Technische Universität Berlin
Department of Aeronautics and Astronautics
Chair of Space Technology

Bachelor Thesis

Spectrum Sensing for the SALSAT Nanosatellite

Fabian Georg Peddinghaus
Matr.-Nr. 391140

March 26, 2021

Reviewer: Prof. Dr.-Ing. Dieter Peitsch
Co-Reviewer: Prof. Dr.-Ing. Friedel Gerfers
Advisor: Jens Großhans M.Sc.

Zusammenfassung

Durch die steigende Anzahl von Kleinstsatelliten in erdnahen Umlaufbahnen ist die Auslastung des für die Satellitenkommunikation verfügbaren Frequenzspektrums signifikant gestiegen. So kam es in jüngster Vergangenheit wiederholt zu Interferenzen und Störungen. Im Rahmen des Forschungsvorhabens SALSAT des Fachgebiets Raumfahrttechnik der TU Berlin wurde ein Satellit zur Messung und Analyse globaler Nutzungsdaten des Frequenzspektrums aus dem Orbit entwickelt. Eine besondere Herausforderung stellt dabei die Erfassung von Signalen und die Erkennung belegter Frequenzbänder dar.

Im Rahmen dieser Arbeit werden neuartige Methoden zur dynamischen Detektion freier Funkfrequenzen im Weltraum vorgestellt. Diese kombinieren breitbandige Spektralanalyse auf Grundlage von Fourier Transformationen mit modernen Rauschmessverfahren. Basierend auf der Eigenwertzerlegung der Kovarianzmatrizen empfangener Daten und informationstheoretischer Modelle werden Signal- und Rauschkomponenten effektiv voneinander getrennt. Durch die analytische Herleitung aus mathematischen und signaltheoretischen Modellen können die Eigenschaften und das Leistungsverhalten der präsentieren Spektralmessalgorithmen präzise beschrieben werden. Mithilfe der eigens entwickelten Simulationsumgebung *SpecSens* werden Funktion und Stabilität der Algorithmen verifiziert. Die Ergebnisse zeigen, dass die hier vorgestellten Methoden auch in rauschbehafteten Umgebungen mit geringen Signal-Rausch-Verhältnissen hochwertige Detektionsergebnisse liefern.

Contents

List of Figures

List of Tables

Acronyms

ADC Analog to digital converter
CAF Cyclic autocorrelation function
CAV Covariance absolute value
CCDF Complementary cumulative distribution function
CDF Cumulative distribution function
CLT Central limit theorem
CR Cognitive radio
CS Compressed sensing
CSD Cyclic spectral density
CT Constraint
CWGN Complex white Gaussian noise
CWT Continuous wavelet transform
dB Decibel
DDR Double data rate
DFT Discrete Fourier transform
DLR German Aerospace Center
DSA Dynamic spectrum access
DWT Discrete wavelet transform
ED Energy detector
EDF Empirical distribution function
FFT Fast Fourier transform
FIR Finite impulse response
FWT Fast wavelet transform
EME Energy with minimum eigenvalue
FPGA Field-programmable gate array
IEEE Institute of Electrical and Electronics Engineers
IID Independent and identically distributed
IPU Image processing unit
IR Interface requirement
ITU International Telecommunication Union
I/Q In-phase and quadrature
KDE Kernel density estimation

LEO Low earth orbit
LO Local oscillator
LPF Lowpass filter
LRT Likelihood-ratio test
MC Monte Carlo
MDL Minimum description length
MIMO Multiple-input and multiple-output
MLE Maximum likelihood estimator/estimate
MME Maximum-minimum eigenvalue
MR Mission requirement
MSE Mean squared error
MT-PSD Multitaper power spectral density
PDF Probability density function
PSD Power spectral density
RF Radio frequency
ROC Receiver operating characteristic
SALSA Spectrum Analysis of LEO Satellite Allocations
SALSAT Spectrum AnaLysis SATellite
SDR Software-defined radio
SIMD Single instruction multiple data
SNR Signal-to-noise ratio
SPI Serial Peripheral Interface
STFT Short-time Fourier transform
SVD Singular value decomposition
UHF Ultra high frequency
UTC Coordinated Universal Time
VHF Very high frequency
WED Wideband energy detector
WGN White Gaussian noise
WLAN Wireless Local Area Network
WMP Wavelet multiscale product
WRAN Wireless Regional Area Network
WSS Wide-sense stationary
WT Wavelet transform
WTMM Wavelet transform modulus maxima

1. Introduction

1.1. Motivation

Radio-frequency spectrum is one of the most important resources required for wireless communication. Nearly all satellites require at least one, often multiple, radio frequency bands for inter-satellite and ground transmissions [1]. With an ever-increasing amount of satellite launches, especially in recent times, the potential for harmful radio interference between satellites has risen substantially [2, 3]. This situation is further exacerbated by so-called *mega-constellations* [4], consisting of hundreds or even thousands of satellites. Prominent examples are SpaceX's *Starlink* [5] and Amazon's *Project Kuiper* [6].

