^T
\]

\[
= \sigma_1^2 \left[ M_1^2 + M_2^2 \right] \begin{bmatrix} M_1^2 + M_2^2 & -M_1 M_2 + M_2 M_1 \\
-M_2 M_1 + M_1 M_2 & M_1^2 + M_2^2 \end{bmatrix}
\]

\[
= \sigma_1^2 \left[ M_1^2 + M_2^2 \right] \begin{bmatrix} M_1^2 + M_2^2 & 0 \\
0 & M_1^2 + M_2^2 \end{bmatrix}
\]

\[
\square
\]
Figure 3.6: Modulation schemes for digital signal processing applications.
The real covariance matrix $C_{yRyR}$ possesses a diagonal form, where the north-west entry is the same as the south-east entry. So $\text{var}(y_r) = \text{var}(y_i) = \sigma^2 (M_1^2 + M_2^2)$ and $\text{cov}(y_r, y_i) = 0$. Conditions (3.46) and (3.47) are fulfilled and thus $y$ is proper.

### 3.3.3 Complex Correlation Coefficient

Like in the real case, there is also a correlation coefficient for complex-valued random variables. In $\mathbb{R}$, it describes the degree of correlation between two random variables [18]. The complex correlation coefficient is used as a measure for the degree of impropriety of $x$. It is defined as [28]

$$\rho = \frac{\sigma_x^2}{\sigma_x^2} = \frac{\text{cov}(x, x^*)}{\text{cov}(x, x)}$$

(3.57)

$\rho$ is a complex-valued scalar, because $\sigma_x^2 \in \mathbb{C}$. Thus it may be rewritten as

$$\rho = |\rho| \exp\{j \theta\}.$$  

(3.58)

Note that $0 \leq |\rho| \leq 1$. Another perspective on the complex correlation coefficient is given by using the real-valued random variables $x_r$ and $x_i$. If inserting these into (3.57) yields

$$\rho = \frac{\sigma_{x_r}^2 - \sigma_{x_i}^2 + 2j \text{cov}(x_r, x_i)}{\sigma_{x_r}^2 + \sigma_{x_i}^2}$$

$$\downarrow \quad \text{cov}(x_r, x_i) = \rho_{\Re} \sigma_{x_r} \sigma_{x_i}$$

$$= \frac{\sigma_{x_r}^2 - \sigma_{x_i}^2}{\sigma_{x_r}^2 + \sigma_{x_i}^2} + 2j \frac{\rho_{\Re} \sigma_{x_r} \sigma_{x_i}}{\sigma_{x_r}^2 + \sigma_{x_i}^2}$$

(3.59)

This form provides a more intuitive view on $\rho$. It makes clear that it depends on the correlation between real and imaginary part. From (3.59), one gets

$$\sigma_{x_r}^2 = \frac{\sigma_x^2 \left(1 + \Re\{\rho\}\right)}{2}$$

(3.60)

$$\sigma_{x_i}^2 = \frac{\sigma_x^2 \left(1 - \Im\{\rho\}\right)}{2}$$

(3.61)

$$\rho_{\Re} = \frac{\Im\{\rho\}}{\sqrt{1 - \Re\{\rho\}}}$$

(3.62)

As $\rho$ is a measure for impropriety, it becomes zero if $x$ is proper.

### 3.4 Complex Gaussian pdf

In this section the likely most important pdf in the field of signal processing is introduced. The Gaussian pdf is altered from the real to the general complex Gaussian pdf. Further, the special case of a proper distribution is shown. The pdf of a scalar real-valued Gaussian distributed random variable is defined as

$$p(u) = \frac{1}{\sqrt{2\pi\sigma_u^2}} \exp\left[ -\frac{1}{2} \left( \frac{u - E(u)}{\sigma_u} \right)^2 \right],$$

(3.63)
where \( E(u) = \mu_u \) is the expected value, or mean, and \( \sigma_u \) the standard deviation. Another synonym for a Gaussian distributed random variable is Normal distribution. It is denoted as \( u \sim \mathcal{N}(\mu_u, \sigma_u^2) \). Before extending to the general vector case, the bivariate Gaussian distribution is given for \( u \sim \mathcal{N}(\mu_u, \sigma_u^2) \) and \( v \sim \mathcal{N}(\mu_v, \sigma_v^2) \). Assume that \( u \) and \( v \) are independent of each other. The joint pdf is then a product of the marginal pdfs

\[
P(u, v) = \frac{1}{2\pi \sigma_u \sigma_v} \exp \left\{ -\frac{1}{2} \left[ \left( \frac{u - \mu_u}{\sigma_u} \right)^2 + \left( \frac{v - \mu_v}{\sigma_v} \right)^2 \right] \right\}.
\] (3.64)

If \( u \) and \( v \) possess the same variance \( \sigma_u^2 = \sigma_v^2 = \sigma^2 \), the bivariate pdf reduces to

\[
P(u, v) = \frac{1}{\pi \sigma^2} \exp \left\{ -\frac{1}{\sigma^2} (u - \mu_u)^2 + (v - \mu_v)^2 \right\}.
\] (3.65)

To extend this to the vector case (multivariate pdf), the random vector \( u = [u_1 \ u_2 \ \cdots \ u_N]^T \) with \( N \) independent random variables is used. Each entry is distributed as \( u_i \sim \mathcal{N}(\mu_{u_i}, \sigma_{u_i}^2) \). The joint pdf is \( p(u) = \prod_{i=1}^N p(u_i) \) and is denoted by \( u \sim \mathcal{N}(\mu_u, C_{uu}) \). \( C_{uu} \) is diagonal with the variances of each \( u_i \) in its main diagonal. The general Gaussian pdf for the vector case is given by

\[
P(u) = \frac{1}{(2\pi)^{N/2} \det^{1/2} C_{uu}} \exp \left\{ -\frac{1}{2} (u - \mu_u)^T C_{uu}^{-1} (u - \mu_u) \right\}.
\] (3.66)

It is important to know that (3.66) can also be used if the entries of \( u \) are not independent of each other. All relations (covariances) are covered by the covariance matrix \( C_{uu} \).

With the above, it is possible to alter the definitions from the real to the complex case, where the random vector is complex-valued \( x = x_r + j x_i \in \mathbb{C}^N \). As described in Section 2.2, the complex representation is just another way to describe the real vector \( x_r = [x_r^T \ x_i^T]^T \in \mathbb{R}^{2N} \). To describe the pdf of \( x \), it requires knowledge of the joint pdf of the real vector \( p(x_r) = p(x_r, x_i) \). The real vector \( x_r \) in (3.66) yields

\[
P(x_r) = \frac{1}{(2\pi)^{N/2} \det^{1/2} (C_{xxR})} \exp \left\{ -\frac{1}{2} (x_r - \mu_{x_r})^T C_{xxR}^{-1} (x_r - \mu_{x_r}) \right\}.
\] (3.67)

The first step to alter the pdf to the complex domain is to use the augmented vector \( \mathbf{x} = \mathbf{T} x_r \in \mathbb{C}_2^N \). For that, the properties of (3.41), (3.42), and (3.43) are implemented in (3.67) to replace the real covariance matrix by the augmented one. The pdf becomes

\[
P(x_r) = \frac{1}{(2\pi)^{N/2} \det^{1/2} (C_{xx})} \exp \left\{ -\frac{1}{2} (x_r - \mu_{x_r})^H C_{xx}^{-1} (x_r - \mu_{x_r}) \right\}.
\] (3.68)

With this, the general pdf of a complex-valued Gaussian distributed random vector \( x \) is

\[
P(x) = \frac{1}{\pi^N \det^{1/2} (C_{xx})} \exp \left\{ -\frac{1}{2} (x - \mu)^H C_{xx}^{-1} (x - \mu) \right\}.
\] (3.69)
(3.69) depends on \(x\) and its conjugate \(x^*\). Anyway, it is interpreted or corresponds to the joint pdf of \(x_r\) and \(x_i\). In literature, a general complex-valued Gaussian distributed is denoted as \(x \sim CN(\mu_x, C_{xx}, \tilde{C}_{xx})\).

It is important to notice that the general complex Gaussian pdf can be used for proper and improper variables/vectors. For the special case of properness it has already been shown in Section 3.3.2 that the pseudo-covariance matrix vanishes and the augmented one reduces to a block diagonal matrix. The determinant of the block diagonal matrix is (3.54). When investigating on the quadratic form of \(q = x^H C_{xx}^{-1} x\), it shows that for the proper case it reduces to

\[
x^H C_{xx}^{-1} x = 2 x^H C_{xx}^{-1} x
\]  

(3.70)

Proof.

\[
x^H C_{xx}^{-1} x = \begin{bmatrix} x^H \\ x^T \end{bmatrix} \begin{bmatrix} C_{xx}^{-1} & 0 \\ 0 & C_{xx}^{-1} \end{bmatrix} \begin{bmatrix} x \\ x^* \end{bmatrix} = x^H C_{xx}^{-1} x + x^T C_{xx}^{-1} x^* = x^H C_{xx}^{-1} x + (x^H C_{xx}^{-1} x)^*
\]

With \(x^H C_{xx}^{-1} x \in \mathbb{R}\) the quadratic form reduces to

\[
x^H C_{xx}^{-1} x = 2 x^H C_{xx}^{-1} x
\]

Finally, using the relations for the determinant (3.54) and the quadratic form (3.70) the pdf for a proper Gaussian distributed complex-valued random vector is given by

\[
p(x) = \frac{1}{\pi^N \det(C_{xx})} \exp \left[ -(x - \mu_x)^H C_{xx}^{-1} (x - \mu_x) \right] \]  

(3.71)

In the proper case, the notation \(x \sim CN(\mu_x, C_{xx})\) is used. This already indicates that the pseudo-covariance matrix is zero. Actually, (3.71) is also valid for non diagonal complex covariance matrices and thus for correlated vectors. The derived form is similar to the common real multivariate Gaussian pdf. In the real case \(2N \times 1\) real vectors and \(2N \times 2N\) real matrices are needed. In contrast, the complex case uses \(N \times 1\) complex vectors and \(N \times N\) complex matrices.

### 3.4.1 Important Properties of the Proper Complex Multivariate Gaussian pdf

Here, a comprehensive overview of the most important properties of the complex multivariate Gaussian pdf is shown. For detailed derivations and further information see [19].

1. Any subvector of a proper complex Gaussian random vector is also proper complex Gaussian, thus the marginal pdfs are complex Gaussian.
2. If the vector \( \mathbf{x} \sim \mathcal{CN}(\mu_x, \mathbf{C}_{xx}) \) and all its elements \( \{x_1, x_2, \ldots, x_N\} \) are uncorrelated, then they are also independent.

3. If \( \{x_1, x_2, \ldots, x_N\} \) are independent, and each element \( x_i \sim \mathcal{CN}(\mu_{x_i}, \sigma_i^2) \), then the vector \( \mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^T \) is also proper complex Gaussian.

4. Affine (linear) transformations of proper complex Gaussian random vectors are again proper complex Gaussian. The mean and the covariance are scaled versions of the original ones. If we assume the affine linear transformation \( \mathbf{y} = A\mathbf{x} + \mathbf{b} \), where \( A \) is a complex \( M \times N \) matrix with \( M \leq N \) and full rank, \( \mathbf{b} \) is a complex \( M \times 1 \) vector and \( \mathbf{x} \sim \mathcal{CN}(\mu_x, \mathbf{C}_{xx}) \), then \( \mathbf{y} \) is distributed as

\[
\mathbf{y} \sim \mathcal{CN}(A\mu_x + \mathbf{b}, \underbrace{AC_{xx}A^H}_{\mu_y}, \underbrace{C_{yy}}_{\sigma_y^2})
\] (3.72)

As already shown in Section 3.3.2, a strictly linear transformation preserves propriety.

5. The sum of independent proper complex Gaussian random variables is also proper complex Gaussian distributed.

6. Assume \( [\mathbf{x}^T \ \mathbf{y}^T]^T \) is a complex Gaussian random vector distributed as

\[
\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{CN}
\begin{bmatrix} E(\mathbf{x}) \\ E(\mathbf{y}) \end{bmatrix},
\begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix}
\].

Then the conditional pdf \( p(\mathbf{y}|\mathbf{x}) \) is also proper complex Gaussian with

\[
E(\mathbf{y}|\mathbf{x}) = E(\mathbf{y}) + \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1} (\mathbf{x} - E(\mathbf{x}))
\]

\[
\mathbf{C}_{y|x} = \mathbf{C}_{yy} - \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1} \mathbf{C}_{xy}
\] (3.73)

which has the identical form as in the real case.

7. If \( \mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T \sim \mathcal{CN}(0, \mathbf{C}_{xx}) \), then

\[
E(x_1^* x_2 x_3^* x_4) = E(x_1^* x_2) E(x_3^* x_4) + E(x_1^* x_4) E(x_2 x_3^*)
\]

Note that \( E(x_1^* x_3^*) = E(x_2 x_4) = 0 \) and hence the form is similar to the real case. This result directly follows from the constraints to the covariance defined in (3.16) and it holds for proper signals only.

### 3.4.2 Effects of Improperness on Complex Gaussian pdf

Here, a closer look is taken on the pdf of a scalar complex-valued Gaussian random variable, i.e. \( x \in \mathbb{C} \) and \( \mathbf{x} \in \mathbb{C}^2 \). Without loss of generality, zero mean is assumed. From \( E(x) = 0 \) it follows \( E(\mathbf{x}) = \mu = [0 \ 0]^T \). The pdf is given by

\[
p(x) = \frac{1}{\pi \det \frac{1}{2} \mathbf{C}_{xx}} \exp \left[-\frac{1}{2} \mathbf{x}^H \mathbf{C}_{xx}^{-1} \mathbf{x}\right],
\] (3.74)
With $x$ being a scalar, the augmented covariance matrix becomes a $2 \times 2$-dimensional one
\[
\mathbf{C}_{xx} = \begin{bmatrix} \sigma^2 & \bar{\sigma}^2 \\ \bar{\sigma}^2 & \bar{\sigma}^2 \end{bmatrix},
\]
(3.75)

with variance $\sigma^2 = \text{var}(x) = E(xx^*)$ and pseudo-variance $\bar{\sigma}^2 = E(xx)$. It is obvious that the pdf in (3.74) is related to the variance and pseudo-variance. Further conclusions and interesting results on the Gaussian pdf can be seen by using the complex correlation coefficient $\rho$, introduced in Section 3.3.3. It provides an intuitive view on the impact of improperness on the Gaussian pdf. (3.75) becomes
\[
\mathbf{C}_{xx} = \begin{bmatrix} \sigma^2 & \rho \bar{\sigma}^2 \\ \rho^* \bar{\sigma}^2 & \bar{\sigma}^2 \end{bmatrix}.
\]
(3.76)

With that, $\det(\mathbf{C}_{xx}) = (\sigma^2)^2 \left( 1 - |\rho|^2 \right)$. The inverse follows
\[
\mathbf{C}_{xx}^{-1} = \begin{bmatrix} \sigma^2 & \rho \bar{\sigma}^2 \\ \rho^* \bar{\sigma}^2 & \bar{\sigma}^2 \end{bmatrix}^{-1} = \frac{1}{(\sigma^2)^2 (1 - |\rho|^2)} \begin{bmatrix} \sigma^2 & -\rho \bar{\sigma}^2 \\ -\rho^* \bar{\sigma}^2 & \bar{\sigma}^2 \end{bmatrix} = \frac{1}{\sigma^2 (1 - |\rho|^2)} \begin{bmatrix} 1 & -\rho \\ -\rho^* & 1 \end{bmatrix}.
\]
(3.77)

Finally, a closed form solution for the quadratic form of the exponential term of the pdf is
\[
\frac{1}{2} \mathbb{E}^H \mathbf{C}_{xx}^{-1} \mathbb{E} = \frac{1}{2} \begin{bmatrix} |x|^2 & x \end{bmatrix} \sigma^2 (1 - |\rho|^2) \begin{bmatrix} 1 & -\rho \\ -\rho^* & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}
\]
\[
= \frac{1}{2 \sigma^2 (1 - |\rho|^2)} \begin{bmatrix} |x|^2 & x \end{bmatrix} \begin{bmatrix} x - \rho x^* \\ -\rho^* x + x^* \end{bmatrix}
\]
\[
= \frac{1}{2 \sigma^2 (1 - |\rho|^2)} \left( |x|^2 - \left( \rho x^2 + \rho^* x^2 \right) + |x|^2 \right)
\]
\[
= \frac{1}{\sigma^2 (1 - |\rho|^2)} \left( |x|^2 - \Re\left( \rho x^2 \right) \right).
\]
(3.78)

Using (3.78), the scalar Gaussian pdf can be expressed as
\[
p(x) = \frac{1}{\pi \sigma^2 \sqrt{1 - |\rho|^2}} \exp \left[ -\frac{|x|^2 - \Re(\rho x^2)}{\sigma^2 (1 - |\rho|^2)} \right].
\]
(3.79)

The pdf is now depending on the variance of $x$ and on the complex correlation coefficient $\rho$. If the correlation coefficient $\rho = 0$ then $x$ is proper and the contour of the pdf has the shape of a circle, illustrated by Figure 3.7. There, $x$ is also circular. With assumed properness, $\rho$ vanishes and (3.79) reduces to the simple form
\[
p(x) = \frac{1}{\pi \sigma^2} \exp \left[ -\frac{|x|^2}{\sigma^2} \right].
\]
(3.80)
This implies that the real part of \( x \) is independent of the imaginary one, \( \tilde{\sigma}^2 = 0 \). If the variance of the real and imaginary part is 1, then the pdf for the real-valued vector \( x_R \) is a standard bivariate Gaussian pdf [18].

Alternative insights on the impact of improperness can also be gained by inspecting the equivalent real vector and the real multivariate Gaussian distribution. To achieve improperness, correlations between \( x_r \) and \( x_i \) are introduced with the correlation coefficient

\[
\rho_R = \frac{\text{cov}(x_r, x_i)}{\sqrt{\sigma^2_{x_r} \sigma^2_{x_i}}}, \tag{3.81}
\]

where \(-1 \leq \rho_R \leq 1 \in \mathbb{R}\). If the correlation coefficient is zero, then \( x_r \) and \( x_i \) are uncorrelated and thus independent. The real covariance matrix of \( x_R = [x_r \ x_i]^T \) is (3.18). As already mentioned, the covariance matrix is symmetric and thus the relation \( \text{cov}(x_r, x_i) = \text{cov}(x_i, x_r) \) holds. Rearranging (3.81) yields the covariance

\[
\text{cov}(x_r, x_i) = \rho_R \sqrt{\sigma^2_{x_r} \sigma^2_{x_i}}. \tag{3.82}
\]

The determinant of (3.82) can now be expressed in terms of the correlation coefficient and the variances, \( \det(C_{x_Rx_R}) = \sigma^2_{x_r} \sigma^2_{x_i} (1 - \rho_R^2) \). The resulting bivariate pdf can be expressed as

\[
p(x) = \frac{1}{2\pi \sigma_{x_r} \sigma_{x_i} \sqrt{1 - \rho_R^2}} \exp \left[ -\frac{1}{2(1 - \rho_R^2)} \left( \frac{(x_r - \mu_{x_r})^2}{\sigma^2_{x_r}} + \frac{(x_i - \mu_{x_i})^2}{\sigma^2_{x_i}} - 2\rho_R (x_r - \mu_{x_r})(x_i - \mu_{x_i}) \right) \right]. \tag{3.83}
\]

With (3.83), an improper complex Gaussian pdf can be produced by variation of \( \rho_R \) and/or the variances. Figure 3.7 shows a strictly proper pdf. The contour of the joint pdf is a symmetric circle, centered around 0. If the correlation coefficient is nonzero, improperness is generated. Figure 3.8 depicts the impact of correlations on the joint pdf. If \( \rho_R \neq 0 \), the pdf loses its purely round shape and shifts towards an elliptic shape. If \( \rho_R = 0 \) and the mean values \( \mu_{x_r}, \mu_{x_i} \) are changed, the symmetric shape is preserved, only an offset is introduced. As already explained in Section 3.3.2, the mean has no influence on properness, but on the circularity characteristic. If the variances \( \sigma^2_{x_r} \neq \sigma^2_{x_i} \), then \( x \) becomes improper too. Figure 3.9 shows the effect if \( \sigma^2_{x_i} \) is increased while \( \sigma^2_{x_r} \) remains the same. In this case, the symmetric shape is destroyed too. \( x \) is improper, although there are no correlations between real and imaginary part. See appendix B.1 for the MATLAB script to create a complex-valued Gaussian random vector.