As radio spectrum occupation grows, static spectrum allocation becomes increasingly difficult [7]. Traditional spectrum coordination, as has been done by the International Telecommunications Union (ITU) [8], will not suffice for the satisfaction of increasing spectrum demand. With the radio spectrum already heavily utilized and more users to come, new and more advanced spectrum allocation techniques are necessary.

To address the challenge of increasing spectrum occupancy, the nanosatellite mission SALSAT (Spectrum AnaLysis SATellite) of Technische Universität Berlin (TU Berlin) and the German Aerospace Center (DLR) was conceived [9]. It aims to investigate and map the radio frequency spectrum usage in orbit and around the world. Of particular interest are the VHF and UHF amateur bands, along with the S-band scientific bands (see table 1.1). As such, SALSAT provides the basis for further research and development as well as for future missions with more advanced spectrum exploration systems.

1.2. Contributions and Organization

In this thesis, several methods for so-called *spectrum sensing* are presented and investigated with respect to application on SALSAT. By analyzing and comparing their properties, appropriate sensing algorithm candidates are selected, implemented, and further scrutinized through simulations. All simulations are conducted using *SpecSens*, a modular spectrum sensing simulation framework developed in the course of this thesis. SpecSens enables seamless development and modeling of spectrum sensing methods by providing synthetic data streams and useful functions, thus aiding in comparing and verifying spectrum sensing performance. With the help of SpecSens, a novel spectrum sensing method combining wideband energy detection and eigenvalue-based noise estimation is developed. Novel performance metrics for the proposed algorithm are analytically derived

Band	Frequencies
VHF	$145.80 - 174.00\,\mathrm{MHz}$
UHF	$400.15 - 420.00\,\mathrm{MHz}$
UHF	$435.00 - 438.00\,\mathrm{MHz}$
S-band	$2075.00 - 2095.00\,\mathrm{MHz}$
S-band	$2255.00 - 2275.00\,\mathrm{MHz}$

Table 1.1.: Frequency bands of interest to SALSAT [9].

and evaluated through simulations. The conducted simulations suggest that the proposed algorithm can effectively detect signals in wideband data and thus improve cognitive radio transmission throughput.

Chapter 2 starts by introducing the necessary mathematical and theoretical background. Because of the subject's vastness, it only focuses on theoretical concepts directly applicable to SALSAT, while more generic mathematical topics are discussed in appendix A. Chapter 3 builds on existing work and presents the most relevant state-of-the-art spectrum sensing methods. Particular emphasis is placed on noise estimation methods in the context of wideband spectrum sensing. Less promising spectrum sensing methods are detailed in appendix B. Chapter 4 presents the modular simulation framework SpecSens and discusses its internals. This is followed by the implementation and simulation of the developed spectrum sensing algorithms and their performance metrics in chapter 5. A conclusion and outlook on future work with special emphasis on hardware acceleration and embedded real-time applications are presented in chapter 6.

1.3. Background

1.3.1. SALSAT

SALSAT is a nanosatellite based on the TUBiX10 bus developed by TU Berlin (see fig. 1.1) [11–13]. Its primary payloads consist of the SALSA spectrum analyzer and the *Image Processing Unit* (IPU). The IPU is primarily used for visual positioning and attitude control. However, it can also be used as a general computing platform. It features an NXP i.MX 7 32-bit ARM Cortex-A7 processor, supporting NEON SIMD vector instructions and running at up to 1 GHz [14]. The system software is built on top of a Linux-based *Yocto* operating system. The SALSA spectrum analyzer (see fig. 1.2) is based on the open source LimeSDR project [15]. It utilizes *software-defined radio* (SDR) technology, modified and adapted for use in space. The SDR front-end consists of a field-programmable and fully re-configurable Lime Microsystems LMS7002M RF transceiver [15]. It down-converts incoming RF signals to baseband and generates I/Q data samples, which are transferred via a 12-bit parallel interface to an Intel Cyclone IV field-programmable gate array (FPGA) [16]. The FPGA is connected to 1 Gbit of redundant DDR2 memory and an STM32F4 microcontroller via the Serial Peripheral Interface (SPI). The STM32F4 microcontroller can be used for data processing and distribution to other parts of the satellite's systems. The IPU and SALSA subsystems exchange data via the satellite's system bus. With the help of custom ground control software and high bandwidth communication links, SALSAT's software can be fully reconfigured in flight [13]. This makes SALSAT a versatile platform for spectrum occupancy research in space.

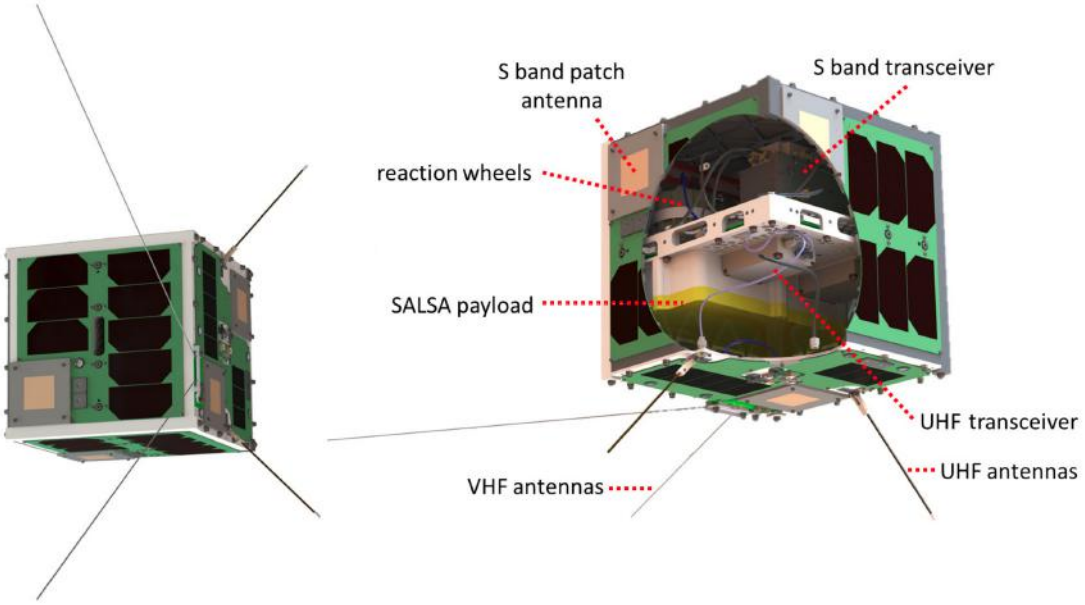

Figure 1.1.: Artists impression of SALSAT and subsystems [10].

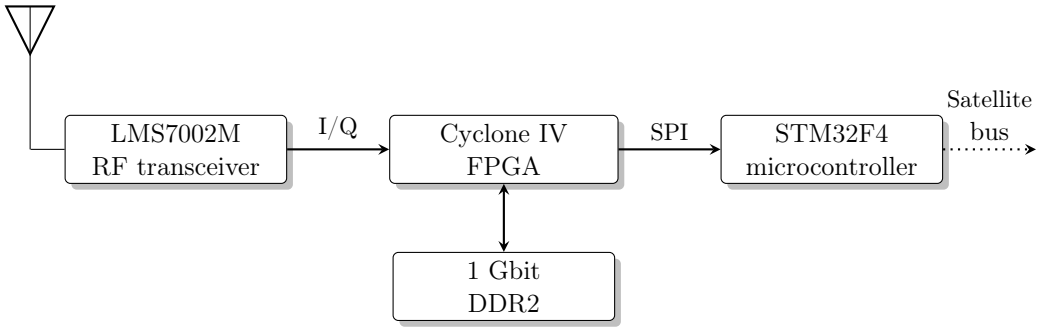

Figure 1.2.: Simplified SALSA spectrum analyzer block diagram (based on [10]).

SALSAT was successfully launched into a 575 km sun-synchronous orbit onboard a Soyuz rocket from the Plesetsk spaceport in north-west Russia on September 28, 2020 at 11:20 UTC [12]. First radio communication contact with the TU Berlin Mission Control Center was at 23:10 UTC. However, no usable spectrum I/Q data is yet available, as SALSAT was still in commissioning phase during the work on this thesis.

1.3.2. Cognitive Radio

Cognitive radio (CR) is a generic term that was first proposed by J. Mitola in 1999 [17]. It is used to describe a radio system that is aware of its environment and capable of adapting its transmissions according to the spectrum usage in its vicinity [18]. It aims to increase spectrum utilization while minimizing congestion and interference with other spectrum users. CRs utilize multiple sources of knowledge, including, but not limited to: radio spectrum usage, geolocation, protocols, and spatial or temporal patterns [19]. With the help of advanced algorithms, often employing machine-learning-based reasoning, this knowledge is further processed and transformed into actions [20].