### 3.4.3 Conditional Gaussian pdf

Conditional pdfs are the fundamental of Bayesian estimation. Given two jointly distributed random variables \( x \) and \( y \) with joint pdf \( p(x, y) \), the conditional pdf gives the probability distribution density of one random variable when the other takes on a particular value.
It is mathematically expressed as $p(x|y)$. One of the most important rules in estimation theory comes with

$$p(x|y) = \frac{p(x,y)}{p(y)},$$

which is called Bayes’ theorem or alternative Bayes’ law [18, 19]. A very simple, but intuitive, example to calculate the conditional pdf is given if $x \sim \mathcal{N}(0, \sigma_x^2)$ and $y \sim \mathcal{N}(0, \sigma_y^2)$ are independent of each other and jointly Gaussian distributed with $p(x,y) = p(x)p(y)$. The joint pdf is given as in (3.64) and with individual pdfs as in (3.63). According to Bayes’ theorem the conditional pdf $p(x|y)$ calculates as

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x)p(y)}{p(y)} = p(x)$$

(3.85)

For nonindependent random variables the joint pdf (3.66) must be taken into account. Although $x$ and $y$ may be correlated, the conditional pdf would also be Gaussian [18] with mean and variance given as

$$E(x|y) = E(x) + \frac{\text{cov}(x,y)}{\sigma_y^2} (y - E(y))$$

(3.86)

$$\sigma_{x|y}^2 = \sigma_x^2 - \frac{\text{cov}(x,y)^2}{\sigma_y^2}$$

(3.87)

These results can be extended to the general vector case, where $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^N$:

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{(2\pi)^N \det \frac{1}{2} (C_{xx|y})} \exp \left[ -\frac{1}{2} (\mathbf{x} - E(\mathbf{x}|\mathbf{y}))^T C_{xx|y}^{-1} (\mathbf{x} - E(\mathbf{x}|\mathbf{y})) \right],$$

(3.88)

where the mean and covariance matrix are [18]

$$E(\mathbf{x}|\mathbf{y}) = E(\mathbf{x}) + C_{xy} C_{yy}^{-1} (\mathbf{y} - E(\mathbf{y}))$$

(3.89)

$$C_{xx|y} = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx}$$

(3.90)
Like before, this can be altered to the complex case. \(x\) and \(y\) are again Gaussian distributed, with individual distributions (3.69). It is assumed that both are jointly Gaussian distributed \(w = [x^T \ y^T]^T \sim CN(0, C_{xy}, \tilde{C}_{xy})\). The first intention to derive the conditional pdf is to use the equivalent real vectors \(w_{\mathbb{R}} = [x_{\mathbb{R}}^T \ y_{\mathbb{R}}^T]^T\). Performing the same steps as in Section 3.4, the properties of the real case can be simply adopted. Inserting the augmented vectors in (3.88) yields the result for the general complex conditional Gaussian pdf

\[
p(x|y) = \frac{1}{\pi^N \det \frac{1}{2}(C_{x|x|y})} \exp \left[ -\frac{1}{2} (x - E(x|y))^H C_{x|x|y}^{-1} (x - E(y|y)) \right].
\] (3.91)

With mean and covariance matrix as

\[
E(x|y) = TE(x_{\mathbb{R}}) + (TC_{x_{\mathbb{R}}y_{\mathbb{R}}} T^H) (T^{-H} C_{y_{\mathbb{R}}y_{\mathbb{R}}}^{-1} T^{-1}) T (y_{\mathbb{R}} - E(y_{\mathbb{R}})) = E(x) + C_{x_{\mathbb{R}}y_{\mathbb{R}}} C_{y_{\mathbb{R}}y_{\mathbb{R}}}^{-1} (y - E(y))
\] (3.92)
\[
C_{x|x|y} = TC_{x_{\mathbb{R}}x_{\mathbb{R}}} T^H - (TC_{x_{\mathbb{R}}y_{\mathbb{R}}} T^H) (T^{-H} C_{y_{\mathbb{R}}y_{\mathbb{R}}}^{-1} T^{-1}) (TC_{x_{\mathbb{R}}y_{\mathbb{R}}} T^H)
\] (3.93)

\section{Complex Random Processes}

This section is the first one, which actually deals with complex-valued random processes. Thus random variables additionally depend on time \(t\), i.e. \(x(t)\) is a random process. As such, also signals are treated as random process and can be described by their statistical properties. Figure 3.10 illustrates two time discrete random processes. From definition, a random process \(x(t)\) is at each time instance, \(\forall t\), a random variable. A discrete random process \(x[n]\) can be generated by sampling a continuous time signal. When it comes to moments, one has to differ between expected values and temporal averages [18].
CHAPTER 3. COMPLEX STATISTICS

Figure 3.9: $\rho_R = 0, \sigma_{x_r}^2 = 1, \mu_{x_r} = 0, \sigma_{x_i}^2 = 2.5, \mu_{x_i} = 0$

**Expected value** Average at a specific time instance, but over all possible realizations of the random process.

**Temporal average** Takes the average over one realization, but over all time.

For an ergodic process, the two definitions above are equivalent. This is true for continuous as well as for discrete signals.

In signal processing, it is common to work with the frequency domain representation of signals. As an example, the frequency response of a multipath channel offers a much better interpretation of its behavior than in time domain. This is also true for random processes/signals. There, the so-called power spectral density (PSD) is used. It is the (time discrete) Fourier transform of the autocorrelation function (ACF) and thus provides all relevant SO information. A classical example is the PSD of white Gaussian noise as shown in Figure 3.11. Looking at the time signal 3.11a, it is not possible to identify $x(t)$ as white process. For white noise, each sample is uncorrelated to the other ones. Thus, the discrete ACF 3.11b has only one peak at $k = 0$. Here for $x[n] \sim \mathcal{N}(0,1)$ the value of the peak is the variance $\sigma^2 = 1$. Consequently the PSD 3.11c is flat over all frequencies. Although the ACF is discrete, the PSD is a continuous function, which follows from the definition of the time discrete Fourier transformation [17]. To conclude, using PSDs and/or the ACFs makes identifying white noise quite easy. The reader may be already familiar with the concepts of PSDs from real-valued random processes/signals [18, 19]. Like in the previous sections these concepts can also be altered to complex-valued ones. There, properties like wide-sense stationary (WSS) are used as well, but sometimes interpreted differently.

Let $x(t)$ be a random process, then the autocorrelation function is given by $r_{xx}(t_1, t_2) = E[x(t_1)x(t_2)]$. It yields the correlation behavior of a signal with itself, but at different time instances. Without loss of generality, all further signals are assumed to have zero-mean. With this, the ACF is equivalent to the autocovariance function, $r_{xx}(t_1, t_2) = \text{cov}[x(t_1), x(t_2)]$. The ACF will be used further on to describe SO statistical properties.
(a) Gaussian random process, \( x[n] \sim \mathcal{N}(0, 1) \).

(b) Bitstream with probabilities, \( Pr(x[n] = 0) = Pr(x[n] = 1) = 1/2 \).

Figure 3.10: Time discrete random processes/signals.
Figure 3.11: White Gaussian process \( x \sim \mathcal{N}(0, \sigma^2 = 1) \).
CHAPTER 3. COMPLEX STATISTICS

If \( x(t) \) is a real-valued process, then the complete SO description is given by \( r_{xx}(t_1, t_2) \). However, if \( x(t) \in \mathbb{C} \), then the ACF is denoted as

\[
r_{xx}(t_1, t_2) = E \left[ x(t_1) x^*(t_2) \right]. \tag{3.94}
\]

This looks similar to (3.11), except for the additional time parameter. For \( x(t) \in \mathbb{C} \), (3.94) is not sufficient for a complete SO description. Hence, also the pseudo-A CF (pACF) must be taken into account. It is defined as

\[
\tilde{r}_{xx}(t_1, t_2) = E \left[ x(t_1) x(t_2) \right] \tag{3.95}
\]

Again, the augmented representation is used to completely describe SO statistics in the complex domain. The augmented autocorrelation matrix is given by

\[
\mathbf{R}_{xx}(t_1, t_2) = \begin{bmatrix} r_{xx}(t_1, t_2) & \tilde{r}_{xx}(t_1, t_2) \\ \tilde{r}_{xx}^*(t_1, t_2) & r_{xx}(t_1, t_2) \end{bmatrix}. \tag{3.96}
\]

All definitions can also be defined for discrete random processes, where \( t \) is replaced with discrete time steps, \( x(t)|_{t=kT_s} \doteq x[k] \). Of course, there also exists a cross-correlation (CCF) and pseudo-cross-correlation function (pCCF), which describes the correlation between two signals at specific time instances.

\[
r_{xy}(t_1, t_2) = E \left[ x(t_1) y^*(t_2) \right] \tag{3.97}
\]

\[
\tilde{r}_{xy}(t_1, t_2) = E \left[ x(t_1) y(t_2) \right] \tag{3.98}
\]

\[
\mathbf{R}_{xy}(t_1, t_2) = \begin{bmatrix} r_{xy}(t_1, t_2) & \tilde{r}_{xy}(t_1, t_2) \\ \tilde{r}_{xy}^*(t_1, t_2) & r_{xy}(t_1, t_2) \end{bmatrix}. \tag{3.99}
\]

Again, assume \( x(t) \in \mathbb{R} \). Then, stationary assumes all statistical properties to be invariant against time shifts. Hence, all moments are the same at each time, i.e. \( r_{xx}(t_1, t_2) = r_{xx}(t_3, t_4) \), where \( t_1 - t_2 = t_3 - t_4 \). The ACF and CCF are no longer dependent on the absolute position of the time, but only on the gap between them. With the introduction of the lag \( \tau = t_1 - t_2 = t_3 - t_4 \), the ACF can be rewritten as \( r_{xx}(\tau) = E \left[ x(t) x(t - \tau) \right] \). This can also be used on the CCF. For real-valued stationary signals the following properties hold:

\[
r_{xx}(0) \geq |r_{xx}(\tau)| \tag{3.100}
\]

\[
r_{xx}(-\tau) = r_{xx}(\tau) \tag{3.101}
\]

\[
r_{xy}(-\tau) = r_{yx}(\tau) \tag{3.102}
\]

The ACF at \( t = 0 \) is equivalent to the (mean) power of the signal, thus \( P_x = r_{xx}(0) \). This only holds if the process is stationary and has zero-mean. From (3.101) it follows that the ACF is an even function, see appendix A.4. In contrast to a stationary process, where all moments are time invariant, WSS only require first and second-order moments to be time shift invariant. Though, higher order moments may depend on absolute time positions. In signal processing applications, it is sometimes enough for a signal to be WSS, because estimators like BLUE, LMMSE only depend on first and SO statistics.

Researchers have defined WSS for complex-valued signals quite differently. For instance, in [26], WSS is only measured by the time shift invariance of the ACF (3.94). In
contrast, in [28], a process is called WSS if its ACF (3.94) and pACF (3.95) are time shift invariant. Whereas in [26], such a process is called second-order stationary (SOS). This thesis will go on with the definitions suggested by [26]. Thus, a process is WSS if the ACF is time shift invariant, and SOS if the time invariant assumption holds for ACF and pACF

\[
\begin{align*}
    r_{xx}(\tau) &= E [x(t)x^*(t-\tau)] \\
    \tilde{r}_{xx}(\tau) &= E [x(t)x(t-\tau)]
\end{align*}
\]  

(3.103) (3.104)

For complex signals SOS implies WSS. For purely real-valued functions, WSS and SOS are the very same. The complex ACF is not an even function but a Hermitian function, i.e.

\[
r_{xx}(\tau) = r^*_{xx}(-\tau).
\]

See appendix A.4 for the proof. Further, for \(x(t)\) to be an SOS process it is required that the pACF is an even function \(\tilde{r}_{xx}(\tau) = \tilde{r}_{xx}(-\tau)\) [26].

As mentioned above, another way to look on the statistics of a random signal is to use the PSD. For real-valued signals it is given as the Fourier transform of the ACF

\[
S_{xx}(f) = \text{FT}[r_{xx}(\tau)] [17].
\]

This concept can also be altered to complex-valued signals, given two different PSDs, one for the ACF, denoted as complex PSD or Hermitian PSD, and one for the pACF, denoted as pseudo-PSD (pPSD):

\[
S_{xx}(f) = \int_{-\infty}^{\infty} r_{xx}(\tau) \exp(-j 2 \pi f \tau) d\tau,
\]

(3.105)

\[
\tilde{S}_{xx}(f) = \int_{-\infty}^{\infty} \tilde{r}_{xx}(\tau) \exp(-j 2 \pi f \tau) d\tau
\]

(3.106)

In the discrete case, these are given as the time-discrete Fourier transform of the ACF and pACF:

\[
S_{xx}(f) = \sum_{m=-\infty}^{\infty} r_{xx}[m] \exp(-j 2 \pi f T_s m),
\]

(3.107)

\[
\tilde{S}_{xx}(f) = \sum_{m=-\infty}^{\infty} \tilde{r}_{xx}[m] \exp(-j 2 \pi f T_s m)
\]

(3.108)

While the Hermitian PSD is defined as the power spectral density of \(x(t)\), the pPSD, or more precisely its magnitude, characterizes the impropriety of a random process in the frequency domain [28]. However, one can extend the definitions to the augmented representation, where the augmented PSD-matrix is the Fourier transform of the augmented correlation matrix, i.e. \(\text{FT} [R_{xx}(\tau)]\). It can be shown that the augmented PSD-matrix takes on the form of [2]

\[
S_{xx}(f) = \begin{bmatrix}
S_{xx}(f) & \tilde{S}_{xx}(f) \\
\tilde{S}^*_{xx}(-f) & S^*_{xx}(-f)
\end{bmatrix},
\]

(3.109)

The following results of [2] and [26] are rather shortly summarized here. The augmented matrix is positive semi-definite. As a consequence the PSD must be real and nonnegative

\[
S_{xx}(f) \geq 0,
\]

(3.110)

but in general must not be even. This is a consequence of the Hermitian form of the ACF. The second condition implies that the pPSD, which is generally complex, must be even

\[
\tilde{S}_{xx}(f) = \tilde{S}_{xx}(-f).
\]

(3.111)
This condition follows from the even form of the pACF. Additionally, the PSD gives a bound on the magnitude of the pPSD

$$|\tilde{S}_{xx}(f)|^2 \leq S_{xx}(f) S_{xx}(-f),$$

(3.112)

which is required for the determinant of the augmented PSD-matrix to be nonnegative, i.e. \(\det[S_{xx}(f)] \geq 0\). Further, due to (3.110) and (3.111) the augmented PSD-matrix reduces to

$$S_{xx}(f) = \begin{bmatrix} S_{xx}(f) & \tilde{S}_{xx}(f) \\ \tilde{S}_{xx}^*(f) & S_{xx}(-f) \end{bmatrix}$$

(3.113)

For the corresponding real PSD as the Fourier transform of the ACFs it can be easily shown that

$$S_{xRxR}(f) = \begin{bmatrix} S_{xRxR}(f) & S_{xRxI}(f) \\ S_{xRxI}^*(f) & S_{xIxi}(f) \end{bmatrix}.$$  

(3.114)

The augmented PSD (3.109) can now be derived from (3.114) with transformation (3.40)

$$S_{xx}(f) = T S_{xRxR} T^H.$$  

(3.115)

It can be shown that \(S_{xRxR}(f) = S_{xRxI}(f)\) hold and therefore the separate block matrices of (3.114) are given as [26]

$$S_{xx}(f) = S_{xRxR}(f) + S_{xRxI}(f) + 2 \Im \{S_{xRxI}(f)\}$$  

(3.116)

$$\tilde{S}_{xx}(f) = S_{xRxR}(f) - S_{xRxI}(f) + 2j \Re \{S_{xRxI}(f)\}$$  

(3.117)

It is important to know that an SOS process \(x(t)\) is called proper if and only if the pPSD vanishes. Of course, this results from the vanishing autocorrelation matrix. From (3.117) it follows that \(\tilde{S}_{xx}(f)\) is only zero if and only if the following conditions hold:

$$S_{xRxR} = S_{xRxI},$$  

(3.118)

$$\Re \{S_{xRxI}(f)\} = 0,$$  

(3.119)

Therefore (3.118) implies that \(r_{xRxR}(\tau) = r_{xRxI}(\tau)\). For (3.119) to be true, it is essential that \(r_{xRxI}(\tau) = -r_{xRxI}(-\tau)\). It is apparent that for proper signals the ACFs of the real and imaginary part must be equal and also the CCF must be a real and odd function. It can be shown that an analytic SOS signal without a DC-component (\(S_{xx}(0) = 0\)) is proper [28]. Further, modulation keeps propriety. Ergo, the equivalent baseband signal of an SOS bandpass signal is always proper.
Chapter 4

Estimators

There are two different approaches in estimation theory. One approach treats the parameter as a deterministic, but unknown constant. The other one as a random variable, given a certain pdf. For instance, imagine a sine wave transmitted over an AWGN channel. The amplitude of the sine wave shall then be estimated. At the receiver, the measured signal is \( y(t) = A s(t) + n(t) \), where \( s(t) \) is a known (carrier) signal, \( n(t) \) is additive noise, and \( A \) is the amplitude. The following statements are, of course, not limited to the mentioned example. The deterministic approach is called \textit{classical estimation}. Hence, it is assume that \( A \) is a fixed, but unknown parameter. Estimators using this concept are part of the \textit{classical estimators}. Well known estimators using this technique are the \textit{best linear unbiased estimator} (BLUE), \textit{least squares} (LS) estimator, \textit{classical mean square error} (MSE) estimator, and the \textit{minimum variance unbiased} (MVU) estimator. In Section 4.2 these estimators are described for proper signal constellations. The other way is denoted as the \textit{Bayesian estimation} approach. Here, the parameter to be estimated is no longer deterministic, but random. It is called the \textit{Bayesian approach} because of its dependence on Bayes’ theorem (3.84). The task for these estimators is to estimate one distinct realization of \( A \). All possible outcomes are fully described by its pdf \( p(A) \), which is called the \textit{prior pdf}. For details on pdfs see Chapter 3.1. The term prior indicates that the pdf is known even before observing the data \( y(t) \). This knowledge influences the estimation algorithms in a positive way as additional knowledge reduces entropy and thus improves the performance of the estimators. In fact, the spread of estimated parameters can be limited or at least, put in a specific direction. Again, imagine the example above. One may know that \( A \) is uniformly distributed over the interval \([1, 2]\). So it is impossible that \( A = 4 \) at the receiver. In contrast, in classical estimation, these boundaries are not considered. Consequently it would be possible that the estimated parameter \( \hat{A} = 4 \). Bayesian estimators are described in Section 4.1.

4.1 Bayesian Estimators

The underlying concept of the Bayesian \textit{minimum mean squared error} (MMSE) approach is the \textit{Bayesian mean square error} (BMSE). It minimizes the MSE in the Bayesian sense,
CHAPTER 4. ESTIMATORS

also considering the prior pdf. The cost function is defined as

\[ \text{BMSE}(\hat{x}) = J = E_{y,x} \left[ (x - \hat{x})^T (x - \hat{x}) \right] \] (4.1)

\[ = \int \int (x - \hat{x})^T (x - \hat{x}) \ p(y,x) \ dy \ dx \] (4.2)

In literature there are several definitions of the BMSE cost function. In [19], the cost function which minimizes the BMSE is defined by minimizing each individual MSE

\[ \text{BMSE}_{\text{Bay}}(\hat{x}_i) = J_{\text{Bay}} = E_{y,x} \left[ (x_i - \hat{x}_i)^2 \right] \] (4.3)

Another alternative definition is [2]

\[ \text{BMSE}_{\text{Schreier}}(\hat{x}) = J_{\text{Schreier}} = E_{y,x} \left[ \text{tr} \left\{ (x - \hat{x}) (x - \hat{x})^T \right\} \right] \] (4.4)

which is the sum of all individual BMSEs as in (4.3) and is equivalent to the definition in (4.1). It can be shown that minimizing the individual BMSEs (4.3) also minimizes the sum of the BMSEs (4.4). Although the definitions above are different, all of them yield minimize the squared error in the Bayesian sense. This thesis will go on with the cost function defined in (4.1). Thus all further mentioned BMSEs minimize the sum of each individual BMSE. The estimator \( \hat{x} \) is the optimal estimator to minimize the BMSE and will further be denoted as the MMSE estimator. In contrast to the MSE in classical estimation,

\[ \text{MSE}_{\text{class}} = E_y \left[ (x - \hat{x})^T (x - \hat{x}) \right] \]

\( y \) is the measurement/observation and \( x \) the signal/source, which is to be estimated. For convenience, the lower case letters at the expectation operators will be dropped. Using Bayes’ theorem (3.84), the integral in (4.2) can be rewritten as

\[ J = \int \int (x - \hat{x})^T (x - \hat{x}) \ p(x|y) \ dx \ p(y) dy, \] (4.5)

whereas the inner integral is on the conditional pdf \( p(x|y) \) and the outer one \( p(y) \). A pdf is never negative. Hence, integrating \( p(y) \) only increases \( J \). To minimize (4.5), it is enough to minimize the inner integral. The optimum estimator is given by the mean of the posterior pdf [19]

\[ \hat{x} = E(x|y) \] (4.6)

The error covariance matrix \( C_{ee} \) is the expectation of the conditional covariance matrix [19]

\[ C_{ee} = E \left[ e e^H \right] = E \left( C_{x|y} \right), \] (4.7)

where the diagonal entries are the BMSE of each parameter. Note, if \( x \) and \( y \) are jointly Gaussian distributed then it can be shown that \( C_{x|y} \) no longer depends on \( y \) and thus the error covariance matrix is equivalent to the conditional covariance matrix [19]

\[ C_{ee} = C_{x|y} = C_{xx} - C_{xy} C_{xx}^{-1} C_{yx}. \] (4.8)

In (4.7) and (4.8), the Hermitian transpose has already been used, making an extra definition for the complex case unnecessary. For \( (x - \hat{x}) \in \mathbb{R}^N \), the Hermitian transpose \((\cdot)^H\) reduces to a normal transpose \((\cdot)^T\).
In the complex case, the cost function (4.1) is altered to
\[
\text{BMSE}(\hat{x}) = J = E \left[ (x - \hat{x})^H (x - \hat{x}) \right].
\]  
(4.9)

For \( \hat{x} \in \mathbb{C}^1 \), the BMSE minimizes to the expectation of the squared error magnitude
\[
J = E \left[ |x - \hat{x}|^2 \right].
\]
Like in the real case, the covariance is also the error covariance (4.7). The optimum estimator is (4.6).

The focus on Bayesian estimation in this thesis is only on the linear case. The estimator is assumed to be linear in the measurement data. It is denoted as the \textit{linear minimum mean square error} (LMMSE) estimator. In general, it is very difficult to find the mean of the posterior pdf. In some cases it is even impossible. The linear (actually affine) constraint, i.e. \( \hat{x} = W y + b \), reduces the derivation to a rather simple task. Matrix \( W \) are weighting coefficients and the offset parameter \( b \) is used to compensate nonzero means of \( y \) and \( x \). \( b \) may be omitted if both vectors have zero-mean. In the linear case, the task is to find weighting coefficients, which minimize (4.9). The LMMSE estimator and its Bayesian MMSE matrix are given by [19]
\[
\hat{x} = E(x) + C_{xy} C_{yy}^{-1} [y - E(y)]
\]  
(4.10)
\[
C_{ee} = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx}
\]  
(4.11)

The LMMSE only requires knowledge of first and second order moments. It is important to know that (4.10) is only optimal if \( x \) and \( y \) are jointly Gaussian because then (4.8) is equivalent to (4.7). If not, then the LMMSE is only suboptimal, but still the best estimator over all linear ones. There may be better non linear ones, but in the non-Gaussian case closed form solutions are hard or even impossible to find [19]. In case of the linear model the LMMSE estimator reads as follows.