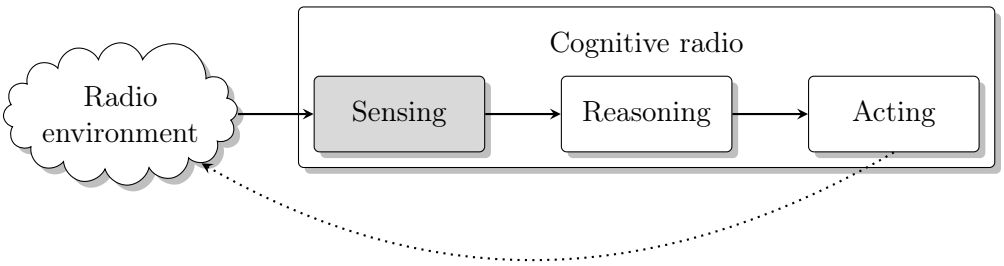

Figure 1.3.: High level cognitive radio flow.

Such cognitive behavior can be extended to networks of radios [20–25]. By exchanging information and coordinating spectrum access, CR networks are able to further improve *dynamic spectrum access* (DSA).

The subject of this thesis is limited to the information-gathering part of the CR flow. More specifically, it will only investigate the radio environment as an information source (see fig. 1.3). Radio spectrum sensing is deemed the most promising and robust source of spectrum usage information [26–28], as it directly measures the medium of interest. Moreover, many other information sources, such as geolocation, are generally not available in satellite communication applications [7].

1.3.3. Spectrum Sensing

The term *spectrum sensing* describes the task of gathering cognition about the radio environment [25, 29–34]. This is primarily done through antennas and sensors, which receive radio waves and convert them into digital signals to be used for further processing and analysis. Possible insights about signals in the radio environment, acquired through spectrum sensing, include signal power, duration, frequency, and modulation type.

In the context of this thesis, spectrum sensing is further narrowed to the task of reliably finding spectrum opportunities, also known as spectrum holes or white spaces (see fig. 1.4) [19, 22]. Spectrum opportunities are defined as areas in the spectrum that are currently not occupied and available for use. They are bound by frequency, time, and location, meaning that they only exist in a specific frequency range, for a certain time span, and at a particular location in space.

The performance of spectrum sensing algorithms may be severely degraded by shadowing, fading, and low signal-to-noise ratios (SNRs). To alleviate some of these factors, cooperative spectrum sensing has been proposed [22, 35]. It can enhance spectrum sensing performance through spatial diversity. In cooperative spectrum sensing, multiple cognitive users obtain and share information about the spectrum environment in a decentralized and cooperative way. The concept of cooperative spectrum sensing is closely related to CR networks (see subsection 1.3.2). While cooperative spectrum sensing is desirable, it is not applicable here, as SALSAT is currently a standalone system. This places considerable demands on the spectrum sensing algorithms deployed on SALSAT (see section 1.4).

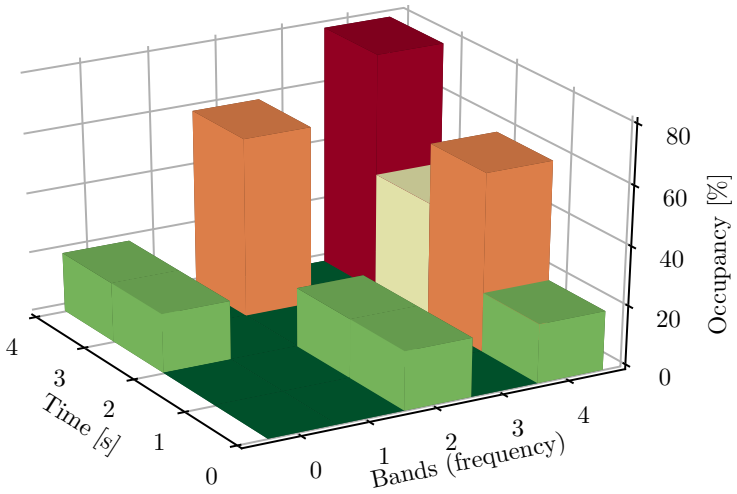

Figure 1.4.: Illustrative spectrum occupancy over time. 0% occupancy indicates a spectrum hole.

1.3.4. Related Work

With cognitive radio being a promising technology for future spectrum use, the amount of research that has been conducted is extensive. While the available literature provides a rigorous treatment of theoretical concepts, it often fails to provide insights with practical relevance. A considerable part of existing literature does not build on real-world scenarios, thereby neglecting important aspects of practical spectrum sensing applications [36–38].