\textbf{Theorem 4.1.1.} (Bayesian Gauss-Markov Theorem) If the observed data can be modeled as \( y = H x + n \), where \( y \) is an \( N \times 1 \) data vector, \( H \) a known \( N \times p \) matrix, \( x \) a \( p \times 1 \) random vector with mean \( E(x) \) and covariance matrix \( C_{xx} \), and \( n \) a \( N \times 1 \) random vector with zero-mean and covariance matrix \( C_{nn} \) and uncorrelated to \( x \), then the LMMSE estimator of \( x \) is
\[
\hat{x} = E(x) + C_{xx} H^T (H C_{xx} H^T + C_{nn})^{-1} [y - E(y)]
\]  
(4.12)

The performance of the estimator is measured by the error \( e = x - \hat{x} \), whose mean is zero and whose covariance matrix is
\[
C_{ee} = C_{xx} - C_{xx} H^T (H C_{xx} H^T + C_{nn})^{-1} H C_{xx}
\]  
(4.13)

The complex LMMSE is a straightforward extension to the real one. It is shown that the difference between real and complex one is only in the Hermitian transpose instead of the normal transpose. Now again the scalar case will be used to derive the complex LMMSE. In the complex case, the estimator is assumed to be linear in the data, i.e.
\[
\hat{x} = w^H y,
\]  
(4.14)
whereas \( \mathbf{w} \in \mathbb{C}^N \) are the weighting coefficients, \( \mathbf{y} \in \mathbb{C}^N \) the observation data, and \( \hat{x} \) the estimate of \( x \). In signal processing one can think of \( \mathbf{w} \) as filter coefficients and \( \mathbf{y} \) as a vector of measurements. In literature, filters based on the LMMSE are called Wiener-Filters. There are two ways to get the optimum weighting coefficients. One way is to differentiate the BMSE cost function, set it to zero and get \( \mathbf{w} \) which fulfills the equation \( \frac{\partial J}{\partial \mathbf{w}^*} = 0 \). It is a rather straightforward approach. Find the extrema, in this case the minimum of \( J \). Then, insert the derived coefficients to get the optimum result. The other approach uses the so called orthogonality principle. There, the minimum is found by setting the error orthogonal to \( \mathbf{y} \). Further details will be explained in the derivations below. The orthogonality principle allows a very intuitive and simple way to derive the LMMSE estimator. The following derivations are first shown for the scalar case and then altered to the vector case. Without loss of generality all random variables/vectors are assumed to have zero-mean. If this is not the case, then \( x' = x - E(x) \) may be used to achieve zero-mean. Therefore, the correlation is equal to the covariance.

### 4.1.1 Complex LMMSE

The estimator model for the scalar case is (4.14). No additional offset is assumed. \( \hat{x} \) is strictly linear dependent on \( \mathbf{y} \). The BMSE cost function is

\[
J = E[|e|^2] = E[(x - \hat{x})(x - \hat{x})^*].
\]

Inserting (4.14) yields

\[
J = E[(x - \mathbf{w}^H \mathbf{y})(x^* - \mathbf{w}^T \mathbf{y}^*)]
= E[|x|^2 - \mathbf{w}^H \mathbf{y} x^* - x \mathbf{w}^T \mathbf{y}^* + \mathbf{w}^H \mathbf{y} w \mathbf{y}^*]
\]

The following approach shows the derivation of \( \mathbf{w} \) by setting the gradient of the cost function \( J \) with respect to \( \mathbf{w}^* \) to zero. Note, it would also be possible to differentiate with respect to \( \mathbf{w} \).

\[
\mathbf{w}^T \mathbf{y}^* = (\mathbf{w}^H \mathbf{y})^* = (\mathbf{y}^T \mathbf{w}^*)^* = \mathbf{y}^H \mathbf{w}
\]

Thus the cost function (4.18) becomes

\[
J = E[|x|^2 - x^* \mathbf{w}^H \mathbf{y} - x \mathbf{y}^H \mathbf{w} + \mathbf{w}^H \mathbf{y} \mathbf{y}^H \mathbf{w}]
\]

Using the linear behavior of the expectation operator yields

\[
J = E(|x|^2) - \mathbf{w}^H \left( E(\mathbf{y} x^*) - E(\mathbf{y} \mathbf{y}^H) \right) \mathbf{w} + \mathbf{w}^H \left( E(\mathbf{y} \mathbf{y}^H) \right) \mathbf{w},
\]

with \( \mathbf{C}_{yx} \in \mathbb{C}^{N \times 1} \) and \( \mathbf{C}_{xy} \in \mathbb{C}^{1 \times N} \) as correlation vectors, and \( \mathbf{C}_{yy} \) as the covariance matrix of the measurements. Now (4.21) is differentiated with respect to \( \mathbf{w}^* \) and set to 0

\[
\frac{\partial J}{\partial \mathbf{w}^*} = \frac{\partial}{\partial \mathbf{w}^*} \left[ E(|x|^2) - \mathbf{w}^H \mathbf{C}_{yx} - \mathbf{C}_{xy} \mathbf{w} + \mathbf{w}^H \mathbf{C}_{yy} \mathbf{w} \right] \equiv 0
\]
Using the rules of the CR-Calculus, see Section 2.1.2, yields

\[ 0 = C_{yy}w - C_{yx} \]  
\[ C_{yy}w = C_{yx}. \] (4.23) (4.24)

(4.24) are the so called normal equations. In contrast to the real case, the additional factor of 2 is missing in (4.23). Rewriting (4.24) results in the Wiener-Hopf solution for the complex case

\[ w_{\text{opt}} = C_{yy}^{-1} C_{yx}. \] (4.25)

The best linear estimator in the Bayesian sense is given by

\[ \hat{x} = w_{\text{opt}}^H y \]
\[ = (C_{yy}^{-1} C_{yx})^H y \]
\[ = C_{xy} C_{yy}^{-1} y \] (4.26)

As mentioned above, the optimum estimator looks very similar to (4.10). The differences are the definitions of the cross-covariance vector and autocovariance matrix. Further, in \( \mathbb{C} \) the Hermitian transpose is used instead of the normal transpose as in \( \mathbb{R} \). Equation (4.26) uses the property \( C_{yx}^H = C_{xy} \).

**Proof.** Let \( y \) be an \( N \)-dimensional complex-valued vector and \( x \) an complex-valued scalar with zero-mean. Then the definitions for the cross-correlation are

\[ C_{yx} = E(\bar{y}x^*) \] (4.27)

and

\[ C_{xy} = E(xy^H) \] (4.28)

It is simple to show that

\[ E(\bar{y}x^*)^H = E(xy^H), \] (4.29)

which is then equivalent to the definition in (4.28). It can be shown that (4.29) also holds for the vector case

\[ E(\bar{y}x^H)^H = E(xy^H) \] (4.30)

Inserting (4.26) into the BMSE cost function (4.9) yields

\[
\text{BMSE}(\hat{x}) = E(e e^*) \\
= E(|x|^2) - C_{xy} C_{yy}^{-1} C_{yx} - C_{yx}^H C_{yy}^{-1} C_{yx} + C_{yx}^H C_{yy}^{-1} C_{yx} C_{yy}^{-1} C_{yx} \\
= E(|x|^2) - C_{xy} C_{yy}^{-1} C_{yx} - C_{yx}^H C_{yy}^{-1} C_{yx} + C_{yx}^H C_{yy}^{-1} C_{yx} \\
= E(|x|^2) - \frac{C_{xy}}{E(xy^H)} \frac{C_{yy}^{-1}}{E(y^*y)} \frac{C_{yx}}{E(y^*x)} \] (4.31)

(4.31) is the optimal result for the BMSE, considering the linear estimator model in (4.14).
The concept of the linear estimator model is now extended to the vector case. Therefore, \( x \) is no longer a scalar, but a \( p \)-dimensional vector of parameters whose particular realization has to be estimated. Let the estimator again be linear depended on the measurements. The estimator model becomes

\[
\hat{x} = W y,
\]

where \( W \) is now a \( p \times N \)-dimensional matrix. The solution of the LMMSE can be simple extended to the vector case, where each entry is defined as in (4.26):

\[
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\vdots \\
\hat{x}_p 
\end{bmatrix} =
\begin{bmatrix}
C_{x_1 y} & C_{y_1 y} \\
C_{x_2 y} & C_{y_2 y} \\
\vdots & \vdots \\
C_{x_p y} & C_{y_p y}
\end{bmatrix}
\begin{bmatrix}
C_{y_1 y}^{-1} \\
C_{y_2 y}^{-1} \\
\vdots \\
C_{y_p y}^{-1}
\end{bmatrix}
\]

\[
\hat{x} = C_{xy} C_{yy}^{-1} y
\]

This simple extension can be done because, as already mentioned, minimizing each individual BMSE also minimizes the sum of the BMSEs. The coefficient matrix \( W = C_{xy} C_{yy}^{-1} \). \( C_{xy} \) is no longer a vector, but a \( p \times N \) dimensional matrix. The BMSE also extends to a matrix, given in the same form as (4.31):

\[
\text{BMSE}(\hat{x}) = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx} \quad \text{with} \quad C_{yy} = \text{diag}(C_{yy})
\]

Note, (4.34) is also the error covariance matrix \( C_{ee} \), where the main diagonal entries are the BMSEs of each parameter, \( \text{BMSE}(\hat{x}_n) = [C_{ee}]_{nn} \). \( \hat{x} \) depends on the cross-covariance matrix between the parameter and the measurements and on the autocovariance matrix of \( y \). \( y \) must somehow be correlated to \( x \). Otherwise, Bayesian estimation would not make any sense.

However, until now it is only assumed that \( y \) depends on \( x \) via an arbitrary function \( y = f(x) \). As already mentioned if \( f(x) \) is a nonlinear function, then the LMMSE is only suboptimal. If the measurement data can be modeled as

\[
y = H x + n,
\]
CHAPTER 4. ESTIMATORS

55

The autocovariance matrix is given by

\[ C_{yy} = E(\mathbf{y}\mathbf{y}^H) \]

\[ = E \left[ (\mathbf{Hx} + \mathbf{n}) (\mathbf{Hx} + \mathbf{n})^H \right] \]

\[ = E \left[ (\mathbf{Hx} + \mathbf{n}) (\mathbf{x}^H \mathbf{H}^H + \mathbf{n}^H) \right] \]

\[ = E \left[ \mathbf{Hx} \mathbf{x}^H \mathbf{H}^H + \mathbf{Hx} \mathbf{n}^H + \mathbf{n} \mathbf{x}^H \mathbf{H}^H + \mathbf{nn}^H \right] \]

\[ = \mathbf{H} E(\mathbf{xx}^H) \mathbf{H}^H + \mathbf{H} E(\mathbf{xn}^H) + E(\mathbf{nx}^H) \mathbf{H}^H + E(\mathbf{nn}^H) \]  

(4.36)

\[ = \mathbf{H} \mathbf{C}_{xx} \mathbf{H}^H + \mathbf{C}_{nn} \]  

(4.37)

Further, the cross-covariance matrix reduces to

\[ C_{xy} = E(\mathbf{xy}^H) \]

\[ = E \left[ \mathbf{x} (\mathbf{Hx} + \mathbf{n})^H \right] \]

\[ = E(\mathbf{xx}^H \mathbf{H}^H + \mathbf{xn}^H) \]

\[ = E(\mathbf{xx}^H) \mathbf{H}^H + E(\mathbf{xn}^H) \]  

(4.38)

\[ = \mathbf{C}_{xx} \mathbf{H}^H \]  

(4.39)

Of course, this is only true if \( \mathbf{n} \) is uncorrelated to \( \mathbf{x} \), which results in the properties in (4.36) and (4.38). The results above are again related to theorem 4.1.1. Inserting these definitions in (4.33) yields

\[ \hat{\mathbf{x}} = \mathbf{C}_{xx} \mathbf{H}^H (\mathbf{H} \mathbf{C}_{xx} \mathbf{H}^H + \mathbf{C}_{nn})^{-1} \mathbf{y} \]  

(4.40)

with error covariance matrix

\[ \mathbf{C}_{ee} = \mathbf{C}_{xx} - \mathbf{C}_{xx} \mathbf{H}^H (\mathbf{H} \mathbf{C}_{xx} \mathbf{H}^H + \mathbf{C}_{nn})^{-1} \mathbf{HC}_{xx} \]  

(4.41)

Following example shows an applied application in signal processing, where the LMMSE is used as data estimator after a channel transmission.

**Example 4.1.1.** Assume random data over an AWGN channel. The data model is \( \mathbf{y} = \mathbf{Hx} + \mathbf{n} \), where \( \mathbf{y} \) and \( \mathbf{n} \) are \( N \)-dimensional random vectors. \( \mathbf{H} \) is an \( N \times p \)-dimensional identity matrix, indicating that this is an AWGN channel. \( \mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_{xx} = \sigma_x^2 \mathbf{I}_p, \mathbf{0}) \) is random data. \( \mathbf{x} \) is white and proper. See sections 3.3.2 and 3.4 for details. It is also assumed that \( \mathbf{x} \) and \( \mathbf{y} \) are cross-proper, i.e. \( \mathbf{C}_{xy} = \mathbf{0} \). The noise is distributed as \( \mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_{nn} = \sigma_n^2 \mathbf{I}_N, \mathbf{0}) \) and independent of \( \mathbf{x} \). As shown above, the autocovariance of the measurements is (4.37). With that, the error covariance matrix (4.41) derives as follows

\[ \mathbf{C}_{ee} = \sigma_x^2 \mathbf{I}_p - \sigma_x^2 \mathbf{I}_p \mathbf{H}^H (\mathbf{H} \mathbf{I}_p \mathbf{H}^H \sigma_x^2 + \sigma_n^2 \mathbf{I}_N)^{-1} \mathbf{H} \sigma_x^2 \mathbf{I}_p \]

\[ = \sigma_x^2 \mathbf{I}_p - \sigma_x^2 \mathbf{I}_{p \times N} \left[ (\sigma_x^2 + \sigma_n^2) \mathbf{I}_N \right]^{-1} \mathbf{I}_{N \times p} \sigma_x^2 \]

\[ = \sigma_x^2 \mathbf{I}_p - \frac{\sigma_x^4}{\sigma_x^2 + \sigma_n^2} \mathbf{I}_p \]

\[ = \left[ \sigma_x^2 - \frac{\sigma_x^4}{\sigma_x^2 + \sigma_n^2} \right] \mathbf{I}_p \]  

(4.42)
From that, the BMSE of each parameter is given in the diagonal entries of (4.42). Further, the optimal estimator is

\[
\hat{x} = C_{xy} C_{yy}^{-1} y \\
= \sigma_x^2 I_{p \times N} \left[ \left( \frac{\sigma_x^2 + \sigma_n^2}{\sigma_x^2} \right) I_N \right]^{-1} y \\
= \frac{1}{\sigma_x^2 + \sigma_n^2} I_{p \times N} y \\
= \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} I_{p \times N} y
\]

(4.43)

Summarized, the best estimator is still the expected value of the posterior pdf. However, in contrast to the general case \( \hat{x} = E(x|y) \), the LMMSE only requires knowledge of first and second order moments. It can be shown that the LMMSE is optimal in the Bayesian MSE sense if \( x \) and \( y \) are jointly Gaussian distributed. If not, then the LMMSE is still the best performing linear estimator. In contrast to nonlinear estimators, if even possible to find, the LMMSE requires relatively low implementation costs. Thus in practice it is often used even when the Gaussian assumption is not fulfilled and the data model is known to be nonlinear. It is up to the engineer to decide if the performance of the estimator meets the requirements. Additionally, \( x \) must be proper and cross-proper to \( y \). If this is not the case, then the estimator is no longer optimal. The complex LMMSE does not consider pseudo-statistics, which as shown in Section 3.3, are crucial for a complete SO description. To overcome these limits the estimator must be extended.

### 4.1.2 WLMMSE

In Section 4.1.1 the estimator model \( \hat{x} = w^H y \) has been used. There, the estimator is strictly linear in \( y \). In general, see again Section 2.3, this is not the case. Because a linear transformation in \( \mathbb{R} \) results in a widely linear transformation (2.52) in \( \mathbb{C} \). Then, \( \hat{x} \) is linear in \( y \) and \( y^* \). The widely linear estimator model is

\[
\hat{x}_{wl} = h^H y + g^H y^*. 
\]

(4.44)

The task is to find weighting coefficients \( h \) and \( g \), which reduce the Bayesian MSE. The estimator model (4.44) varies in literature. While some researchers use the normal transpose for the weighting coefficients, others use the Hermitian transpose. Nevertheless, in the end the result is the very same, regardless which approach is used to derive the coefficients. This thesis will go on with (4.44). Using (4.44) in the BMSE cost function results in a quite long and complicated form: \( E \left[ (x - h^H y - g^H y^*) \left( x - h^H y - g^H y^* \right)^H \right] \). Even with predefined rules of the \( \mathbb{C}\mathbb{R}\)-Calculus, the derivation is very time-consuming. It can be done, but there is a much simpler way, making the derivation very similar to that of the LMMSE. The reader may look back to Section 2.3. There, the WL transformation was derived after using the augmented representation. The same principle can be used here. Instead of using (4.44), the augmented form shall be used

\[
\hat{x}_{wl} = w^H \underline{y}. 
\]

(4.45)

With \( w = [h^T \; g^T]^T \) and \( \underline{y} = [y^T \; y^H]^T \). \( w \) is not an augmented vector, because as will be shown \( h^* \neq g \). The estimator \( \hat{x}_{wl} \) is still a scalar and thus also the error, \( e_{wl} = x - \hat{x}_{wl} \).
The big advantage when using (4.45) is the similar form to the scalar LMMSE data model (4.14). Although, the theory behind is different, the differentiation rules are the very same. Again, zero-mean vectors are assumed. For convenience, the \( w \) sign will be dropped in the derivation and added to the result at the end. The following lines are not shown in detail in any used literature.

\[
J' = E[ee^*] = E[(x - \hat{x})(x - \hat{x})^*].
\]

\[
= E[(x - w^H y)(x^* - w^T y^*)] = E[|x|^2 - w^H y x^* - x w^T y^* + w^H y w^T y^*] = E(|x|^2 - x^* w^H y - x y^H w + w^H y y^H w].
\]

\[
= E(|x|^2) - w^H E(y x^*) - E(x y^H) w + w^H E(y y^H) w
\]

\[
\downarrow \frac{\partial}{\partial w^*} \equiv 0
\]

\[
0 = \frac{\partial}{\partial w^*} E(|x|^2) - \frac{\partial}{\partial w^*} w^H C_{yx} - \frac{\partial}{\partial w^*} C_{xy} w + \frac{\partial}{\partial w^*} w^H C_{yy} w
\]

Using the complex gradients (2.25), (2.26), and (2.29) yields

\[
0 = 0 - C_{yx} w - 0 + C_{yy} w
\]

\[
= C_{yy} w - C_{yx} w
\]

(4.46)

This again produce the normal equations and the Wiener-Hopf solution:

\[
C_{yy} w = C_{yx}
\]

(4.47)

\[
w_{opt} = C_{yy}^{-1} C_{yx}
\]

(4.48)

From (4.48) follows the solution for the WLMMSE

\[
\hat{x}_{wl} = w^H y = C_{xy} C_{yy}^{-1} y.
\]

(4.49)

Notice that $C_{yy}$ is an augmented matrix with structure as defined in (3.23). Contrary, $C_{xy}$ is not an augmented matrix. It is a $1 \times 2N$-dimensional vector defined as

\[
C_{xy} = E \left( x \begin{bmatrix} y \\ y^* \end{bmatrix}^H \right) = \begin{bmatrix} E(x y^H) \\ E(x y^T) \end{bmatrix} = \begin{bmatrix} C_{xy} \\ \tilde{C}_{xy} \end{bmatrix}
\]

(4.50)

With the same arguments as for the LMMSE estimator the solution can be simple extended to the vector case, where $\hat{x}_{wl}$ is a $p$-dimensional vector of estimators.

\[
\hat{x}_{wl} = C_{xy} C_{yy}^{-1} y.
\]

(4.51)

Each entry is defined as in (4.49). The effect of $\hat{x}_{wl}$ being a vector influences the cross-covariance matrix $C_{xy}$. It becomes a $p \times 2N$-dimensional matrix. As mentioned, the
derivation is the same, and one gets similar looking solutions as for the LMMSE. The big
difference are the considered pseudo-covariances in the WLMMSE.