A prevalent, space-related example is the *CoRaSat* project funded in part by the European Commission [39]. It span over a time of three years (2012-2015), involved five European universities, and had a budget of more than four million Euros. However, besides a dozen publicly available and theoretically oriented publications [40–42], very little practically relevant research, such as spectrum sensing algorithms or software solutions, has come out of this project. Needless to say, important work on business-, regulatory-, and standardization-related topics has been conducted. While this aids in providing a framework for CR in society, it does not directly advance CR and spectrum sensing as a viable technology for real-world applications.

To a lesser extent, this phenomenon is also reflected in the field of spectrum sensing. Originating from the mid-20th century work on detection theory for radar signals [43, 44], spectrum sensing, with respect to DSA and CR, has established itself as an independent discipline in the late 1990s and early 2000s. This lead to an increased interest by the research community and consequently to the development of more specialized spectrum sensing methods. In 2004, the IEEE 802.22 working group on Wireless Regional Area Networks (WRAN) was established [45–48]. It utilized rapid advances in the fields of CR and spectrum sensing and published the first cognitive radio-based standard for DSA in 2011. Other standards, such as the IEEE 802.11af Wireless Local Area Network (WLAN) standard, published in 2014, followed [49–51]. However, most CR standards primarily rely on geolocation information (IEEE 802.22 allows for some spectrum sensing approaches). Additionally, these standards only lay out guidelines and requirements for spectrum sensing but do not provide any actual algorithms. Thus, they have only seen minuscule adaption in real-world applications. As

already hinted at, this is also likely caused by a lack of application-oriented research in spectrum sensing. While there exists some publicly available software written in Python [52] and C++ [53], it is very rudimentary, poorly documented, and focuses on other aspects of the CR flow (see fig. 1.3). Without robust and well-tested spectrum sensing algorithms or frameworks available for academia and industry, the practical adoption of spectrum sensing and CR is not foreseeable.

1.3.5. Space-related Challenges

The ITU [8], as the main regulatory body for the shared global use of radio spectrum, facilitates international cooperation by assigning satellite orbits and radio frequencies. While the ITU pursues the timely and efficient coordination of the radio spectrum from a regulatory perspective, the general practice, as conducted by the spectrum users (embodied through governments and corporations), often deviates from the notified use to the ITU [7]. Combined with a myriad of international interests represented in space-related spectrum usage, increasing disorganization of actual spectrum utilization will likely cause severe interference in the near future. This situation is further exacerbated by the ever-increasing demand for radio spectrum, especially in recent times. So-called *mega-constellations* [4], consisting of hundreds or even thousands of satellites, such as SpaceX's *Starlink* [5] and Amazon's *Project Kuiper* [6], require vast amounts of radio-frequency spectrum [7]. The influence of terrestrial interference on satellite communications further complicates this situation. Most satellites cross dozens of countries (each with different frequency spectrum regulations) multiple times per day, often receiving radio signals from numerous regions simultaneously. Such diversified radio usage makes the development of space-borne spectrum sensing systems an intricate process.

1.4. Formal Requirements

With the basic background information presented and the general task laid out, more concrete specifications for the spectrum sensing system on SALSAT must be devised. So-called *formal requirements* aid in aligning and substantiating goals. They guide the development process and help to verify development results. The following section motivates such formal requirements with respect to spectrum sensing on SALSAT.

The requirements will be mainly grouped into *mission requirements* (MR), which are similar to functional requirements, and *constraints* (CT), which define non-functional aspects. Additionally, as the system will be tightly integrated into an existing architecture, specialized *interface requirements* (IR) will be defined. While this grouping can be ambiguous at times, it helps to clarify the development process.

As already mentioned, spectrum usage in space is very complex and highly diversified. Different technologies, from different entities and technology generations, are operating and utilizing spectrum resources in space. Therefore, there is no single radio frequency or telecommunication standard in use. Spectrum sensing systems cannot rely on any specific property of radio signals and must be able to sense and detect using characteristics shared by all radio signals; they must be *blind* to signal specific features and be able to sense all types of signals (MR-01). Similarly, no assumptions about

background noise can be made in advance, as the noise environment in space is very dynamic [7]. Thus, spectrum sensing systems must cope with unknown and varying background noise levels (CT-02).

SALSAT aims to investigate spectrum bands with a width of more than 20 MHz (see table 1.1). Consequently, *wideband* spectrum sensing methods are required, monitoring spectrum occupancy over wide frequency ranges and providing detailed information on individual subbands (MR-02). Because the measurement of high bandwidth signals requires high data rates, the system must handle excessive data throughput; while simultaneously providing spectrum sensing results hundreds of times per second (CT-04). Frequency ranges, resolution, probability of detection, probability of false alarm, and other relevant parameters have to be dynamically configurable (MR-03). This allows for on-demand and dynamic operation, as required by research-oriented cognitive radio applications.

To reduce the chance of possible interference, the spectrum sensing system must be able to detect radio signals with very low latency, minimizing the time of simultaneous transmissions (CT-05). Because of channel shadowing and multipath fading, it is generally difficult to distinguish between noise and signals. Fading and shadowing may result in the *hidden terminal problem*, where the spectrum sensing system cannot detect a signal source and therefore cause harmful interference by transmitting in the primary frequency band [28]. Very sensitive receivers can alleviate this problem. It is thus of paramount importance that the spectrum sensing system can operate at low SNR (CT-01). Because spectrum interference needs to be avoided at all times, the spectrum sensing system must be robust and reliable (CT-03)

Since SALSAT is a standalone system, it will operate without networking or cooperation with other systems. Consequently, the spectrum sensing algorithms must work independently and cannot rely on information from external platforms (CT-07). Additionally, only a single antenna, analog front-end channel, and data stream may be used by the spectrum sensing algorithms. SALSAT actually provides multiple antennas and analog channels, but due to other measurement equipment running in parallel and because of maximum power consumption constraints, only one of them is available for use (IR-02). As illustrated in fig. 1.2, the analog front-end provides a continuous stream of I/Q data. This constitutes the signal input to the spectrum sensing system (IR-01). Due to resource constraints (power consumption, space, weight, etc.), only small embedded computing systems are available on SALSAT. Thus, the spectrum sensing algorithms must operate with limited processing and memory resources (CT-06).

In order to demonstrate conformance, the requirements and constraints must be verified through one or more of the following *verification methods* [54]:

1. **Analysis** (A) - Performing theoretical or empirical evaluation.

2. **Simulation** (S) - Validation of performance and function under various simulated environments using simulated data samples.

3. **Target simulation** (T) - Measuring performance and function of target implementation using real data samples recorded in space.

Req.-Number	**Content of requirement**	**Verif.**
MR-01	The system shall be able to sense all types of signals without prior knowledge of signal properties.	A,S
MR-02	The system shall be able to sense over a wide frequency range simultaneously with configurable frequency resolution.	A,S
MR-03	Desired performance metrics, such as the probability of false alarm and probability of detection, shall be configurable.	A
CT-01	The system shall be able to sense signals with low SNR.	S
CT-02	The system shall be able to sense in environments with unknown and varying background noise.	A,S
CT-03	The system shall be robust, stable, and reliable, meaning that it must stay operational in all sensing scenarios.	S,T
CT-04	The system shall operate at high data rates and with high throughput. Spectrum occupancy information shall be available hundreds of times per second.	A,S,T
CT-05	The system shall operate with very little latency so that minimal time passes between actual spectrum occupancy change and detection.	A,S,T
CT-06	The system shall operate in a constrained resource environment; with respect to computing power, memory footprint, and power consumption.	A,S,T
CT-07	The system shall sense as a standalone entity and without the need for networked or cooperative operation.	A,S
IR-01	The system shall use a continuous I/Q data stream as the (only) information input.	A
IR-02	The system shall be able to sense with a single antenna, RF chain, and data stream.	A

Table 1.2.: Formal requirements for the spectrum sensing system on SALSAT.

2. Theoretical Background

This chapter is going to introduce the necessary theoretical background required for the design of spectrum sensing systems. Starting with a short overview of mathematical concepts, the employed channel model and the most important theoretical aspects of signal detection will be presented.

2.1. Mathematical Prerequisites

Spectrum sensing, detection theory, and statistical signal processing rely heavily on mathematical tools and models. This thesis and the presented algorithms make use of the following statistical distributions: multivariate and complex Gaussian (section 3.1.1), chi-square (section 3.1.1), Tracy-Widom (section 3.1.2), and Marchenko-Pastur (section 3.3). Spectral analysis, in terms of eigenvalue decomposition of matrices (section 3.1.2 and section 3.3), as well as spectral density estimation of complex time signals using the Fourier series (appendix B.1.3), Fourier transform, discrete Fourier transform, fast Fourier transform, and short-time Fourier transform will be employed (section 3.2). Additionally, the wavelet transform is used for singularity detection in section 3.2.4. Because of the breadth of the discussed methods and to best accommodate a variety of readers, only a very brief overview of the most important concepts will be given in this section and throughout this thesis. For formal definitions and a general presentation of the mentioned concepts, consult appendix A.