In the following, the augmented representation will be used. Thus expressions like
$C_{xy} = E(xy)$, where a normal scalar/vector is multiplied with an augmented one, can be
avoided. The following estimator model will be used

$$\hat{x} = Wy.$$  

(4.52)

where $W$ is an augmented matrix (2.49). (4.52) is also denoted as the augmented LMMSE
(ALMMSE), [16]. The ALMMSE is just the augmented representation of the WLMMSE,
which shall now be modeled as

$$\hat{x}_{\text{wl}} = W_1 y + W_2 y^*$$  

(4.53)

This approach is also used in [2, 28]. Again, the following detailed derivations of $W_1$ and
$W_2$ have not been found in used literature. Though, the basic concept is suggested in [28].
Altering (4.51) to the augmented representation ($x \rightarrow \bar{x}$) yields the ALMMSE

$$\hat{\bar{x}} = \frac{C_{xy}}{\bar{w}} C_{yy}^{-1} y.$$  

(4.54)

with $\bar{W}$ given from the augmented Wiener-Hopf solution [2]

$$\bar{W} = C_{xy} C_{yy}^{-1}$$  

(4.55)

Note that (4.54) yields the same results for $\hat{x}$ as (4.51) plus the conjugate results $\hat{x}^*$. $W$
can be divided into the four block matrices, whose particular realizations are given by the
equation

$$W = \begin{bmatrix} W_1 & W_2 \\ W_2^* & W_1^* \end{bmatrix} = \begin{bmatrix} C_{xy} & \bar{C}_{xy} \\ \bar{C}_{xy}^* & C_{yy} \end{bmatrix} \begin{bmatrix} C_{yy} & \bar{C}_{yy} \\ \bar{C}_{yy}^* & C_{yy} \end{bmatrix}^{-1}.$$  

(4.56)

$W_1$ and $W_2$ are $p \times N$-dimensional matrices. From (4.55) it is required to invert the
augmented autocovariance matrix. $C_{yy}$ can also be divided in four block matrices and
hence inverted by the matrix inversion lemma. Given the following equation

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix},$$  

(4.57)

the matrix inversion lemma yields following result:

$$E = (A - BD^{-1}C)^{-1}$$  

(4.58)

$$F = -A^{-1}B (D - CA^{-1}B)^{-1}$$  

(4.59)

$$G = -D^{-1}C (A - BD^{-1}C)^{-1}$$  

(4.60)

$$H = (D - CA^{-1}B)^{-1}$$  

(4.61)
CHAPTER 4. ESTIMATORS

For more details see [20] or appendix of [2]. Using the lemma on the augmented autocovariance matrix \( C_{yy} \) yields

\[
C_{yy}^{-1} = \begin{bmatrix}
(C_{yy} - \tilde{C}_{yy} C_{yy}^{-1} \tilde{C}_{yy}^*) & -C_{yy}^{-1} \tilde{C}_{yy} \left( C_{yy} - \tilde{C}_{yy} C_{yy}^{-1} \tilde{C}_{yy} \right) \\
-p_{yy} & -C_{yy}^{-1} \tilde{C}_{yy} C_{yy}^{-1} \tilde{C}_{yy}
\end{bmatrix}^{-1}
\]

where \( P_{yy} \) is the Schur complement. Note, in this special case the block matrices of (4.58) and (4.61) are just conjugates of each other. That is why this particular form of the matrix inversion lemma has been used. When inserting (4.62) in (4.56) the block matrices \( W_1 \) and \( W_2 \) are

\[
W_1 = \left( C_{xy} - \tilde{C}_{xy} C_{yy}^{-1} \tilde{C}_{yy} \right) P_{yy}^{-1} \\
W_2 = \left( \tilde{C}_{xy} - C_{xy} C_{yy}^{-1} \tilde{C}_{yy} \right) P_{yy}^{-*}
\]

In general, \( W_1 \neq W_2^* \). Finally, the WLMMSE is denoted by

\[
\hat{x}_{wl} = \left( C_{xy} - \tilde{C}_{xy} C_{yy}^{-1} \tilde{C}_{yy} \right) P_{yy}^{-1} y + \left( \tilde{C}_{xy} - C_{xy} C_{yy}^{-1} \tilde{C}_{yy} \right) P_{yy}^{-*} y^*
\]

As this chapter is about the basics of WLMMSE estimation, a further approach for deriving the WLMMSE is shown. The reader may be familiar with the orthogonality principle, used to derive the LMMSE in the real case [19]. This concept can also be used in the complex case [2, 27]. The orthogonality principle uses the geometrical concept from Hilbert space. See Section 2.5 for a brief review. Again the estimator model is given as in (4.52). \( \tilde{x} \) and \( y \) must have zero-mean, or one may use \( \tilde{x}' = \tilde{x} - E(\tilde{x}) \) and \( y' = y - E(y) \) respectively. Geometrically spoken, the estimator is a linear combination (function) of the observed data \( y \). Thus \( \tilde{x} \) is lying in a subspace spanned by \( y \). The task is to find the minimum error vector between \( \tilde{x} \) and \( \hat{x} = Wy \). As the estimator is a linear combination of the measurements, all components of \( \tilde{x} \) must lie in \( H(y) \), where \( H(y) \) is a subspace spanned by \( y \). This concept is illustrated in Figure 4.1. The error vector \( e = x - \hat{x} \) is minimized if it is orthogonal to \( H(y) \). See Figure 4.2 for a visual explanation of this concept. It can be seen that the error vector \( e' \) has a higher magnitude than \( e \). This is true for all nonorthogonal error vectors. Minimizing the MSE is nothing else but minimizing the squared magnitude of the error vector. As already explained \( E(|e|)^2 = E(|x - \hat{x}|)^2 \) reaches its minimum if \( e \perp \hat{x} \), or \( E(e y^H) = 0 \). With the orthogonality principle the ALMMESE is derived as follows

\[
0 = E(e y^H) \\
E((x - \hat{x}) y^H) \\
E((x - Wy) y^H) \\
E(x y^H - Wy y^H) \\
E(x y^H) - W E(y y^H)
\]
The expectation operator is on \( \mathbf{x} \) and \( \mathbf{y} \). (4.66) yield the normal equations and the Wiener-Hopf solution for the augmented representation

\[
0 = \mathbf{C}_{xy} - \mathbf{W} \mathbf{C}_{yy} \tag{4.67}
\]
\[
\mathbf{W} = \frac{\mathbf{C}_{xy} \mathbf{C}_{yy}^{-1}}{\vert_1} \tag{4.68}
\]

These results are the very same as (4.47) and (4.48). It shows that the orthogonality principle is just another, but maybe simpler and more intuitive approach to derive the ALMMSE and hence the WLMMSE. Note, replacing the augmented vectors with the normal complex vectors yields the LMMSE. When further inspecting the orthogonality between the augmented error vector and the augmented data vector

\[
E(\mathbf{e} \mathbf{y}^H) = E\left(\begin{bmatrix} \mathbf{e} \\ \mathbf{e}^* \end{bmatrix} [\mathbf{y}^H \mathbf{y}^T]\right) = \mathbf{0},
\]

(4.69)

it can be seen that \((\mathbf{x} - \hat{\mathbf{x}}) \perp \mathbf{y}\) and \((\mathbf{x} - \hat{\mathbf{x}}) \perp \mathbf{y}^*\). This guarantees that if \(\mathbf{y}\) is improper, these properties are also considered in the ALMMSE and WLMMSE approach.

![Subspace $H(y)$ spanned by $\mathbf{y}$](image)

Figure 4.1: Subspace $H(y)$ spanned by $\mathbf{y}$

Given the augmented error \( \mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \), the error covariance is also an augmented matrix, denoted as

\[
\mathbf{C}_{ee} = E(\mathbf{e} \mathbf{e}^H)
\]
\[
= \begin{bmatrix}
\mathbf{C}_{ee} & \mathbf{C}_{ee}^* \\
\mathbf{C}_{ee}^* & \mathbf{C}_{ee}^*
\end{bmatrix}
\]

(4.70)

Whereas the north-west block matrix is the estimation error covariance matrix, with the BMSEs on the main diagonal. The overall estimation error is given by the trace of the
error covariance matrix, or by half of the trace of the augmented one \([2]\). Mathematically expressed this is \( E\left(||e||^2\right) = \text{tr}\left(C_{ee}\right) = \frac{1}{2} \text{tr}\left(C_{ee}^{'\prime}\right) \). Inserting the augmented estimator model into (4.70) yields
\[
C_{ee} = C_{xx} - C_{xy} C_{yy}^{-1} C_{yx},
\] (4.71)
which has a similar form as the LMMSE error covariance matrix. It may now be of great interest to have a closed form solution of \( C_{ee} \). From (4.71) it can be seen that the derivation problem is very similar to the WLMMSE derivation. Again, the augmented autocovariance matrix must be inverted. During the following derivation the inverse of \( C_{yy} \) will be denoted as
\[
C_{yy}^{-1} = \begin{bmatrix} E & F \\ F^* & E^* \end{bmatrix},
\] (4.72)
with \( E = P_{yy}^{-1} \) and \( F = -C_{yy}^{-1} \tilde{C}_{yy} P_{yy}^{-1} \). Refer to (4.62) for the complete description. With
that, the augmented error covariance matrix is given by

\[
\mathbf{C}_{ee} = \begin{bmatrix}
C_{xx} & \tilde{C}_{xx} \\
\tilde{C}_{xx}^* & C_{xx}^*
\end{bmatrix} - \begin{bmatrix}
C_{xy} & \tilde{C}_{xy} \\
\tilde{C}_{xy}^* & C_{xy}^*
\end{bmatrix} \begin{bmatrix}
E & F \\
F^* & E^*
\end{bmatrix} \begin{bmatrix}
C_{yx} & \tilde{C}_{yx} \\
\tilde{C}_{yx}^* & C_{yx}^*
\end{bmatrix}
\]

\[
= \begin{bmatrix}
C_{xx} & \tilde{C}_{xx} \\
\tilde{C}_{xx}^* & C_{xx}^*
\end{bmatrix} - \begin{bmatrix}
C_{xy} E + \tilde{C}_{xy} F^* & C_{xy} F + \tilde{C}_{xy} E^* \\
\tilde{C}_{xy}^* E + C_{xy}^* F & \tilde{C}_{xy}^* F + C_{xy}^* E^*
\end{bmatrix} \begin{bmatrix}
C_{yx} & \tilde{C}_{yx} \\
\tilde{C}_{yx}^* & C_{yx}^*
\end{bmatrix}
\]

\[
= \begin{bmatrix}
C_{xx} & \tilde{C}_{xx} \\
\tilde{C}_{xx}^* & C_{xx}^*
\end{bmatrix} - \begin{bmatrix}
C_{xy} E + \tilde{C}_{xy} F^* \tilde{C}_{yx} + (C_{xy} F + \tilde{C}_{xy} E^*) C_{yx}^* \\
\tilde{C}_{xy}^* E + C_{xy}^* F \tilde{C}_{yx} + (\tilde{C}_{xy}^* F + C_{xy}^* E^*) C_{yx}^*
\end{bmatrix}
\]

(4.73)

The error covariance matrix of the WLMMSE is given by the north-west block of (4.73). Resubstituting \( E \) and \( F \) yields

\[
\mathbf{C}_{ee} = C_{xx} - \left( C_{xy} P^{-1}_{yy} - \tilde{C}_{xy} C^{-*}_{yy} \tilde{C}_{yy} P^{-1}_{yy} \right) C_{yx} +
\]

\[
\left( \tilde{C}_{xy} P^{-1}_{yy} - C_{xy} C^{-1}_{yy} \tilde{C}_{yy} P^{-*}_{yy} \right) \tilde{C}_{yx}
\]

\[
= C_{xx} - \left( C_{xy} - \tilde{C}_{xy} C^{-*}_{yy} \tilde{C}_{yy} \right) P^{-1}_{yy} C_{yx} +
\]

\[
\left( \tilde{C}_{xy} - C_{xy} C^{-1}_{yy} \tilde{C}_{yy} \right) P^{-*}_{yy} \tilde{C}_{yx}
\]

(4.74)

Note, \( W_1 \) and \( W_2 \) are equivalent to (4.63) and (4.64). In contrast to the LMMSE, (4.74) considers the pseudo-covariances, resulting in a better BMSE. Proof is shown in Section 4.1.3. Finally the BMSE of each particular estimator is given by

\[
\text{BMSE}_{wl}(\hat{x}_n) = [C_{xx}]_{nn} - \sum_{m=1}^{N} [W_1]_{nm} [C_{yx}]_{mn} - \sum_{m=1}^{N} [W_2]_{nm} [\tilde{C}_{yx}]_{mn},
\]

(4.75)

where \( n = 1, 2, \ldots, p \).

The following subsections cover three special cases. One introduces proper signals, the second one real-valued parameter estimation, and the last one shows the WLMMSE for measurements being strictly linear dependent on the parameters.

**WLMMSE for Proper Signals**

Here, \( x \) and \( y \) are assumed to have zero-mean, are proper and cross-proper. Consequently all pseudo-covariance matrices vanish, i.e. \( \tilde{C}_{yy} = 0, \tilde{C}_{xy} = 0, \tilde{C}_{yx} = 0 \). With that, (4.63) and (4.64) reduce to

\[
W_1 = \left( C_{xy} - \tilde{C}_{xy} C^{-*}_{yy} \tilde{C}_{yy} \right) \left( C_{yy} - \tilde{C}_{yy} C^{-*}_{yy} \tilde{C}_{yy} \right)^{-1}
\]

\[
= (C_{xy} - 0 C^{-*}_{yy} 0) (C_{yy} - 0 C^{-*}_{yy} 0)^{-1}
\]

\[
= C_{xy} C^{-1}_{yy}
\]

(4.76)
\[ \mathbf{W}_2 = \left( \tilde{C}_{xy} - C_{xy} C^{-1}_{yy} \tilde{C}_{yy} \right) \left( \tilde{C}_{yy} - C_{yy} \tilde{C}_{yy} \right)^{-1} \]
\[ = (0 - C_{xy} C^{-1}_{yy} 0) \left( C_{yy} - 0 C_{yy} 0 \right)^{-1} \]
\[ = (0) C_{yy}^{-*} \]
\[ = 0 \]  \hspace{1cm} (4.77)  

Inserting (4.76) and (4.77) in (4.53) yields
\[ \hat{x}_{wl} = \mathbf{W}_1 \mathbf{y} + 0 \mathbf{y}^* \]
\[ = C_{xy} C^{-1}_{yy} \mathbf{y} \]  \hspace{1cm} (4.78)  

This can also be proved by the orthogonality principle, see appendix A.5.

Comparing (4.78) with (4.33) shows that the WLMMSE reduces to the LMMSE if and only if \( \mathbf{x} \) and \( \mathbf{y} \) are proper and cross-proper. This is a very important result, showing that for proper signals the LMMSE is already the optimal linear estimator.

**WLMMSE for Real-Valued Signals**

Here it is shown what happens to the closed form solution of the WLMMSE if the parameters to be estimated are real-valued. For instance, consider a BPSK modulated signal, shown in Figure 3.6a. There, the imaginary part is zero and from Section 3.3.2 it is already known that the data is improper. Improper statistics are not considered in the LMMSE and thus the WLMMSE would be the optimal choice. Assume that \( \mathbf{x} \) is now a real-valued \( p \)-dimensional vector, whose particular realization must be estimated. However, \( \mathbf{y} \) is still an \( N \)-dimensional complex-valued vector. Once again, the WLMMSE estimator model (4.53) is used, with \( \mathbf{W}_1 \) and \( \mathbf{W}_2 \) as in (4.63) and (4.64). With these assumptions the conjugate cross-covariance corresponds to the pseudo-cross-covariance, \( C_{xy}^* = \tilde{C}_{xy} \).

**Proof.** Given \( \mathbf{x} \in \mathbb{R}^p \) and \( \mathbf{y} \in \mathbb{C}^N \) the pseudo-cross-covariance becomes
\[ \tilde{C}_{xy} = E (\mathbf{x} \mathbf{y}^T) \]
\[ = E (\mathbf{x} \mathbf{y}^H)^* \]
\[ = C_{xy}^* \]  \hspace{1cm} (4.79)  

Inserting this properties into (4.64) yields
\[ \mathbf{W}_2 = \left( \tilde{C}_{xy} - C_{xy} C^{-1}_{yy} \tilde{C}_{yy} \right) \mathbf{P}_{yy}^{-*} \]
\[ = \left( C_{xy}^* - \tilde{C}_{xy} C^{-1}_{yy} \tilde{C}_{yy} \right) \mathbf{P}_{yy}^{-*} \]
\[ = \left[ \left( C_{xy} - \tilde{C}_{xy} C_{yy} \tilde{C}_{yy} \right) \mathbf{P}_{yy}^{-1} \right]^* \]
\[ = \mathbf{W}_1, \]  \hspace{1cm} (4.80)  

Thus, $W_2 = W_1^*$ for $x \in \mathbb{R}^p$. Further, the WLMMSE reduces to

$$\hat{x}_{wl} = W_1 y + W_1^* y^* = W_1 y + (W_1 y)^* = 2 \Re(W_1 y).$$

(4.81)

This shows that the WLMMSE is initially real-valued, $\hat{x}_{wl} \in \mathbb{R}^p$, if $x \in \mathbb{R}^p$. In contrast, the LMMSE (4.33) is complex-valued in general.

Figure 4.3 shows a block model of an IQ-receiver. After demodulation the signal is represented in baseband with IQ-paths. Then, after sampling, the particular realization of the data is estimated in the equalizer block. If the LMMSE is used, an additional block would be needed to limit the (general) complex result of the estimation algorithm to a real-valued output. Instead, when using the WLLMSE estimator the additional $\Re(\cdot)$ block can be spared. Of course, this may not be crucial, but it is still a nice feature of the WLMMSE. Figure 4.4 shows the scatter plot of an estimated BPSK signal. The WLMMSE estimates only consist of real parts, while the LMMSE estimates are in general complex-valued.

Figure 4.3: IQ-receiver block diagram. The WLMMSE makes the additional $\Re(\cdot)$ redundant.

**WLMMSE for Linear Data Model**

As for the LMMSE it is also worth to look how the WLMMSE behaves under the linear model assumption. Like before, assume the measurements are strictly linear in the parameters (4.35). To get the closed form solution of the WLMMSE the augmented representation
is used. The WLMMSE is then deduced from the ALMMSE. The augmented data model is

\[
\mathbf{y} = \begin{bmatrix} y \\ y^* \end{bmatrix} = \begin{bmatrix} \mathbf{H} \mathbf{x} \\ \mathbf{H}^* \mathbf{x}^* \end{bmatrix} + \begin{bmatrix} \mathbf{n} \\ \mathbf{n}^* \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^* \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix} + \begin{bmatrix} \mathbf{n} \\ \mathbf{n}^* \end{bmatrix}
\]

(4.82)

\(\mathbf{H}\) is an augmented matrix with \(\mathbf{H}\) the \(N \times p\)-dimensional observation matrix from the linear model and \(\mathbf{0}\) a \(N \times p\)-dimensional zero matrix. The ALMMSE is given in (4.54), where now the covariance matrices can be calculated using the linear model (4.82). Hence,
the augmented autocovariance matrix becomes

\[ C_{yy} = E \left[ y y^H \right] \]

\[ = E \left[ (H x + n) (H x + n)^H \right] \]

\[ = H E (x x^H) H^H + H E (x n^H) + E (n x^H) H^H + E (n n^H) \]

\[ = H C_{xx} H^H + C_{nn} \]

\[ = \begin{bmatrix} H & 0 \\ 0 & H^* \end{bmatrix} \begin{bmatrix} C_{xx} & C_{xx}^* \\ C_{xx}^* & C_{xx}^* \end{bmatrix} \begin{bmatrix} H^H & 0^T \\ 0^T & H^T \end{bmatrix} + \begin{bmatrix} C_{nn} & C_{nn} \\ C_{nn}^* & C_{nn}^* \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & H C_{xx} H^T \\ H^* C_{xx}^* H^H & H^* C_{xx}^* H^T \end{bmatrix} + \begin{bmatrix} C_{nn} & \tilde{C}_{nn} \\ C_{nn}^* & \tilde{C}_{nn}^* \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & H C_{xx} \tilde{H} \\ \tilde{H}^* C_{xx}^* H^H & \tilde{H}^* C_{xx}^* H^T \end{bmatrix} + \begin{bmatrix} C_{nn} & \tilde{C}_{nn} \\ C_{nn}^* & \tilde{C}_{nn}^* \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & H C_{xx} \tilde{H} \\ \tilde{H}^* C_{xx}^* H^H & \tilde{H}^* C_{xx}^* H^T \end{bmatrix} + \begin{bmatrix} C_{nn} & \tilde{C}_{nn} \\ C_{nn}^* & \tilde{C}_{nn}^* \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & \tilde{C}_{xx} H^T \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & \tilde{C}_{xx} H^T \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & \tilde{C}_{xx} H^T \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} H C_{xx} H^H & \tilde{C}_{xx} H^T \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

The autocovariance and the pseudo-autocovariance matrices are given by

\[ C_{yy} = H C_{xx} H^H + C_{nn} \]

and

\[ \tilde{C}_{yy} = H \tilde{C}_{xx} H^T + \tilde{C}_{nn} \]

The concept structure of the covariance matrices looks similar to the augmented ones, see (4.83). With (2.36), the augmented results can be transformed into equivalent real representations, see Section 2.2. Next, the augmented cross-covariance matrix is determined in the same way as above

\[ C_{xy} = E \left[ x y^H \right] \]

\[ = E \left[ x (H x + n)^H \right] \]

\[ = E (x x^H) H^H + E (x n^H) \]

\[ = C_{xx} H^H \]

\[ = \begin{bmatrix} C_{xx} & C_{xx}^* \\ C_{xx}^* & C_{xx}^* \end{bmatrix} \begin{bmatrix} H^H & 0^T \\ 0^T & H^T \end{bmatrix} \]

\[ = \begin{bmatrix} C_{xx} H^H & C_{xx} \tilde{H} \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} C_{xx} H^H & C_{xx} \tilde{H} \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} C_{xx} H^H & C_{xx} \tilde{H} \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

\[ = \begin{bmatrix} C_{xx} H^H & C_{xx} \tilde{H} \\ \tilde{C}_{xx}^* H^H & C_{xx}^* H^T \end{bmatrix} \]

The form of the augmented cross-covariance matrix yields

\[ C_{xy} = C_{xx} H^H \]

and

\[ \tilde{C}_{xy} = \tilde{C}_{xx} H^T \]

Inserting (4.83) and (4.85) into the ALMMSE results in

\[ \hat{x} = C_{xx} H^H (H C_{xx} H^H + C_{nn})^{-1} y \]

While the ALMMSE yields the same form as the LMMSE in the real case, the WLMMSE yields quite a long expression. Inserting the derived covariance and pseudo-covariance ma-
trices in (4.65) yields to
\[
\hat{x}_{wl} = \left[ C_{xx} H^T - \hat{C}_{xx} H^T \left( H^* C_{xx}^* H^T + C_{nn}^* \right)^{-1} \left( H^* \tilde{C}_{xx}^* H + \tilde{C}_{nn}^* \right) \right] \cdot \left[ H C_{xx} H + C_{nn} - \left( H \hat{C}_{xx} H^T + \tilde{C}_{nn} \right) \left( H^* C_{xx}^* H^T + C_{nn}^* \right)^{-1} \cdot \left( H^* \tilde{C}_{xx}^* H + \tilde{C}_{nn}^* \right) \right]^{-1} y + \\
\left[ \hat{C}_{xx} H^T - C_{xx} H^T \left( H C_{xx} H^* + C_{nn} \right)^{-1} \left( H \hat{C}_{xx} H^* + \tilde{C}_{nn} \right) \right] \cdot \left[ H^* C_{xx}^* H^T + C_{nn}^* - \left( H^* \tilde{C}_{xx}^* H + \tilde{C}_{nn}^* \right) \left( H C_{xx} H^* + C_{nn} \right)^{-1} \cdot \left( H \hat{C}_{xx} H^* + \tilde{C}_{nn} \right) \right]^{-1} y^* .
\] (4.88)

### 4.1.3 Comparison of LMMSE and WLMMSE

The most obvious result, when comparing LMMSE to WLMMSE, is the high calculation effort of the latter one. While the WLMMSE requires several matrix multiplications and inversions, the LMMSE goes with just one inversion and three vector-matrix multiplications. A quick view on (4.40) and (4.88) illustrates the statement very roughly. On the other hand, the LMMSE does not consider any pseudo characteristics. Hence, properness is a must have for the LMMSE to be the best linear estimator. From [27] and [28] the difference of the MSE between the two estimators is given by the following equation
\[
\Delta e = \left( \hat{C}_{yx}^* - \hat{C}_{yy}^* C_{yy}^{-1} C_{yx} \right)^H \left( C_{yy}^* - \hat{C}_{yy}^* C_{yy}^{-1} \hat{C}_{yy} \right)^{-1} \left( C_{yx}^* - \hat{C}_{yx}^* C_{yy}^{-1} C_{yx} \right) ,
\] (4.89)

where \( \Delta e \) can be seen as the difference between the WLMMSE and the LMMSE cost function \( \Delta e = J_{wl} - J \). The differential error is \( \Delta e \geq 0 \). It gets zero if \( x \) and \( y \) are proper and cross-proper. Because then, all pseudo matrices vanish and the WLMMSE reduces to the LMMSE (4.78).

One further property is the similar look of the LMMSE and the ALMMSE. The derivation steps of both, using the CCR-Calculus, are identical. Anyhow, these results must not be mixed. They are completely different, refer to the final solutions of the LMMSE (4.33) and the WLMMSE (4.65). Again, only with assumed properness, these two are really equivalent.

However, the WLMMSE is just the upper half of the ALMMSE, i.e.
\[
\hat{x} = \left[ \hat{x}_1 \hat{x}_2 \cdots \hat{x}_p \hat{x}_1^* \hat{x}_2^* \cdots \hat{x}_p^* \right]^T .
\] (4.90)

Because of that, the ALMMSE can easily be implement in MATLAB simulations. The WLMMSE is then given by cutting off the lower half of \( \hat{x} \). This results in a better readable source code, but it must be considered that the ALMMSE requires more calculation effort.

**Example 4.1.2.** The following example points out the differences between the LMMSE and the WLMMSE for prediction of a data stream in an AWGN channel. The source code is attached in the appendix B.2. The measurements are modeled as (4.35). As above, \( H \)
is the observation/channel matrix, x the transmitted data stream and n is additive noise. The following settings are used in the simulation:

\[
\begin{align*}
\textbf{d} &\sim \mathcal{CN} \left( \mathbf{0}, C_{dd} = \sigma^2_d \mathbf{I}, \hat{C}_{dd} \right) \in \mathbb{C}^p \\
\textbf{n} &\sim \mathcal{CN} \left( \mathbf{0}, C_{nn} = \mathbf{I}, \hat{C}_{nn} = \mathbf{0} \right) \in \mathbb{C}^N \\
\textbf{H} &= \mathbf{I} \in \mathbb{R}^{N \times N}
\end{align*}
\] (4.91)

(4.92)

(4.93)

With \( p = N \) and hence \( \textbf{H} = \mathbf{I} \), the channel reduces to an additive noise channel. Further, as \( C_{nn} = \mathbf{I} \) and \( \hat{C}_{nn} = \mathbf{0} \), \( n \) is complex white Gaussian noise (CWGN) noise. The real part is completely independent of the imaginary part, and both have variance \( \sigma^2_n \). The pseudo-covariance matrix of \( d \) is variable. While the variances \( \sigma^2_d \) and \( \sigma^2_i \) do not change, the correlation between real and imaginary part is linearly increased (r-loop), denoted as \( \rho_d \) in the MATLAB simulation. This ensures that \( C_{dd} \) keeps constant and \( \hat{C}_{dd} \) changes with \( \rho_d \). The covariance matrices are all diagonal ones. Hence, there are no time dependent correlations. Data and noise are therefore both white random processes, refer to Section 3.5. The simulation uses \( N = 100 \) measurements and \( p = N \) parameters, which have to be estimated. At one specific correlation coefficient, \( M = 100 \) data bursts are transmitted. In the source code, the LMMSE estimate is denoted as \( \textbf{x} \) and the WLMMSE estimate as \( \textbf{x}_{WL} \). Wheras, the latter is calculated from the ALMMSE \( \textbf{ax} \). At last, the error covariances and the BMSEs are calculated. The plotted MSE is the sum of all BMSE as defined in (4.1). Figure 4.5 shows the MSEs of the simulation on a logarithmic scale. The LMMSE is outperformed by the WLMMSE as the correlation between real and imaginary data stream increases. A further interesting result is the value of the MSE of the LMMSE. While the MSE of the WLMMSE reduces with increasing correlation, the MSE of the LMMSE still fluctuates around the same level. This results from the neglected pseudo-correlations in the estimator equation. In other words, as \( C_{dd} \) keeps untouched by \( \rho_d \), the MSE of the LMMSE do not change with the correlation coefficient.

Looking at Figure 4.5, the results indicate that there is no large benefit when using the WLMMSE. However, this changes if the channel is no longer a strictly linear one, but a WL channel. From Section 2.3, a transformation on the equivalent real variables generally results in a WL transformation like in (2.52). Thus, the model above is extended to \( \textbf{y} = \textbf{H} \textbf{d} + \textbf{H}_2 \textbf{d}^* + \textbf{n} \). The measurements are now dependent on \( \textbf{d} \) and \( \textbf{d}^* \). The channel matrix for the conjugate data is set to \( \textbf{H}_2 = \frac{1}{2} \mathbf{I} \). This is the very reason why the WLMMSE has been introduced [27]. In contrast to the LMMSE, it now shows its full potential and considers not only complete SO statistics, but also the WL channel. In Figure 4.6, the advantages of the WLMMSE are obvious. Again, the MSE of the LMMSE does not change with varying correlation coefficient. Here, the WLMMSE outperforms the LMMSE even if there are no correlations in the data stream. This results from the imprperness of \( \textbf{y} \) because of the WL channel.

It may also be of great interest how much calculation effort is needed to calculate the LMMSE and the ALMMSE. From the simulation, 10000 runs, the average calculation times are \( t_{LMMSE} = 0.0015 \) s and \( t_{ALMMSE} = 0.0052 \) s. According to the average simulation time, the ALMMSE requires \( \sim 3.5 \) times longer than the LMMSE.
Figure 4.5: $\text{MSE} = E[e^H e]$ of WLMMSE and LMMSE for white Gaussian data over strictly linear channel with additive CWGN.
Figure 4.6: $\text{MSE} = E[e^H e]$ of WLMMSE and LMMSE for white Gaussian data over widely linear channel with additive CWGN.
4.2 Classical Estimators for Proper Signals

In the classical approach, the parameter to be estimated is assumed to be deterministic, but unknown. In this section, a rather short overview of some classical estimators is given. The following results and statements are based on [19]. The estimators mentioned are for proper signals only. Complex data vectors are restricted to have the complex multivariate Gaussian pdf (3.71):

\[
p(y; x) = \frac{1}{\pi^N \det(C_{yy})} \exp \left[ - (y - \mu_y)^H C_{yy}^{-1} (y - \mu_y) \right]
\]

(4.94)

Where \( x \) is the parameter vector, \( y \) the data vector with \( \mu_y \) its mean and \( C_{yy} \) as Hermitian covariance matrix. Although not especially mentioned, mean and covariance matrix of \( y \) depend on the parameter \( x \). The pdf can also be rewritten in dependence of the equivalent real vector \( x_R \), i.e. \( p(y; x) \equiv p(y; x_R) \). The following chapters restricts the representations to complex parameters \( x \) and the equivalent real vectors \( x_R \). Refer to [2] for classical estimators of improper signals.

4.2.1 Minimum Variance Unbiased (MVU) Estimator

The task is to search for the estimator of \( x \) which has minimum variance and is unbiased. Unbiased assumes that the expected value of the estimator is the true value. Further the variance shall be minimum. These two conditions are mathematically expressed as:

\[
E(x_R) = x_R \quad (4.95)
\]

\[
\text{var}(x_R) \rightarrow \text{min}. \quad (4.96)
\]

The Cramer-Rao Lower Bound (CRLB) provides a way to calculate a lower bound on the variance of all possible unbiased estimators. In the real case, i.e. \( x \in \mathbb{R} \), it can be shown that [19]

\[
\text{var}(\hat{x}) \geq \frac{1}{-E \left\{ \frac{\partial^2 \ln[p(y;x)]}{\partial x^2} \right\}} = \frac{1}{\mathcal{I}(x)}, \quad (4.97)
\]

where \( \mathcal{I}(x) \) is the Fisher information matrix [19]. It is sometimes possible to deduce the MVU estimator from the CRLB. Therefore, the first derivative of the logarithm of \( p(y,x) \) must be in the form of

\[
\frac{\partial \ln p(y;x)}{\partial x} = \mathcal{I}(x) (g(y) - x) \quad (4.98)
\]

If this form can be achieved, then \( g(y) = \hat{x} \) is the MVU estimator for \( x \) that attains the bound for all \( x \). Such an estimator is called efficient.

This concept can be altered to the complex case. There, \( x \) is no longer a real, but a complex-valued parameter/vector, defined as \( x = x_r + j x_i \). It can be shown that if the first derivative of \( \ln[p_x(y,x)] \) with respect to \( x^* \) can be written in the form of

\[
\frac{\partial \ln p(y;x)}{\partial x^*} = \mathcal{I}(x) \left( \frac{g(y) - x}{x} \right), \quad (4.99)
\]
then \( g(y) = \hat{x} \) is the efficient estimator for \( x \). The fundamental concept to get this result is the special form of the Fisher information matrix of the equivalent real vector \( x_R \). It can be denoted by

\[
\mathcal{I}(x_R) = 2 \begin{bmatrix} E & -F \\ F & E \end{bmatrix}.
\] (4.100)

This form can only be achieved by proper signals, compare to the proper real covariance (3.52). With (4.100) the Fisher matrix of the complex parameter is reduced to

\[
\mathcal{I}(x) = (E + jF).
\] (4.101)

where

\[
\begin{pmatrix} E & jF \\ jF & E \end{pmatrix}.
\] (4.102)

If \( y \) is improper and not jointly Gaussian distributed with \( x \), then the CRLB may be very hard, or even impossible, to find. To overcome this, a linearization may be used to get a suboptimal estimator.

### 4.2.2 Complex Linear Model and BLUE

As mentioned, the determination of the CRLB in its special form to find the MVU is very difficult in general. But if the data model is linear, the MVU estimator can be easily found. The complex linear model has the form

\[
y = Hx + n.
\] (4.102)

\( H \) is a known \( N \times p \) complex observation matrix with \( N > p \) and full rank, \( x \) is a complex \( p \times 1 \) parameter vector to be estimated and \( n \) is \( N \times 1 \) complex noise vector with the pdf \( n \sim \mathcal{CN}(0, C_{nn}, \tilde{C}_{nn} = 0) \). As already explained in Section 3.4, strictly linear transformations of proper complex Gaussian random vectors yield again proper complex Gaussian distributed random vectors. Thus \( y \sim \mathcal{CN}(E(y), C_{yy}, \tilde{C}_{yy} = 0) \), where it can be shown that \( E(y) = Hx \) and \( C_{yy} = C_{nn} \). If this relations are used, the complex multivariate Gaussian pdf for \( y \) can be denoted as in (3.71). When this pdf is inserted into (4.99), the complex gradient becomes [19]

\[
\frac{\partial \ln p(y; x)}{\partial x^*} = -\frac{\partial (y - Hx)^H C_{yy}^{-1} (y - Hx)}{\partial x^*} = H^H C_{yy}^{-1} (y - Hx).
\] (4.103)

After rearranging (4.103) to the special form in (4.99) the gradient is

\[
\frac{\partial \ln p(y; x)}{\partial x^*} = H^H C_{yy}^{-1} \begin{bmatrix} (H^H C_{yy}^{-1} H)^{-1} H^H C_{yy}^{-1} y - x \end{bmatrix}.
\] (4.104)

Hence the MVU estimator of \( x \), which in this case is also efficient, is given by

\[
\hat{x} = (H^H C_{yy}^{-1} H)^{-1} H^H C_{yy}^{-1} y.
\] (4.105)

If the noise and hence \( y \) is not Gaussian, (4.105) represents the complex equivalent to the best linear unbiased estimator (BLUE). With complex variance (3.12), the BLUE minimizes the sum of the variances of \( x_r \) and \( x_i \) [19].
4.2.3 Least Squares (LS) Estimation

The least squares estimator tries to find the estimator with the minimum error in the LS sense. The estimator can be found by taking the derivative of the corresponding cost function and setting the result to zero. This subsection shows the advantages of the CR-Calculus compared to $\mathbb{R}$ differentiation approach. If the least squares error (LSE) of a complex amplitude $x = x_r + j x_i$ has to be found, the corresponding cost function is

$$ J(x) = \sum_{n=0}^{N-1} |y[n] - x s[n]|^2, \quad (4.106) $$

where $y[n]$, $x$, and $s[n]$ are complex. Note, that in contrast to the MSE, there is no expectation operator in (4.106). Further, the cost function is a real-valued function. The straight forward approach is to decompose the complex signals and parameters into their real and imaginary part. Hence the new cost function $J'$ can be denoted by

$$ J'(x_r, x_i) = \sum_{n=0}^{N-1} |y_r[n] + j y_i[n] - (x_r + j x_i) (s_r[n] + j s_i[n])|^2 $$

$$ = \sum_{n=0}^{N-1} \frac{(y_r[n] - x_r s_r[n] + x_i s_i[n])^2 + (y_i[n] - x_r s_i[n] - x_i s_r[n])^2}{2(J)^2} \quad (4.107) $$

A linear LS problem always yields a quadratic minimization problem [19]. If the vector notation is used, such that $y_r = [y_r[0] \ y_r[1] \ldots \ y_r[N-1]]^T$, $y_i = [y_i[0] \ y_i[1] \ldots \ y_i[N-1]]^T$, $s_r = [s_r[0] \ s_r[1] \ldots \ s_r[N-1]]^T$ and $s_i = [s_i[0] \ s_i[1] \ldots \ s_i[N-1]]^T$, the cost function in (4.107) can be rewritten as

$$ J'(x_r, x_i) = (y_r - x_r s_r + x_i s_i)^T (y_r - x_r s_r + x_i s_i) + (y_i - x_r s_i - x_i s_r)^T (y_i - x_r s_i - x_i s_r). \quad (4.108) $$

For a more convenient calculation a substitution of the variables $s_1 = [s_r - s_i]$, $s_2 = [s_i \ s_r]$ and $x_R = [x_r \ x_i]^T$ is performed. If the substitution is done at (4.109), $J'$ becomes

$$ J'(x_R) = (y_r - s_1 x_R)^T (y_r - s_1 x_R) + (y_i - s_2 x_R)^T (y_i - s_2 x_R) \quad (4.110) $$

After multiplying, the gradient with respect to $x_R$ is calculated and thus

$$ \frac{\partial J'}{\partial x_R} = -2 s_1^T y_r + 2 s_1^T s_1 x_R - 2 s_2^T y_i + 2 s_2^T s_2 x_R. \quad (4.111) $$

To get $x_R$, which minimizes $J'$, equation (4.111) is set to zero and solved with respect to $x_R$:

$$ \hat{x}_R = (s_1^T s_1 + s_2^T s_2)^{-1} (s_1^T y_r + s_2^T y_i). \quad (4.112) $$

Re-substitution of $s_1$ and $s_2$ yields the estimator

$$ \hat{x}_R = \left[ \begin{array}{c} \hat{x}_r \\ \hat{x}_i \end{array} \right] = \left[ \begin{array}{c} s_1^T y_r + s_2^T y_i \\ s_1^T s_1 + s_2^T s_2 \end{array} \right] \left[ \begin{array}{c} \hat{x}_r \\ \hat{x}_i \end{array} \right]. \quad (4.113) $$
If \( \hat{x}_R \) is rewritten in its complex form \( \hat{x} = \hat{x}_r + j \hat{x}_i \) with \( y = y_r + j y_i \) and \( s = s_r + j s_i \), the complex estimator finally becomes

\[
\hat{x} = \frac{\sum_{n=0}^{N-1} y[n] s^*[n]}{\sum_{n=0}^{N-1} |s[n]|^2}. \tag{4.114}
\]

The direct approach, to write the complex variable as vector of real parameters, calculate the LSE and then substitute back, is a very tedious and complicated method. A more convenient and efficient way is to use the \( \mathbb{C} \mathbb{R} \)-Calculus. It is possible to directly minimize the cost function (4.106) with the help of the complex gradient (2.12).

\[
\frac{\partial J}{\partial x} = \frac{\partial }{\partial x} \sum_{n=0}^{N-1} |y[n] - x s[n]|^2 = \sum_{n=0}^{N-1} \frac{\partial }{\partial x} \left( |y[n]|^2 - y[n] s^*[n] - x s[n] y^*[n] + x^* x |s[n]|^2 \right) \tag{4.115}
\]

Using (2.15) yields

\[
\frac{\partial J}{\partial x} = \sum_{n=0}^{N-1} \left( s[n] y^*[n] + x^* |s[n]|^2 \right). \tag{4.116}
\]

If this result is set to zero and the equation is solved for \( x \), it yields the same estimator \( \hat{x} \) as in (4.114).

Now assume the cost function \( J \) is an Hermitian function, like the exponent of a complex multivariate Gaussian pdf, then it becomes

\[
J = (y - H x)^H W (y - H x) \tag{4.117}
\]

Taking the complex gradient with multivariate \( \mathbb{C} \mathbb{R} \)-Calculus rules, Section 2.1.2, and setting the result to zero, produces the estimator

\[
\hat{x} = (H^H W H)^{-1} H^H W y. \tag{4.118}
\]

With \( W = C_{yy}^{-1} \) the weighted LS yields the very same result as the BLUE (4.105).
Chapter 5

Applications for Complex-Valued Data Estimation

5.1 Estimation of a Constant Complex Envelope

This example shows the difference between parameter estimation out of a bandpass signal or from its complex envelope. In practice, it is not usual to estimate the bandpass signal directly, but it is a good example to intuitively show the difference between real and complex parameter estimation. Figure 5.1 shows the block diagram of the assumed transceiver, separated into four parts. Here, the classical approach is used, see Section 4.2. The transmitted in-phase and quadrature components are assumed to be deterministic, but unknown. Further, signal modulation and sampling are ideal. The used lowpass filter possesses no gain. Only quantization is not assumed to be ideal and will be added as additional noise. The data stream is assumed to be deterministic but unknown, i.e. \( x(t) = x \).

In part I, the in-phase and quadrature (DC-)components are modulated to bandpass domain by multiplying the data stream with a carrier signal. After that, the data is transmitted over an additive white Gaussian noise (AWGN) channel. The noise, \( w(t) \), consists only of real components. \( s(t) \) is the input signal of the quadrature receiver. In II, \( s(t) \) is separated into in-phase and quadrature components. After modulation and filtering, II & III, the signal is again represented in baseband. As \( B << f_0 \) sampling at a low frequency is possible. After sampling, additional quantization noise is added. Finally, the measurement data is given as \( y[n] = y_r[n] + j y_i[n] \), which is the distorted complex envelope of \( s(t) \).

Following lines show the mathematical representation of the signals at each part of the transceiver chain. The transmitted signal, after AWGN channel, is

\[
s(t) = x_r \cos(2\pi f_0 t) - x_i \sin(2\pi f_0 t) + w(t),
\]

where \( w(t) \sim \mathcal{N}(0, \sigma_w^2) \). In part II, \( s(t) \) is multiplied with cosine and sine to get

\[
s_r(t)[1 - \cos(2 \cdot 2\pi f_0 t)] - s_i(t) \sin(2 \cdot 2\pi f_0 t).
\]

The same principle is true for the quadrature component. The next part is the lowpass filter. The stop frequency of the filter is set to the bandwidth \( f_0 = \frac{B}{2} \). Thus the lowpass
data (with explicit noise) is

\[ s_r(t) = x_r + w'(t) \]  \hspace{1cm} (5.2)
\[ s_i(t) = x_i + w''(t). \]  \hspace{1cm} (5.3)

\( w(t) \) is WGN and thus not effected by modulation, see Section 3.5. Because of that \( w'(t) \) and \( w''(t) \) possess same statistical properties. In part IV, signals are sampled with at least a sample frequency of \( f_s > B \). Here, quantization noise \( q[n] \) (CWGN) is added to both paths, \( q_r[n] \sim \mathcal{N}(0, \frac{\sigma^2_q}{2}) \) and \( q_i[n] \sim \mathcal{N}(0, \frac{\sigma^2_q}{2}) \). After multiplying the imaginary component with \( j \) both parts are added up. The distorted complex envelope signal is then given by

\[ y[n] = y_r[n] + j y_i[n] \]
\[ = x + w[n] + q[n] \]  \hspace{1cm} (5.4)

with \( w[n] = w'[n] + j w''[n] \) and \( x = x_r + j x_i \). Consequently (5.4) can be rewritten in vector notation

\[ y = x \mathbf{1} + w + q. \]  \hspace{1cm} (5.5)

where \( \mathbf{1} = [1 \ 1 \ \cdots \ 1]^T_N \). The noise vector is distributed as \( w \sim \mathcal{CN}(0, 2\sigma_w^2 \mathbf{I}) \). The quantization noise is modeled as complex noise, as in-phase and quadrature components do have the same variance and are uncorrelated. Furthermore, \( q_r[n] \) and \( q_i[n] \) are white noise, which means that each sample is independent of all others, resulting in proper \( q \), distributed as \( q \sim \mathcal{CN}(0, \sigma_q^2 \mathbf{I}) \).