2.2. Signal Model

Communicating information from one location to another requires some form of pathway or medium, also known as *channel*. In the design of communication systems, it is convenient to construct mathematical models that reflect the fundamental characteristics of the transmission medium. A simple but powerful model is the *linear time-variant filter channel*, which, as the name suggests, changes its characteristics over time $t \in \mathbb{R}$ [55, 56]. Its channel response $h(\tau, t)$ models small-scale effects, like multipath fading, as well as larger-scale effects, such as path loss via distance attenuation and shadowing by obstacles [57]. This model is augmented with additive noise $w(t)$. Noise can come from various natural sources, such as the thermal vibrations of atoms or celestial sources, e.g., the Sun [56, 58]. Electronic components and amplifiers can also introduce noise at the receiver. The *central limit theorem* (CLT) suggests that this type of noise can be characterized statistically as a Gaussian noise process [59]. Furthermore, because its power is uniformly distributed across the frequency band, just as white light is uniformly distributed across the visible spectrum, this type of noise is called *white Gaussian noise* (WGN) [60]. Hence, when a signal $s(t)$ is transmitted through a channel $h(\tau, t)$ (see fig. 2.1), the channel output $x(t)$, which is the received signal or more generally

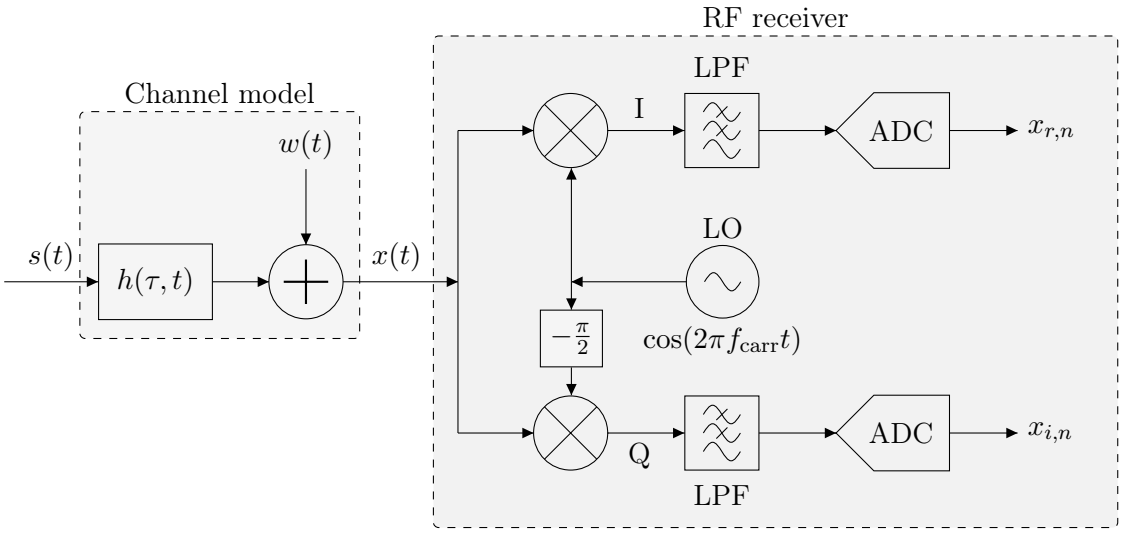

Figure 2.1.: Simplified SALSAT channel model and RF receiver block diagram (based on [55] and [15]).

the received data, can be derived as

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} h(\tau,t)s(t-\tau)d\tau + w(t) \\ &= h(\tau,t) * s(t) + w(t), \end{aligned} \tag{2.1}$$

where $*$ denotes the convolution of two functions [55]. This concept can be generalized to multiple-input multiple-output (MIMO) systems, yielding a $K \times P$ channel response matrix $\mathbf{H} = (h_{i,j}(\tau,t))$ and a $K \times 1$ additive noise vector $\mathbf{w}$, with P denoting the number of signal sources and K the number of receivers respectively, so that $\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{w}(t)$ [61–63].

While this channel model is often chosen for terrestrial links, it can be simplified for many satellite and space communication applications to only include additive WGN; because multipathing, terrain blocking, ground clutter, etc., are less common in space [7]. The channel response is replaced with a Dirac-impulse $h(\tau,t) = \delta(\tau,t)$, which can be left out [64]. As there is only a single antenna and data stream available on SALSAT (IR-02), the signal model relevant to this thesis is