The data model (5.5), is similar to the complex linear data model, but with two different noise sources. The MVU is

\[ \hat{x} = (\mathbf{1}^H \mathbf{C}_{yy}^{-1} \mathbf{1})^{-1} \mathbf{1}^H \mathbf{C}_{yy}^{-1} y. \]  \hspace{1cm} (5.6)

To calculate \( \hat{x} \), the autocovariance matrix of \( y \) is required. With both, \( w \) and \( q \), zero mean, the expected value of \( y \) is

\[ E(y) = x \mathbf{1} \]  \hspace{1cm} (5.7)

The Hermitian covariance matrix of \( y \) is then given by

\[ \mathbf{C}_{yy} = E[(y - E(y))(y - E(y)^H)] = E[(w + q)(w^H + q^H)]. \]
Since $E(w) = 0$ and $E(q) = 0$ and $w$ independent of $q$, $C_{yy}$ becomes

$$C_{yy} = C_{ww} + C_{qq}.$$  

As $C_{ww} = 2\sigma_w^2 I$ and $C_{qq} = \sigma_q^2 I$, the Hermitian covariance matrix of $y$ is finally given by

$$C_{yy} = (2\sigma_w^2 + \sigma_q^2) I,$$

(5.8)

with $I$ the $N$-dimensional unity matrix. Inserting (5.8) into (5.6), $\hat{x}$ becomes the sample mean

$$\hat{x} = \frac{1}{N} \sum_{n=0}^{N-1} y[n].$$

(5.9)

For the estimation algorithm, $s(t)$ is sampled to get $s[n]$. As $s(t)$ is the bandpass signal with a carrier frequency of $f_0$, a sampling frequency of $f_s > 2f_0$ is needed. In contrast, the baseband signal only requires a sampling frequency of $f_s > B$, where $B << f_0$. Refer to Figure 1.1a and Figure 1.1b, which show the spectrum of a bandpass and a baseband signal. Considering the bandpass case, where $x$ shall be estimated out of $s(t)$, a non complex estimator is sufficient as all signals are real-valued. According to (5.1), the data model is

$$s = H x_R + w,$$

(5.10)

where $s$ is the vector of received data, $w$ the noise vector, and $x$ the parameter vector, with $x_R = [x_r \ x_i]^T$. $H$ is the so called observation matrix. By using the normalized frequency $\Omega_0 = 2\pi f_0 T_s$, $H$ is defined as

$$H = [h_1 \ h_2]^T$$

$$= \begin{bmatrix}
1 \\
\cos[\Omega_0] & -\sin[\Omega_0] \\
\vdots & \vdots \\
\cos[\Omega_0(M-1)] & -\sin[\Omega_0(M-1)]
\end{bmatrix}.$$  

(5.11)

Since (5.10) possesses the form of the linear model, the estimator is [19]

$$\hat{x}_R = (H^T C_{ss}^{-1} H)^{-1} H^T C_{ss}^{-1} s$$

(5.12)

Modulations are assumed to be ideal. The estimator can now be reduced to

$$\hat{x}_R = \begin{bmatrix}
\hat{x}_r \\
\hat{x}_i
\end{bmatrix}$$

$$= \begin{bmatrix}
\frac{\sum_{m=0}^{M-1} s[m] \cos(\Omega_0 m)}{\sum_{m=0}^{M-1} \cos^2(\Omega_0 m)} \\
\frac{\sum_{m=0}^{M-1} s[m] \sin(\Omega_0 m)}{\sum_{m=0}^{M-1} \sin^2(\Omega_0 m)}
\end{bmatrix}$$

$$= \begin{bmatrix}
h_1^T s \\
h_1^T h_1 \\
h_1^T s \\
h_2^T h_2
\end{bmatrix}$$

(5.13)
Both estimators, (5.9) and (5.13), have minimum variance and are efficient. The band-pass estimator works at a high sampling frequency. Further (5.13) requires \(2M\) multiplications. The complex envelope estimator (5.9) only needs one complex multiplication and a lower sampling frequency. To compare the calculation effort of the two estimators, \(M = N\) samples are used for both approaches. In reality, much more samples are available in the bandpass case, i.e. \(M >> N\). Thus the effort for (5.13) is even higher. Knowing the signals before estimation allows precalculation of the denominator in (5.13), which reduces the required computation effort.

A MATLAB program was used to simulate the performance of the estimators. The script simulates the complete transceiver chain as illustrated by Figure 5.1. WGN is created with the `randn` command. 100 experimental runs, where each generates a unique noise with same statistical properties, are used to show the difference between the estimators. In simulation, \(N = M = 1000\) samples are used per run. Figure 5.2 and Figure 5.3 show the results of the in-phase and quadrature estimators, as well as their squared errors. The simulation was executed with quantization noise \(q \sim \mathcal{CN}(0, 0.02 I)\) and \(w(t) \sim \mathcal{N}(0, 0.02)\). Here, it is assumed that sampling in the passband is ideal. The resulting MSE, in this case the sum of in-phase and quadrature estimator variances, of the bandpass/baseband estimator (quadrature and in-phase) are \(0.6589 \times 10^{-4}/0.3591 \times 10^{-3}\). To have a fair result for comparison, figures 5.4 and 5.5 assume that the additive quantization noise in the baseband is zero, i.e. \(q[n] = 0\). The simulated MSEs for bandpass/baseband are \(0.8410 \times 10^{-4}/1.3215 \times 10^{-4}\). It can be seen that the MSE of the baseband estimator significantly decreases if the quantization noise is removed from equivalent baseband signal. Anyhow, it can be seen that the statistical performance of the bandpass estimator is still better. This is a consequence of the non perfect FIR filter in the MATLAB simulation, but the required calculation effort of the equivalent baseband estimator is much lower. The implementation of (5.9) is straight forward and does not consume much space on hardware.

### 5.2 Advantages of WLMMSE for CP-OFDM and UW-OFDM

This example shows the benefits of the WLMMSE algorithm in contrast to the LMMSE on a real world example. Therefore, the basic concept of the widely used modulation technique Orthogonal Frequency-Division Multiplexing (OFDM) serves as setup to simulate the behavior of the estimation algorithms. OFDM is nowadays one of the most efficient methods implemented in wireless communication standards (WLAN, WPAN, LTE). This multicarrier transmission scheme reduces the effects of multipath fading and lowers the complexity of equalizers, which are the focus of this example. OFDM is less sensible to narrowband interference or channel fading than single carrier transmission systems [31]. However, a drawback of OFDM is a higher sensitivity to frequency offsets, phase noise, and a high peak-to-average power ratio. FFT and its inverse, the iFFT, are important blocks of the process. In OFDM the QUAM symbols are mapped to subcarriers at a given frequencies. All subcarriers are orthogonal to each other, thus there is no symbol interference. At the center frequency of each symbol all other subcarrier spectra are zero. The basic principle of the OFDM transmission scheme is shown in Figure 5.7, which shows the scheme for cyclic prefix (CP-)OFDM. In CP-OFDM transmitters, a fixed number of data samples is copied from the end to the beginning of an OFDM symbol. These samples
are also called guard interval. It reduces the channel influence on the data stream. Thus the impact of channel impulse response is limited to the guard interval, which is removed at the receiver. For further details on CP-OFDM refer to [31]. Another technique, introduced in [11], is the so called unique word (UW-)OFDM. There, the CP is replaced with a deterministic sequence, called unique words. Simulation results show that UW-OFDM, linked with a proper coding technique, outperforms the usual CP-OFDM [11, 15].

The focus on this example is on equalizers for OFDM systems based on the Bayesian approach, Section 4.1. They have the task to inverse the channel influence and reduce the noise. Figure 5.6 shows a basic block model. Equalizer may also be denoted as channel inversion filter. In recent publications, [11, 15, 21, 10], the mainly used algorithm for the channel inversion filter is the zero forcing (ZF) equalizer in CP-OFDM and LMMSE in UW-OFDM. As shown in Section 4.1.3, the LMMSE is only the optimal linear estimator if the proper conditions (3.46) and (3.47) are fulfilled. Now, as one can see in Figure 5.7, a specific modulation technique is used before the FFT. The most common ones have been introduced in Section 3.3.2, where also the impact on the statistics is described. If the modulation is improper, the LMMSE is only suboptimal and shall be replaced by the
Figure 5.3: Results of quadrature bandpass/baseband estimator. Mean: 2.9997/2.9990; Variance: 0.3418 $10^{-4}$/0.1744 $10^{-3}$

WLMMSE. Furthermore, UW-OFDM introduces a redundancy in the data stream [15]. The following simulation examples focuses on ASK modulated signals and thus the data is improper. The example is split in a CP and a UW part. From here, no further attention is paid on the details on the OFDM transmission scheme, except if properties influence the estimation result.

The simulation is performed over increasing signal-to-noise ratio (SNR). With zero-mean the noise variance is given by

$$\sigma_n^2 = E_b \cdot 10^{-\frac{\text{SNR}_{\text{dB}}}{10}}.$$  \hspace{1cm} (5.14)

At each SNR value, 20 channels are simulated. 5 data bursts with 64k symbols are transmitted per channel. The BER for each channel is the sum of all bit errors divided by the lengths of the overall data stream. The plotted BER and MSE curves are the average over all channels. The MSE is calculated as in (4.1), thus it is the mean over the sum of all individual squared errors. This principle is repeated for all ASK constellation sizes. Noise is assumed to be CWGN, i.e. $n \sim \mathcal{CN}(0, \sigma_n^2 I, \tilde{C}_{mn} = 0)$. Note, the estimator
works in the frequency domain. Consequently the noise properties must be transformed. It is shown in [15] that the autocovariance of $\tilde{n}$ calculates as $C_{\tilde{n}\tilde{n}} = N C_{nn}$, where $N$ is the length of the noise vector. The statistical distribution of the data is based on the chosen modulation scheme. It is assumed that each point in the ASK scheme has the same probability. With zero mean, the variance can be calculated by

$$\sigma_d^2 = \frac{1}{m\text{PAM}} \sum_{a=1}^{m\text{PAM}} |\tilde{d}_a|^2,$$

(5.15)

where $m\text{PAM}$ is the number of symbols in the ASK modulation and $\tilde{d}_a$ the symbol value. Similar, the pseudo-variance is given by

$$\tilde{\sigma}_d^2 = \frac{1}{m\text{PAM}} \sum_{a=1}^{m\text{PAM}} (\tilde{d}_a)^2.$$

(5.16)

As every $\tilde{d}_a$ is real valued, (5.16) and (5.15) are equivalent. The complex correlation coefficient becomes $\rho = \tilde{\sigma}_d^2/\sigma_d^2 = 1$. With each sample being independent of all others the
covariance matrices of the data are $\mathbf{C}_{\tilde{d}\tilde{d}} = \sigma^2_d \mathbf{I} = \mathbf{C}_{\tilde{d}\tilde{d}}$. Two different channel models are used, AWGN and predefined IEEE indoor radio channels. For more details on how these channels are generated see [9]. MATLAB scripts can be found in appendix B.3 and B.4.

5.2.1 CP-OFDM

The binary data stream $d$ is first generated as a white random process and then mapped to ASK symbols. These symbols are represented in the frequency domain $\tilde{d}$. Except for covariance matrices, the ($\tilde{\cdot}$) denotes that a signal/vector is given in frequency domain. The iFFT transforms $\tilde{d}$ into time domain $x = \text{iDTFT}(\tilde{d})$. Next, the CP is added. After that, the stream is filtered with the impulse response of a specific channel and noise is added, i.e. $y = Hx + n$. At the receiver, the guard intervals are removed and the measurements are transformed into frequency domain. Then, the original transmitted stream is estimated, $\hat{\tilde{d}}$, with either the LMMSE, the WLMMSE, or Channel Inversion (CI) filter. Note, the CI estimator only inverts the influence of the channel, i.e. $\hat{\tilde{d}} = \mathbf{H}^{-1}\tilde{y}$. The estimated ASK...
symbols are demodulated, \( \hat{d} \rightarrow \tilde{d} \). Afterwards, the demodulated signals are compared to the original data symbols. Finally the bit error rate (BER) and the MSE are calculated. The data model of an CP-OFDM system is given by [31]

\[
\tilde{y} = \tilde{H} \tilde{d} + \tilde{n},
\]

(5.17)

where \( \tilde{y} \) are the received measurements. The CP is already removed. \( \tilde{H} \) is the diagonal frequency domain channel matrix, \( \tilde{d} \) the source data, and \( \tilde{n} \) the frequency domain representation of the noise. \( \tilde{y} \) is fed into the estimation algorithm. (5.17) is the linear model, described in sections 4.1.1 and 4.1.2. With the assumptions above, the covariance matrices are given by

\[
C_{\tilde{y}\tilde{y}} = \sigma^2_d \tilde{H} \tilde{H}^H + \sigma^2_n \mathbf{I}_N
\]

(5.18)

\[
\tilde{C}_{\tilde{y}\tilde{y}} = \sigma^2_d \tilde{H} \tilde{H}^T
\]

(5.19)

\[
C_{\tilde{d}\tilde{y}} = \sigma^2_d \tilde{H}^H
\]

(5.20)

\[
\tilde{C}_{\tilde{d}\tilde{y}} = \sigma^2_d \tilde{H}^T
\]

(5.21)

The pseudo-covariance matrices do not vanish, indicating that the measurements are improper. These are used in the LMMSE and WLMMSE equations, (4.40) and (4.88), to yield an estimate \( \hat{d} \) of the source data. Note, the LMMSE only uses the Hermitian covariance matrices, ignoring the pseudo ones completely. Figures 5.8 and 5.9 show the results from the simulations. Regarding the MSE, the WLMMSE outperforms the LMMSE and the CI, shown in figures 5.8b and 5.9b. These simulation results prove the theoretical results derived in Section 4.1.3. Interestingly, only little benefit is achieved when looking at BER curves. Only for 4-ASK and 8-ASK a small gain can be recognized, however, this comes with the price of about 3.5 higher computation effort of the WLMMSE, see Section 4.1.3. The WLMMSE shows a significant performance gain in the MSE plots. Anyhow, these results are not reflected in the BER curves. This effect is caused by the demodulator. It only considers the real components of \( \tilde{d} \). The LMMSE yields complex results in general, illustrated by the scatter plot in Figure 5.10. In contrast to the demodulator, the MSE considers the imaginary components. Therefore, the squared error magnitude of the LMMSE
Figure 5.7: Block diagram of OFDM transmission with ASK subcarrier modulation.
is higher. When examining the MSE of the real parts only, see Figure 5.11, it can be seen that also the MSE difference between the estimators decreases. Still, the WLMMSE is the best. As already mentioned above, the CI only inverts the channel influence, but does not care about the noise at all. It follows that the MSE results are even worse than the one from the LMMSE. However, looking on the BER results the CI is more reliable than the WLMMSE.

5.2.2 UW-OFDM

In UW-OFDM, the guard interval is a known, fixed sequence of symbols. Further it is added to the data stream before the iFFT [15]. These can be chosen to ease channel estimation in a training phase. However, these words do not influence the estimation process. Like the CP, the UW is simply cut off at the receiver before the equalizer. In UW-OFDM redundant data samples \( \tilde{\mathbf{r}} \) are implemented. Further, a code generator matrix \( \mathbf{G} \) generates codewords out of \( \mathbf{d} \), i.e. \( \tilde{\mathbf{c}} = \mathbf{G} \mathbf{d} \). The codeword \( \tilde{\mathbf{c}} \) is composed of data plus redundant subcarriers. Refer to [11, 15, 23, 7, 12, 24, 13, 8, 14, 6, 25] for more details on UW-OFDM. The resulting data model at the receiver is given by [15]

\[
\tilde{\mathbf{y}} = \tilde{\mathbf{H}} \mathbf{G} \tilde{\mathbf{d}} + \mathbf{n}
\]

(5.22)

\( \tilde{\mathbf{H}} \) is again the frequency domain channel matrix [23]. \( \tilde{\mathbf{H}} \) and \( \mathbf{G} \) are known matrices and can be combined to one, i.e. \( \tilde{\mathbf{M}} = \tilde{\mathbf{H}} \mathbf{G} \). (5.22) becomes (assuming channel knowledge at the receiver)

\[
\tilde{\mathbf{y}} = \tilde{\mathbf{M}} \tilde{\mathbf{d}} + \mathbf{n},
\]

(5.23)

The resultant covariance matrices are given by

\[
\begin{align*}
\mathbf{C}_{\tilde{\mathbf{y}} \tilde{\mathbf{y}}} & = \sigma_d^2 \tilde{\mathbf{M}} \tilde{\mathbf{M}}^H + \sigma_n^2 \mathbf{I}_N \\
\tilde{\mathbf{C}}_{\tilde{\mathbf{y}} \tilde{\mathbf{y}}} & = \sigma_d^2 \tilde{\mathbf{M}} \tilde{\mathbf{M}}^T \\
\mathbf{C}_{\tilde{\mathbf{y}} \tilde{\mathbf{d}}} & = \sigma_d^2 \tilde{\mathbf{M}}^H \\
\tilde{\mathbf{C}}_{\tilde{\mathbf{y}} \tilde{\mathbf{d}}} & = \sigma_d^2 \tilde{\mathbf{M}}^T
\end{align*}
\]

(5.24) (5.25) (5.26) (5.27)

Inserting these into the LMMSE and WLMMSE formulas yield the results shown in figures 5.12 and 5.13. As illustrated, the WLMMSE outperforms the LMMSE not only regarding the MSE, but also at the BER. At maximum, there is about 5dB difference. Further, the MSE plots illustrate that WLMMSE results at high SNRs (\( \sim 25 \)dB) outperform the LMMSE at even lower constellation size. For instance, in Figure 5.12b, the MSE of the LMMSE at 4-ASK is higher than the one of the WLMMSE at 8-ASK. Thus a higher order modulation can be used, while still keeping the same MSE. Unfortunately, regarding the BER, this is not the case. The effect is the same as at CP-OFDM. The biased estimator would need a rearrangement of the decision boundaries of the demodulator. Anyhow, with the introduced codewords \( \tilde{\mathbf{c}} \) by \( \mathbf{G} \), the WLMMSE is the better choice.

5.2.3 Conclusion

With ASK modulation, impropriety is introduced to the system. Therefore, from Section 4.1.3, the WLMMSE is the best performing linear estimator. From figures 5.8a to 5.13a,
Figure 5.8: CP-OFDM: Results for 2-to-16-ASK modulation over AWGN channel.
Figure 5.9: CP-OFDM: Results for 2-to-16-ASK modulation over (20) IEEE indoor channels.
the simulation verified these theoretical results. The MSE of the LMMSE is at each constellation size and at each SNR worse than for the WLMMSE. Unfortunately, in CP-OFDM these advantages are not transferred to the BER behavior. The significant improved MSE of the WLMMSE almost vanishes after demodulation, see Figure 5.11. As already mentioned this effect has its origin in the biased estimator and the ignored imaginary components in the demodulation process. In CP-OFDM, only at 4-ASK and 8-ASK a maximum gain of $\sim 0.6$ dB can be achieved. In contrast, with the introduced correlations in UW-OFDM, the WLMMSE shows an up to 5 dB performance gain in the BER curves. The simulation points out, that pseudo-characteristics considerably influence the performance of the estimator. The WLMMSE in improper systems always yields superior results in contrast to the LMMSE. The following table shows the parameters used for the CP-OFDM and the UW-OFDM MATLAB simulation.

### 5.3 Channel Estimation with LMS Algorithm

In the examples and derivations above, the LMMSE and WLMMSE are also interpreted as so called Wiener-Filters. Figure 5.14a shows the block diagram of such a filter. The optimum weighting coefficients of the LMMSE, or WLMMSE respectively, are used as coefficients. In literature, this is denoted as batch-approach. A separate training phase is
required to gather all necessary statistical information. These are then used for calculating the optimal estimator at a specific time step. If characteristics change, the derived coefficients are no longer optimal. Thus a Wiener-Filter is static, until the training process is repeated. Mathematically expressed this is $w(t) = w(\infty) = w$. This example restricts to FIR Wiener-Filters. See Section 2.4 for a brief review and the realization of a complex FIR filter.

A possible solution to overcome this penalty is an update of the filter coefficients via a feedback and correction algorithm. It alters the solution to an adaptive filter. These are an active research topics since decades [32]. Figure 5.14b illustrates a simple block model of an adaptive filter. The update of $w$ can be done with the Least Mean Square (LMS) algorithm. It modifies the filter coefficients at time step $(k + 1)$ by using the actual ones at $k$. This type is called sample-adaptive filter, because at each $k$ the coefficients are updated. The estimated signal is described by

$$\hat{x}(k) = w(k)^H y(k).$$  (5.28)
Figure 5.12: UW-OFDM: Results for 2-to-16-ASK modulation over AWGN channel.
Figure 5.13: UW-OFDM: Results for 2-to-16-ASK modulation over (20) IEEE indoor channels.
CHAPTER 5. APPLICATIONS FOR COMPLEX-VALUED DATA ESTIMATION

(a) Wiener-Filter.

(b) Adaptive filter. The coefficients are no longer static, but updated at each time step \( k \).

Figure 5.14: Block diagram of a Wiener-Filter and an adaptive filter. \( y \) is the input signal, \( x \) the desired signal, \( \hat{x} \) the estimate, \( e \) the error, and \( w \) are the filter coefficients.

The complex LMS algorithm for the strictly linear model (4.14) is derived in [33]. There, the original Widrow-Hoff LMS solution [32] is altered to the complex case. The fundamental idea of the LMS is also shown in [28, 16]. The algorithm from [33, 28], is given by

\[
 w(k+1) = w(k) + \mu e^*(k) y(k). \tag{5.29}
\]

The new filter coefficients are based on the old ones and corrected by an additional term, which depends on the conjugate of the estimation error and the measurement vector. \( \mu > 0 \) is the step size of the algorithm. High \( \mu \) allow faster convergence, but usually causes higher misadjustment. In contrast, a lower step size requires a long convergence time. The convergence time depends on the eigenvalues of the input covariance matrix \( C_{yy} \) [32]. The error is given by \( e(k) = x(k) - w^H(k) y(k) \).

Next, the LMS is extended to the widely linear case. It is then called widely least mean square (WLMS) algorithm. It is the adaptive extension to the WLMMSE, which also considers pseudo-characteristics. Using the augmented representation, the WLMS is denoted by [28]

\[
 w_{wl}(k+1) = w_{wl}(k) + \mu e^*(k) y(k). \tag{5.30}
\]

The error \( e(k) = x(k) - w_{wl}^H(k) y(k) \) is defined above. From Section 4.1.2, the linear filter \( w_{wl} \) can be divided into two filters, yielding the widely linear form for the estimated parameter

\[
 \hat{x}(k) = h(k)^H y(k) + g(k)^H y^*(k), \tag{5.31}
\]

where \( w_{wl}(k) = [h(k)^H g(k)^H]^H \). The LMS algorithm for both filters simply follows to [16]

\[
 h(k+1) = h(k) + \mu e^*(k) y(k) \tag{5.32}
\]

\[
 g(k+1) = g(k) + \mu e^*(k) y^*(k). \tag{5.33}
\]

Adaptive filters are often used in signal processing applications. For example, it is often required to estimate a channel impulse response, which is then needed for further processing. For instance in OFDM systems, LMMSE and/or the WLMMSE require the

\[
 w(k+1) = w(k) + \mu e^*(k) y(k) \tag{5.29}
\]
channel frequency response. This one is not known in general, or may change over time. The task is to come as close to the unknown system as possible. With the above described LMS an adaptive filter can be used. In the following example, illustrated by Figure 5.15, the system is described as
\[ y = Sx + Mx^* + n, \]
where \( y \in \mathbb{C}^N \) is the output of the system. \( S \) and \( M \) are convolution matrices. The complex white data stream is
\[ x \sim \mathcal{CN}\left(0, C_{xx} = \sigma_x^2 I, \tilde{C}_{xx} = \sigma_x^2 I\right) \]
and CWGN is
\[ n \sim \mathcal{CN}\left(0, C_{nn} = \sigma_n^2 I, \tilde{C}_{nn} = 0\right). \]
In MATLAB, the LMS filter (5.28) is denoted as \( h \) and the WL filter (5.31) as \( a \) and \( b \) respectively. (5.34) is the model of widely linear system. The simulation is first performed with a strictly linear system, \( M = 0 \), and varying impropriety \( \rho = \tilde{\sigma}^2 / \sigma^2 \). Then the system is altered to the widely linear case, where \( M \neq 0 \). The coefficient size of the filters is fixed to \( p = 10 \). \( N = 4000 \) samples are used at each simulation.

![Figure 5.15: Block diagram of system identification example.](image)

Figures 5.16 - 5.19 show the simulation results. Beside the results of the MSEs the convergence time is a crucial component in the design of LMS filters. The convergence time depends on the eigenvalues of the input covariance matrix [3]. As a measure, the condition number \( \kappa(C_{yy}) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \) [28] can be used, where \( \lambda \) are the eigenvalues of \( C_{yy} \). This is only possible for a Hermitian covariance matrix. In case of the WLMS, the augmented input covariance matrix \( \tilde{C}_{yy} \) must be used. Only if the signal is proper, then the condition numbers of \( C_{yy} \) and \( \tilde{C}_{yy} \) are equivalent [28]. Otherwise, \( \kappa(C_{yy}) \) is always higher than \( \kappa(C_{yy}) \). In the strictly linear and proper case the adjustment time of the WLMS and the LMS are quite the same, see Figure 5.16a. In contrast, Figure 5.17a shows that if the
signals are improper the time required for the WLMS to converge is significantly higher. In the strictly linear case the WLMS is not a good choice as the estimation results of the LMS are the very same but it converges much faster. In the WL case, the WLMS shows its dominance by exploiting all SO characteristics. Even if the input signal is proper the LMS does not have a steady state, which can be seen in Figure 5.18. Further as shown in Figure 5.19 the same is true if the input signal is improper. In the figures 5.18b and 5.19b it can be seen that the trajectories of the coefficient energy concur with the system quite fast. The coefficient energy is calculated as the sum of the squared coefficient magnitudes, i.e. \( E_{\text{coeff}} = h^H h \) where \( h \) is the coefficient vector, respectively. Summarized, if the system itself is WL, then the WLMS is the better choice. For linear ones the LMS can be used without losing performance.
CHAPTER 5. APPLICATIONS FOR COMPLEX-VALUED DATA ESTIMATION

(a) Squared Error Magnitude.

(b) Coefficient energy, i.e. $E_{\text{coeff}} = h^H h$, where $h$ is the vector of filter coefficients, respectively.

Figure 5.16: Simulation results for proper input signal $x[n]$ ($\rho = 0$) over strictly linear channel, $M = 0$. 
(a) Squared Error Magnitude.

(b) Coefficient energy, i.e. $E_{\text{coeff}} = h^H h$, where $h$ is the vector of filter coefficients, respectively.

Figure 5.17: Simulation results for improper input signal $x[n]$ ($\rho = j 0.89$) over strictly linear channel, $M = 0$. 

(a) Squared Error Magnitude.

(b) Coefficient energy, i.e. $E_{\text{coeff}} = h^H h$, where $h$ is the vector of filter coefficients, respectively.

Figure 5.18: Simulation results for proper input signal $x[n]$ ($\rho = 0$) over widely linear channel, $M \neq 0$. 
CHAPTER 5. APPLICATIONS FOR COMPLEX-VALUED DATA ESTIMATION

Figure 5.19: Simulation results for improper input signal $x[n]$ ($\rho = j 0.89$) over widely linear channel, $M \neq 0$.

(a) Squared Error Magnitude.

(b) Coefficient energy, i.e. $E_{\text{coeff}} = h^H h$, where $h$ is the vector of filter coefficients, respectively.
Appendix A

Derivations

A.1 Derivation of CR Gradients

A.1.1 Gradient of a Linear Form

This proof shows $\frac{\partial a^H x}{\partial x} = a^*$. Let $a$ and $x$ be $N$-dimensional complex valued vectors. Then the linear form $a^H x$ is

$$a^H x = [a_1^* a_2^* \cdots a_N^*] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$= \sum_{n=1}^{N} a_n^* x_n \quad (A.1)$$

For convenience, the superscript and subscript of the sum symbols will be dropped. Note, range of all sums is as in (A.1). With (2.16) the gradient becomes

$$\frac{\partial a^H x}{\partial x} = \begin{bmatrix} \sum_{n=1}^{N} a_n^* x_n \\ \sum_{n=1}^{N} a_n^* x_n \\ \vdots \\ \sum_{n=1}^{N} a_n^* x_n \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial}{\partial x_1} (a_1^* x_1 + a_2^* x_2 + \cdots + a_N^* x_N) \\ \frac{\partial}{\partial x_2} (a_1^* x_1 + a_2^* x_2 + \cdots + a_N^* x_N) \\ \vdots \\ \frac{\partial}{\partial x_N} (a_1^* x_1 + a_2^* x_2 + \cdots + a_N^* x_N) \end{bmatrix} \quad (A.2)$$

Using (2.15) the gradient is finally given by

$$\frac{\partial a^H x}{\partial x} = \begin{bmatrix} a_1^* \\ a_2^* \\ \vdots \\ a_N^* \end{bmatrix} = a^* \quad (A.3)$$
A.1.2 Gradient of Hermitian Form

This proof shows $\frac{\partial x^H Ax}{\partial x} = (A x)^*$. The solution is derived in different steps. Let $x$ be an $N$-dimensional complex vector and $A$ a Hermitian $N \times N$-dimensional matrix in the form of

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{12}^* & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1N}^* & a_{2N}^* & \cdots & a_{NN} \end{bmatrix}, \quad (A.4)$$

whereas the diagonal elements $a_{nn}$ are real-valued scalars. From (A.4) it follows that $A^H = A$ and $A^T = A^*$. The next step is

$$x^H A x = [x_1^* x_2^* \cdots x_N^*] \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$= \left[ \sum a_{n1} x_n^* \sum a_{n2} x_n^* \cdots \sum a_{nN} x_n^* \right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$= x_1 \sum a_{n1} x_n^* + x_2 \sum a_{n2} x_n^* + \cdots + x_N \sum a_{nN} x_n^*$$

$$= \theta \quad (A.5)$$

Note, all sums (above and below) are $\sum_{n=1}^N$. The CR-Calculus rules (2.16) and (2.15) are used to calculate the gradient

$$\frac{\partial x^H Ax}{\partial x} = \frac{\partial \theta}{\partial x}$$

$$= \begin{bmatrix} \frac{\partial \theta}{\partial x_1} \\ \frac{\partial \theta}{\partial x_2} \\ \vdots \\ \frac{\partial \theta}{\partial x_N} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial}{\partial x_1} \left( x_1 \sum a_{n1} x_n^* + x_2 \sum a_{n2} x_n^* + \cdots + x_N \sum a_{nN} x_n^* \right) \\ \frac{\partial}{\partial x_2} \left( x_1 \sum a_{n1} x_n^* + x_2 \sum a_{n2} x_n^* + \cdots + x_N \sum a_{nN} x_n^* \right) \\ \vdots \\ \frac{\partial}{\partial x_N} \left( x_1 \sum a_{n1} x_n^* + x_2 \sum a_{n2} x_n^* + \cdots + x_N \sum a_{nN} x_n^* \right) \end{bmatrix} \quad (A.6)$$

As from (2.15), $x_n$ and $x_n^*$ are treated independent, thus the gradient is finally given by

$$\frac{\partial x^H Ax}{\partial x} = \begin{bmatrix} \sum a_{n1} x_n^* \\ \sum a_{n2} x_n^* \\ \vdots \\ \sum a_{nN} x_n^* \end{bmatrix} \quad (A.7)$$
To show that $\frac{\partial x^H A x}{\partial x} = (A x)^*$, it is also required to calculate the latter term:

$$(A x)^* = \left( \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \right)^*$$

$$= \left[ \sum a_{1n}^* x_n^* \right] \left[ \sum a_{2n}^* x_n^* \right] \cdots \left[ \sum a_{Nn}^* x_n^* \right]$$

$$\downarrow \quad A^T = A^* \leftrightarrow a_{mn}^* = a_{nm}; \; a_{nn}^* = a_{nn}$$

$$= \left[ \sum a_{n1} x_n^* \right] \left[ \sum a_{n2} x_n^* \right] \cdots \left[ \sum a_{nN} x_n^* \right]$$  \hspace{1cm} (A.8)

### A.2 Derivation of $\mathbf{H}$

The augmented matrix can be calculated by $\mathbf{H} = \frac{1}{2} \mathbf{T}_M \mathbf{M} \mathbf{T}_N^H$, where $\mathbf{T}_M$ and $\mathbf{T}_N$ are the $M$- and $N$-dimensional transformation matrices (2.36). The real-valued transformation matrix $\mathbf{M}$ is split into four block matrices:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}. \quad (A.9)$$

With these definitions, the augmented matrix $\mathbf{H}$ is derived as

$$\mathbf{H} = \frac{1}{2} \mathbf{T}_M \mathbf{M} \mathbf{T}_N^H$$

$$= \frac{1}{2} \begin{bmatrix} \mathbf{I}_M & j\mathbf{I}_M \\ \mathbf{I}_M & -j\mathbf{I}_M \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ -j\mathbf{I}_N & j\mathbf{I}_N \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} \mathbf{I}_M & j\mathbf{I}_M \\ \mathbf{I}_M & -j\mathbf{I}_M \end{bmatrix} \begin{bmatrix} (\mathbf{M}_{11} - j\mathbf{M}_{12})\mathbf{I}_N \\ (\mathbf{M}_{21} - j\mathbf{M}_{22})\mathbf{I}_N \end{bmatrix} \begin{bmatrix} (\mathbf{M}_{11} + j\mathbf{M}_{12})\mathbf{I}_N \\ (\mathbf{M}_{21} + j\mathbf{M}_{22})\mathbf{I}_N \end{bmatrix}$$  \hspace{1cm} (A.10)

With $\mathbf{A}_{M \times N} \mathbf{I}_{N \times N} = \mathbf{A}_{M \times N}$ and $\mathbf{B}_{M \times N} \mathbf{I}_{M \times M} = \mathbf{B}_{M \times N}$ it follows

$$\mathbf{H} = \frac{1}{2} \begin{bmatrix} \mathbf{M}_{11} - j\mathbf{M}_{12} + j(\mathbf{M}_{21} - j\mathbf{M}_{22}) \\ \mathbf{M}_{11} - j\mathbf{M}_{12} - j(\mathbf{M}_{21} - j\mathbf{M}_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} + j\mathbf{M}_{12} + j(\mathbf{M}_{21} + j\mathbf{M}_{22}) \\ \mathbf{M}_{11} + j\mathbf{M}_{12} - j(\mathbf{M}_{21} + j\mathbf{M}_{22}) \end{bmatrix}, \quad (A.11)$$

and finally

$$\mathbf{H} = \begin{bmatrix} \frac{1}{2} [\mathbf{M}_{11} + \mathbf{M}_{22} + j(\mathbf{M}_{21} - \mathbf{M}_{12})] \\ \frac{1}{2} [\mathbf{M}_{11} - \mathbf{M}_{22} - j(\mathbf{M}_{21} + \mathbf{M}_{12})] \end{bmatrix} \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \quad \begin{bmatrix} \frac{1}{2} [\mathbf{M}_{11} - \mathbf{M}_{22} + j(\mathbf{M}_{21} + \mathbf{M}_{12})] \\ \frac{1}{2} [\mathbf{M}_{11} + \mathbf{M}_{22} - j(\mathbf{M}_{21} - \mathbf{M}_{12})] \end{bmatrix} \begin{bmatrix} \mathbf{H}_1 \mathbf{H}_2 \end{bmatrix}. \quad (A.12)$$
A.3 Inner Products

Proof of the inner products $\mathbf{x}^T \mathbf{y}_R = \Re \left( \mathbf{x}^H \mathbf{y} \right) = \frac{1}{2} \mathbf{x}^H \mathbf{y}$. First the dot product of the equivalent real vectors is given as

$$
\mathbf{x}^T \mathbf{y}_R = \begin{bmatrix} \mathbf{x}_r^T & \mathbf{x}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{y}_r \\ \mathbf{y}_i \end{bmatrix} = \sum_{k=1}^N x_{rk} y_{rk} + \sum_{k=1}^N x_{ik} y_{ik} \quad (A.13)
$$

For the augmented vectors one gets

$$
\frac{1}{2} \mathbf{x}^H \mathbf{y} = \frac{1}{2} \left[ \mathbf{x}^H \mathbf{x}^T \right] \begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} = \frac{1}{2} \left[ x_1^* x_2^* \cdots x_n^* \right] \left[ x_1 x_2 \cdots x_n \right] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_1^* \\ y_2^* \\ \vdots \\ y_n^* \end{bmatrix} = \frac{1}{2} \left[ (x_1^* y_1 + x_2^* y_2 + \cdots + x_n^* y_n) + (x_1 y_1^* + x_2 y_2^* + \cdots + x_n y_n^*) \right] \\
= \frac{1}{2} \left[ \frac{(x_1^* y_1 + x_1 y_1^*) + \cdots + (x_n^* y_n + x_n y_n^*)}{2(x_{r1} y_{r1} + x_{i1} y_{i1})} \right] \\
= \frac{1}{2} \left[ 2 \left( x_{r1} y_{r1} + x_{i1} y_{i1} + \cdots + x_{rn} y_{rn} + x_{in} y_{in} \right) \right] \\
= \left[ \mathbf{x}_r^T \mathbf{x}_i^T \right] \begin{bmatrix} \mathbf{y}_r \\ \mathbf{y}_i \end{bmatrix} = \sum_{k=1}^N x_{rk} y_{rk} + \sum_{k=1}^N x_{ik} y_{ik} \quad (A.14)
$$
Finally for the complex-valued vectors one gets

\[
\Re(x^Hy) = \Re \left( \begin{bmatrix} x_1^* & x_2^* & \cdots & x_n^* \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \right) \\
= \Re \left( \sum_{k=1}^{N} x_k^* y_k \right) \\
= \Re \left[ \sum_{k=1}^{N} (x_{rk} y_{rk} + x_{ik} y_{ik}) + j \sum_{k=1}^{N} (x_{ik} y_{rk} - x_{rk} y_{ik}) \right] \\
= \sum_{k=1}^{N} x_{rk} y_{rk} + \sum_{k=1}^{N} x_{ik} y_{ik}
\]

(A.15)

### A.4 ACF and CCF

This appendix sections shows the even and Hermitian property of the ACF of a real and a complex SOS random process, respectively. First follows the proof that the ACF is an even function for real-valued random processes. Here, SOS is assumed. Assume a real-valued random process, where its ACF is given as

\[
r_{xx}(\tau) = \int_{-\infty}^{\infty} x(u) x(u - \tau) du.
\]

(\(A.16\))

\(\tau\) is the time lag between two time instances \((u_1 - u_2)\). Now the ACF of \(-\tau\) shows to be

\[
r_{xx}(-\tau) = \int_{-\infty}^{\infty} x(w) x(w + \tau) dw,
\]

(\(A.17\))

where \(\tau\) is the very same time lag as in (\(A.16\)). If substituting \(u = w + \tau\), it follows that

\[
\begin{align*}
  u &= w + \tau, \\
  w &= u - \tau, \\
  dw &= du
\end{align*}
\]

Consequently

\[
r_{xx}(-\tau) = \int_{-\infty}^{\infty} x(u - \tau) x(u) du = r_{xx}(\tau).
\]

(A.18)

As a result of the substitution, the ACF of \(-\tau\) becomes the very same as for \(+\tau\).

For the second proof, assume that \(x(t)\) is a complex-valued SOS random process. Its ACF for \(\tau\) and \(-\tau\) are then given as below

\[
r_{xx}(\tau) = \int_{-\infty}^{\infty} x(u) x^*(u - \tau) du
\]

(A.19)

\[
r_{xx}(-\tau) = \int_{-\infty}^{\infty} x(w) x^*(w + \tau) dw
\]

(A.20)
Now, use the same substitution as above, i.e. \( u = w + \tau \):

\[
r_{xx}(-\tau) = \int_{-\infty}^{\infty} x(u - \tau) x^*(u) du
\]  (A.21)

The reader may note, in contrast to (A.19), the conjugate in (A.21) is performed on \( x(u) \) instead of \( x(u - \tau) \). Hence \( r_{xx}(\tau) = r_{xx}^*(-\tau) \) if \( x(t) \) is at least a WSS random process.

### A.5 Derivation of the WLMMSE for Proper Signals via Orthogonality Principle

This section shows that if proper conditions are assumed, the WLMMSE reduces to the LMMSE. All vectors are assumed to have zero-mean. As shown in the orthogonality derivation of the WLMMSE, the error vector is orthogonal to \( y \) and \( y^* \), i.e.

\[
E \left[ (x - W_1 y - W_2 y^*) y^H \right] = 0
\]  (A.22)

\[
E \left[ (x - W_1 y - W_2 y^*) y^T \right] = 0
\]  (A.23)

By solving these equations with proper and cross-proper conditions

\[
\tilde{C}_{yy} = 0
\]  (A.24)

\[
\tilde{C}_{xy} = 0
\]  (A.25)

\[
\tilde{C}_{yx} = 0
\]  (A.26)

the WLMMSE reduces to the LMMSE, since \( W_1 \) is given by

\[
0 = E \left[ (x - W_1 y - W_2 y^*) y^H \right]
= E(x y^H) - W_1 E(y y^H) - W_2 E(y y^T)^*
= C_{xy} - W_1 C_{yy} - W_2 \tilde{C}_{yy}
\]

\[
\downarrow \quad \tilde{C}_{yy} = 0;
\]

\[
= C_{xy} - W_1 C_{yy} - 0
\]

\[
W_1 = C_{xy} C_{yy}^{-1}
\]  (A.27)

and \( W_2 \) becomes

\[
0 = E \left[ (x - W_1 y - W_2 y^*) y^T \right]
= E(x y^T) - W_1 E(y y^T) - W_2 E(y y^H)^*
= \tilde{C}_{xy} - W_1 \tilde{C}_{yy} - W_2 C_{yy}^*
\]

\[
\downarrow \quad \tilde{C}_{yy} = 0;
\]

\[
= 0 - W_1 0 - W_2 C_{yy}^*
\]

\[
= W_2 C_{yy}^*
\]  (A.28)

With assumed nonzero covariance matrix, (A.28) can only be fulfilled if \( W_2 = 0 \). Inserting this in the WLMMSE model shows that it becomes the LMMSE (4.33) if \( y \) is proper and
also cross-proper with $x$:

$$
\dot{x}_{wl} = W_1 y + 0 y^* 
= C_{xy} C_{yy}^{-1} y 
$$

(A.29) (A.30)
Appendix B

MATLAB scripts

B.1 Complex-Valued Gaussian Random Vector

The following function is used to generate a complex-valued Gaussian random vector. The mean and the variances of the real and imaginary parts are needed as inputs to the function. Further the correlation between real and imaginary part can be set to a value between $-1 < \rho_R < 1$. The output is the random vector and all related covariance matrices.

```matlab
function [y, Cyy_R, Cyy, pCyy, aCyy] = 
    createComplexGaussVector(m_r, var_r, m_i, var_i, rho, N)

% real covariance matrix
Cyy = [var_r, rho*sqrt(var_r*var_i);
      rho*sqrt(var_r*var_i), var_i];
R = chol(Cyy); % Cholesky factorization
y = randn(N,2)*R; % create correlated vectors
y = y(:,1) + j*y(:,2); % create complex vector
Cyy_R = [Cyy(1,1)*eye(N) Cyy(1,2)*eye(N);
          Cyy(2,1)*eye(N) Cyy(2,2)*eye(N)];

% complex covariance
Cyy = (var_r+var_i)*eye(N);

% pseudo covariance
pCyy = var_r-var_i+j*2*rho*sqrt(var_r*var_i)*eye(N);

% augmented covariance
aCyy = [Cyy pCyy; conj(pCyy) conj(Cyy)];
end
```
B.2 WLMMSE vs. LMMSE

This section shows the MATLAB source code for the Example in Section 4.1.3.

%% settings
M = 100; % number of transmission bursts
N = 100; % number of data samples
p = 100; % number of parameters

%% statistical properties
% data stream
m_dr = 0; % mean noise real
var_dr = 1; % variance real noise
m_di = 0; % mean imaginary noise
var_di = 1; % variance imaginary noise
delta_rho = 0.01; % step size for correlation coefficient
rho_d = 0:delta_rho:1-delta_rho; % correlation coefficient
% rho_d = 0.2; % fixed

% noise
m_nr = 0; % mean noise real
var_nr = 1e-2; % variance real noise
m_ni = 0; % mean imaginary noise
var_ni = 1e-2; % variance imaginary noise
rho_n = 0;

for r=1:length(rho_d)

% init
d = 0;
n = 0;
MSEm = 0;
WLMSEm = 0;
e=zeros(p,M);
eWL=zeros(p,M);
rho_d(r)
tstart = 0;

for m=1:M

% data generator
[d, Cdd_R, Cdd, pCdd, aCdd] =
    createComplexGaussVector(m_dr, var_dr, m_di, var_di, rho_d(r), p);

APPENDIX B. MATLAB SCRIPTS

44 % noise generator
45 [n, Cnn_R, Cnn, pCnn, aCnn] = 
        createComplexGaussVector(m_nr, var_nr, m_ni, var_ni, rho_n, N);

48 % channel
49 % observation matrix
50 H = eye(N,p); % channel matrix
51 H2 = eye(N,p); % conjugate channel matrix
52 aH = [H H2; conj(H2) conj(H)]; % augmented channel matrix
54 % transmission
55 y = H*d+H2*conj(d)+n; % widely linear channel
57 % covariance matrices
58 Cdy = Cdd*H';
59 Cyy = H * Cdd * H' + Cnn;
61 % augmented covariance matrices
62 aCdy = aCdd * aH';
63 aCyy = aH * aCdd * aH' + aCnn;
65 % Estimation
67 % LMMSE
69 t_start = tic;
70 x = Cdy * Cyy^(-1) * y;
71 % estimation error
72 e(:,m)=d−x;
73 t_LMMSE(r,m) = toc(t_start);
74
77 t_start = tic;
78 % ALMSE
79 ay = [y; conj(y)]; % augmented measurement vector
80 ax = aCdy * inv(aCyy) * ay; % ALMSE
81 xWL = ax(1:p);
82 t_WLMMSE(r,m) = toc(t_start);
84 % estimation error
85 aeWL(:,m)=d−conj(d)−ax;
86 eWL(:,m) = aeWL(1:p,m);
88 end;
%% error covariance matrices
%% LMMSE
Cee = 1/length(e(1,:))*e*e';
MSEr = diag(Cee);
MSE(r) = sum(MSEr);
MSEdb(r) = 10*log10(MSE(r));

%% WLMISE
aCeeWL = 1/length(aeWL(1,:))*aeWL*aeWL';
CeeWL = aCeeWL(1:p,1:p);
MSEWLr = diag(CeeWL);
MSEWL(r) = sum(MSEWLr);
MSEWLdb(r) = 10*log10(MSEWL(r));

end

B.3 MATLAB script for CP-OFDM

This appendix section shows the MATLAB script for the CP-OFDM example in Section 5.2.

%% initial parameters
nSymbols = 1e3; % number of OFDM symbols to transmit at once
N = 64; % DFT length
Nd = N; % length of data symbols
Ng = 16; % length of CP

% load IEEE channels
H=load('chIEEE_100ns.mat');

% SNR
SNR_db_min = 0;
SNR_db_max = 35;
SNR_db = SNR_db_min:SNR_db_max;

% constellation sizes for ASK modulation
const_min = 1; % minimum constellation size 2^x
const_max = 4; % maximum constellation size 2^x
APPENDIX B. MATLAB SCRIPTS

23 \% channels
24 sim_channels = 20; \% simulated channel loops
25
26 \% data transmission bursts per channel
27 data_bursts = 5;
28
29 \% preallocate memory
30 BER_cl = zeros(length(SNR_db), sim_channels);
31 mBER_cl = zeros(length(SNR_db), 4);
32 BER_wl = zeros(length(SNR_db), sim_channels);
33 mBER_wl = zeros(length(SNR_db), 4);
34 BER_ci = zeros(length(SNR_db), sim_channels);
35 mBER_ci = zeros(length(SNR_db), 4);
36 BER_cl = zeros(length(SNR_db), sim_channels);
37 mBER_ci = zeros(length(SNR_db), 4);
38 SER_cl = zeros(length(SNR_db), sim_channels);
39 mSER_wl = zeros(length(SNR_db), sim_channels);
40 mSER_ci = zeros(length(SNR_db), sim_channels);
41 mMSE_wl = zeros(length(SNR_db), 4);
42 mMSE_cl = zeros(length(SNR_db), 4);
43 mMSE_ci = zeros(length(SNR_db), 4);
44 sim_time = 0;
45
46 \% algorithm
47 for m=const_min:const_max \% constellation size loop
48    disp([num2str(2^m), '−ASK']);
49
50    mPAM = 2^m;
51    bps = log2(mPAM); \% bits / symbol
52    bpb = Nd * nSymbols * bps; \% bits / burst
53
54 \% create modulator object
55    modObj = modem.pammod('M', mPAM, 'SymbolOrder',
56                   'binary', 'InputType', 'Bit');
57    demodObj = modem.pamdemod(modObj, 'OutputType', 'Bit');
58
59 \% covariance matrices for data stream
60    var_d = (pammod(0:mPAM-1,mPAM)+pammod(0:mPAM-1,mPAM)) / mPAM;
61    pvar_d = (pammod(0:mPAM-1,mPAM)+pammod(0:mPAM-1,mPAM)) / mPAM;
62    Cdd = var_d * eye(Nd); \% covariance matrix of d_tilde
63    pCdd = pvar_d * eye(Nd); \% pseudo covariance matrix of d_tilde
64
65 for r=(SNR_db_min:SNR_db_max)−SNR_db_min+1 \% SNR loop
66    disp([num2str(SNR_db(r))]);
for ch = 1:sim_channels  % channel loop

% channel selection
% AWGN channel
h = eye(Ng,1);
plot_title = 'CP-OFDM ASK - AWGN channel';

% IEEE channels
% h = H.channels(1:Ng,ch);
% h = h / (sqrt(h'*h)); % normalize channel (no gain)
plot_title = 'CP-OFDM ASK - chIEEE\_100ns';

% random channel
% h = rand(Ng,1) + 1j * rand(Ng,1);  % channel impulse response
% h = h / (sqrt(h'*h)); % normalize channel (no gain)
plot_title = 'CP-OFDM ASK - random channel';

H_tilde = diag(fft(h,N));  % channel frequency response

% preallocate memory and reset results
bf_wl = 0;
bf_cl = 0;
bf_ci = 0;

for b = 1:data_bursts  % data transmission bursts

% preallocate memory and reset results
d = zeros(Nd,nSymbols);
d_tilde = zeros(Nd,nSymbols);
d_bits = zeros(Nd*nSymbols,1);
y = zeros(Nd*nSymbols,1);
xCL = zeros(Nd,nSymbols);
xCI = zeros(Nd,nSymbols);
xWL = zeros(2*Nd,nSymbols);
xWL = zeros(Nd,nSymbols);
d_hat = zeros(Nd,nSymbols);

% transmitter
% using modem class
d_bits = randi([0 1], Nd*n, nSymbols);  % generate bit stream
d_tilde = modulate(modObj,d_bits);  % map to ASK symbols

x = ifft(d_tilde,N);  % transform to time domain
APPENDIX B. MATLAB SCRIPTS

115 $x_e = [x(N-N_g+1:N,:); x]$; % add CP, nSymbols=1
116 $x_e = x_e(:);$  
117 $E_b = (x_e'*x_e)/(N_d*bps*nSymbols);$ % signal energy / bit
118
119 \% noise
120 \% noise power
121 $E_n = E_b * 10^{-SNR_{db}(r)/10};$
123 % noise
124 $m_{nr} = 0;$ % mean noise – real
125 $\text{var}_{nr} = E_n/2;$ % variance noise – real
126 $m_{ni} = 0;$ % mean noise – imaginary
127 $\text{var}_{ni} = E_n/2;$ % variance noise – imaginary
128 $\rho_n = 0;$
129 \% noise generator
130 $n = (\text{randn(size}(x_e,1),1)+1j*\text{randn(size}(x_e,1),1))\sqrt{E_n/2};$
131
132 \% transmission
134 $y = \text{filter}(h,1,x_e) + n;$
135
136 \% receiver
137 $y = \text{reshape}(y,N+N_g,nSymbols);$ % reshape in matrix form
138 $y = y(Ng+1:end,:);$ % remove CP
139 $y_{\tilde{}} = \text{fft}(y);$ % transform to frequency domain
140
141 \% ESTIMATION
142 \% covariance matrices
143 $C_{nm} = N \times E_n \times \text{eye}(N);$ % noise autocovariance matrix
145 $pC_{nm} = 0;$ % noise pseudo-autocovariance matrix
146 $C_{dy} = C_{dd} \times H_{\tilde{}}';$
148 $pC_{dy} = pC_{dd} \times H_{\tilde{}}';$
149 $aC_{dy} = [C_{dy} pC_{dy}; \text{conj}(pC_{dy}) \text{conj}(C_{dy})];$
150 $C_{yy} = H_{\tilde{}} \times C_{dd} \times H_{\tilde{}}' + C_{nm};$
152 $pC_{yy} = H_{\tilde{}} \times pC_{dd} \times H_{\tilde{}}' + pC_{nm};$
153 $aC_{yy} = [C_{yy} pC_{yy}; \text{conj}(pC_{yy}) \text{conj}(C_{yy})];$
154 \% LMMSE
156 $x_{CL} = C_{dy} \times C_{yy}^{-1} \times y_{\tilde{}};$
157 $e_{CL} = d_{\tilde{}} - x_{CL};$
158 \% e_{CL} = \text{real}(e_{CL}); % only error of real part
159 \% WMMSE
160 $x_{WL} = aC_{dy} \times aC_{yy}^{-1} \times [y_{\tilde{}}; \text{conj}(y_{\tilde{}})];$
xWL = xWL(1:Nd,:);
eWL = d_tilde*xWL;

% channel inversion
xCI = diag(1./fft(h,N)) * y_tilde;
eCI = d_tilde-xCI;

% eCI = real(eCI);  % only error of real part

% demod
% modem class
d_wl_bits = demodulate(demodObj,xWL);
d_cl_bits = demodulate(demodObj,xCL);
d_ci_bits = demodulate(demodObj,xCI);

% FAILURE RATES
% bit failures
bf_wl = bf_wl + abs(d_bits - d_wl_bits);
bf_cl = bf_cl + abs(d_bits - d_cl_bits);
bf_ci = bf_ci + abs(d_bits - d_ci_bits);

end  % data burst

BER_wl(r,ch) = sum(bf_wl(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);
BER_cl(r,ch) = sum(bf_cl(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);
BER_ci(r,ch) = sum(bf_ci(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);

% Squared Error (Magnitudes)
% LMMSE
Cee_cl = 1/size(eCL,1)*eCL'*eCL;
MSE_cl(r,ch) = trace(Cee_cl);

% WMMSE
Cee_wl = 1/size(eWL,1)*eWL'*eWL;
MSE_wl(r,ch) = trace(Cee_wl);

% CI
Cee_ci = 1/size(eCI,1)*eCI'*eCI;
MSE_ci(r,ch) = trace(Cee_ci);

end  % channel
end  % SNR

mBER_cl(:,m) = mean(BER_cl,2);
B.4 MATLAB script for UW-OFDM

This appendix section shows the MATLAB script for the UW-OFDM example in Section 5.2.

% % initial parameters
nSymbols = 1e3;  % number of OFDM symbols to transmit at once

N = 64;         % DFT length
Nd = N;         % length of data symbols
Ng = 16;        % length of CP
% load IEEE channels
H = load('chIEEE_100ns.mat');

% SNR
SNR_db_min = 0;
SNR_db_max = 35;
SNR_db = SNR_db_min:SNR_db_max;

% constellation sizes for ASK modulation
const_min = 1; % minimum constellation size 2^x
const_max = 4; % maximum constellation size 2^x

% channels
sim_channels = 20; % simulated channel loops

% data transmission bursts per channel
data_bursts = 5;

% preallocate memory
BER_cl = zeros(length(SNR_db), sim_channels);
mBER_cl = zeros(length(SNR_db), 4);
BER_wl = zeros(length(SNR_db), sim_channels);
mBER_wl = zeros(length(SNR_db), 4);
BER_ci = zeros(length(SNR_db), sim_channels);
mBER_ci = zeros(length(SNR_db), 4);
SER_cl = zeros(length(SNR_db), sim_channels);
SER_wl = zeros(length(SNR_db), sim_channels);
SER_ci = zeros(length(SNR_db), sim_channels);
mMSE_wl = zeros(length(SNR_db), 4);
mMSE_ci = zeros(length(SNR_db), 4);
sim_time = 0;

% algorithm
for m = const_min : const_max % constellation size loop
    display(['num2str(',2^m,')', '-ASK']);
end

mPAM = 2^m; % select pam alphabet
bps = log2(mPAM); % bits / symbol
bpb = Nd * nSymbols * bps; % bis / burst
APPENDIX B. MATLAB SCRIPTS

% create modulator object
modObj = modem.pammod('M', mPAM, 'SymbolOrder',
    'binary', 'InputType', 'Bit');
demodObj = modem.pamdemod(modObj, 'OutputType', 'Bit');

% covariance matrices for data stream
var_d = (pammod(0:mPAM-1,mPAM)*pammod(0:mPAM-1,mPAM))'/mPAM;
pvar_d = (pammod(0:mPAM-1,mPAM)*pammod(0:mPAM-1,mPAM).')/mPAM;
Cdd = var_d * eye(Nd);  % covariance matrix of d_tilde
pCdd = pvar_d * eye(Nd);  % pseudo covariance matrix of d_tilde

for r=(SNR_db_min:SNR_db_max-SNR_db_min+1)  % SNR loop
    display(['SNR: ', num2str(SNR_db(r))]);
end

for ch=1:sim_channels  % channel loop
    % channel selection
    % AWGN channel
    h = eye(Ng,1);
    plot title='CP-OFDM_ASK_-_AWGN_channel';

    % IEEE channels
    h = H.channels(1:Ng,ch);
    h = h / (sqrt(h'*h));  % normalize channel (no gain)
    plot title='CP-OFDM_ASK_-_chIEEE\_100ns';

    % random channel
    h = rand(Ng,1) +1j* rand(Ng,1);  % channel impulse response
    h = h / (sqrt(h'*h));  % normalize channel (no gain)
    plot title='CP-OFDM_ASK_-_random_channel';

    H_tilde = diag(fft(h,N));  % channel frequency response

    % preallocate memory and reset results
    bf_wl=0;
    bf_cl=0;
    bf_ci=0;

    for b=1:data_bursts  % data transmission bursts
        % preallocate memory and reset results
        d=zeros(Nd,nSymbols);
        d_tilde=zeros(Nd,nSymbols);
        d_bits=zeros(Nd*nSymbols,1);
APPENDIX B. MATLAB SCRIPTS

100 y = zeros(Nd+nSymbols, 1);
101 xCL = zeros(Nd, nSymbols);
102 xCI = zeros(Nd, nSymbols);
103 axWL = zeros(2*Nd, nSymbols);
104 xWL = zeros(Nd, nSymbols);
105 d_hat = zeros(Nd, nSymbols);
106
107 % transmitter
108 % using modem class
109 d_bits = randi([0 1], Nd*m, nSymbols); % generate bit stream
110 d_tilde = modulate(modObj, d_bits); % map to ASK symbols
111
112 x = ifft(d_tilde, N); % transform to time domain
113 xe = x(N-Ng+1:N,:); % add CP, nSymbols=1
114 xe = xe(:);
115 Eb = (xe'*xe)/(Nd*bps*nSymbols); % signal energy / bit
116
117 % noise
118 % noise power
119 En = Eb * 10^(-SNR_db(r)/10);
120 % noise
121 m_nr = 0; % mean noise – real
122 var_nr = En/2; % variance noise – real
123 m_ni = 0; % mean noise – imaginary
124 var_ni = En/2; % variance noise – imaginary
125 rho_n = 0;
126 % noise generator
127 n = (randn(size(xe, 1), 1) + 1j*randn(size(xe, 1), 1))*sqrt(En/2);
128
129 % transmission
130 y = filter(h, 1, xe) + n;
131
132 % receiver
133 y = reshape(y, Nd*Ng, nSymbols); % reshape in matrix form
134 y = y(Ng+1:end,:); % remove CP
135 y_tilde = fft(y); % transform to frequency domain
136
137 % ESTIMATION
138 % covariance matrices
139 Cmn = N * En * eye(N); % noise autocovariance matrix
140 pCmn = 0; % noise pseudo-autocovariance matrix
APPENDIX B. MATLAB SCRIPTS

Cdy = Cdd * H_tilde';
pCdy = pCdd * H_tilde.';
aCdy = [Cdy pCdy; conj(pCdy) conj(Cdy)];

Cyy = H_tilde.*Cdd.*H_tilde' + Cnn;
pCyy = H_tilde.*pCdd.*H_tilde.' + pCnn;
aCyy = [Cyy pCyy; conj(pCyy) conj(Cyy)];

% LMMSE
xCL = Cdy * Cyy^−1 * y_tilde;
eCL = d_tilde − xCL;
% eCL = real(eCL); % only error of real part

% WMMSE
axWL = aCdy * aCyy^−1 * [y_tilde; conj(y_tilde)];
xWL = axWL(1:Nd,:);
eWL = d_tilde−xWL;

% channel inversion
xCI = diag(1./fft(h,N)) * y_tilde;
eCI = d_tilde−xCI;
% eCI = real(eCI); % only error of real part

demod
% modem class
d_wl_bits = demodulate(demodObj,xWL);
d_cl_bits = demodulate(demodObj,xCL);
d_ci_bits = demodulate(demodObj,xCI);

% FAILUR RATES
% bit failures
bf_wl = bf_wl + abs(d_bits − d_wl_bits);
bf_cl = bf_cl + abs(d_bits − d_cl_bits);
bf_ci = bf_ci + abs(d_bits − d_ci_bits);

end % data burst

BER_wl(r, ch) = sum(bf_wl(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);
BER_cl(r, ch) = sum(bf_cl(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);
BER_ci(r, ch) = sum(bf_ci(:))/(Nd*log2(mPAM)*nSymbols*data_bursts);

% Squared Error (Magnitudes)
% LMMSE
APPENDIX B. MATLAB SCRIPTS

```matlab
Ceecl = 1 / size(eCL, 1) * eCL' * eCL;
MSEC_cl(r, ch) = trace(Ceecl);
% WLMSE
Ceewl = 1 / size(eWL, 1) * eWL' * eWL;
MSEC_wl(r, ch) = trace(Ceewl);
% CI
Ceeci = 1 / size(eCI, 1) * eCI' * eCI;
MSEC_ci(r, ch) = trace(Ceeci);

end % channel
end % SNR

mBER_cl(:, m) = mean(BER_cl, 2);
mBER_wl(:, m) = mean(BER_wl, 2);
mBER_ci(:, m) = mean(BER_ci, 2);

mMSE_cl(:, m) = mean(MSE_cl, 2);
mMSE_wl(:, m) = mean(MSE_wl, 2);
mMSE_ci(:, m) = mean(MSE_ci, 2);

end % constellation size

%% plots

% BER plots
figure('Name', 'BER over SNR');
semilogy(SNR_db, mBER_cl, 'x');
grid minor;
hold on;
semilogy(SNR_db, mBER_wl, '^');
semilogy(SNR_db, mBER_ci, '+');
title(plottitle); xlabel('E_b/E_n'); ylabel('BER');
legend('CL_2a', 'CL_4a', 'CL_8a', 'CL_16a',
       'WL_2a', 'WL_4a', 'WL_8a', 'WL_16a',
       'CI_2a', 'CI_4a', 'CI_8a', 'CI_16a');

% MSE plots
figure('Name', 'MSE over SNR');
semilogy(SNR_db, mMSE_cl, 'x');
grid on;
hold on;
semilogy(SNR_db, mMSE_wl, '^');
semilogy(SNR_db, mMSE_ci, '+');
title(plottitle); xlabel('E_b/E_n'); ylabel('E(|e|^2)');
legend('CL_2a', 'CL_4a', 'CL_8a', 'CL_16a',
```

'WL_2a', 'WL_4a', 'WL_8a', 'WL_16a',
'CI_2a', 'CI_4a', 'CI_8a', 'CI_16a');
Bibliography