$$x(t) = s(t) + w(t). \tag{2.2}$$

After the RF signal $x(t)$ is received, the first major step at the receiver (of interest to the work in this thesis) is *frequency shifting*, where the signal is converted to the equivalent baseband in-phase I and quadrature Q components. This is accomplished through multiplication with a local oscillator (LO), which produces cosine and sine waves of frequency $2\pi f_{\text{carr}}$ (see fig. 2.1). f_{carr} is the so-called *carrier frequency* with which the signal was originally modulated (see table 1.1) [55]. After I/Q demodulation, the signal (now at baseband) is lowpass filtered (LPF), leaving only frequencies in the range of $[-B/2, B/2]$, where B represents the signal bandwidth (see fig. 2.2). Finally, the signal is sampled at (or above) the Nyquist rate $f_s > 2B$ and digitized by an analog to digital converter (ADC), which converts analog signals into digital signals. The digital I and Q signals can be thought of as the real and imaginary parts of a complex number $x \in \mathbb{C}$, where $I = \text{Re}(x)$ and $Q = \text{Im}(x)$, respectively[1] (IR-01) [56]. The resulting discrete baseband signal x_n with discrete-time $n \in \mathbb{N}$, a

[1] A complex number $z \in \mathbb{C}$ is written as $z = z_r + iz_i$ where $i = \sqrt{-1}$. $z_r = \text{Re}(z)$ and $z_i = \text{Im}(z)$ refer to the real and imaginary parts respectively. The complex conjugate is written as $z^* = z_r - iz_i$.

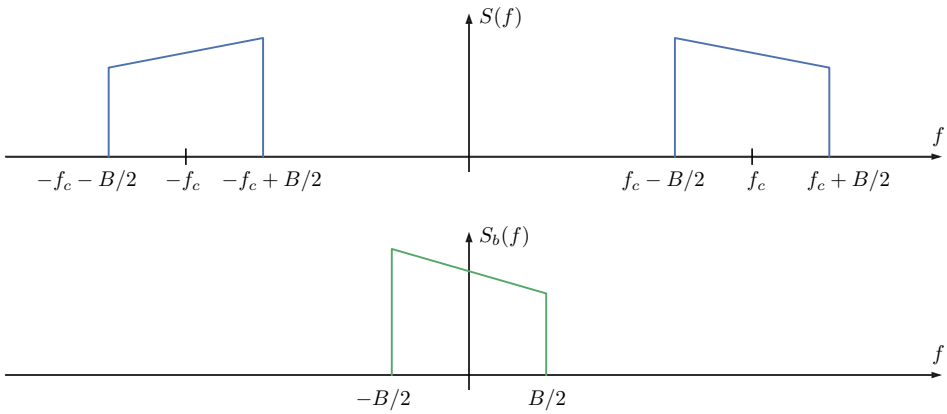

Figure 2.2.: Illustration of the relationship between a passband spectrum $S(f)$ and its baseband equivalent $S_b(f)$. The passband spectrum (blue) is down-converted to baseband (green) and subsequently centered around near-zero frequencies (derived from [55]).

complex-valued representation of the real-valued modulated physical signal $x(t)$, can be represented by its discrete signal model

$$x_n = s_n + w_n, \tag{2.3}$$

where s_n and w_n are the discrete-time complex signal and noise counterparts of $s(t)$ and $w(t)$, respectively.

2.3. Detection Theory

Signal detection, in the context of this work, involves deciding whether a signal is present or not in a set of noisy observations [44, 60, 65]. In contrast to signal (or parameter) estimation, where high-quality estimates are desired, signal detecting concerns itself with the more specific task of detecting weak signals buried in noise. A signal detector has to distinguish between two possible hypotheses: $\mathcal{H}_0$ (noise only) and $\mathcal{H}_1$ (signal and noise). The detector has to make this decision based on the data samples it receives.

Let the received input to the detector be some N-dimensional complex data vector $\mathbf{x} = [x_1, \ldots, x_N]^{\mathrm{T}} \in \mathbb{C}^N$. With the results from section 2.2, the detection problem becomes

$$\mathcal{H}_0 : \quad \mathbf{x} = \mathbf{w} \tag{2.4}$$

$$\mathcal{H}_1 : \quad \mathbf{x} = \mathbf{s} + \mathbf{w}, \tag{2.5}$$

where $\mathbf{s} = [s_1, \ldots, s_N]^{\mathrm{T}} \in \mathbb{C}^N$ is the signal the detector is trying to detect, $\mathbf{w} = [w_1, \ldots, w_N]^{\mathrm{T}} \in \mathbb{C}^N$ is the additive *complex white Gaussian noise* (CWGN) [60] and N is the length of the sample vectors. If not stated otherwise, the additive CWGN will be assumed to be zero-mean and circularly symmetric (see appendix A.1.1), as well as independent and identically distributed (IID) with variance $\mathrm{Var}[\mathbf{w}] = \sigma_w^2$. IID means that any two data points $x_n, x_m \in \mathbf{x}$ do not statistically depend on each other and that they all come from the same underlying statistical distribution. It is common to

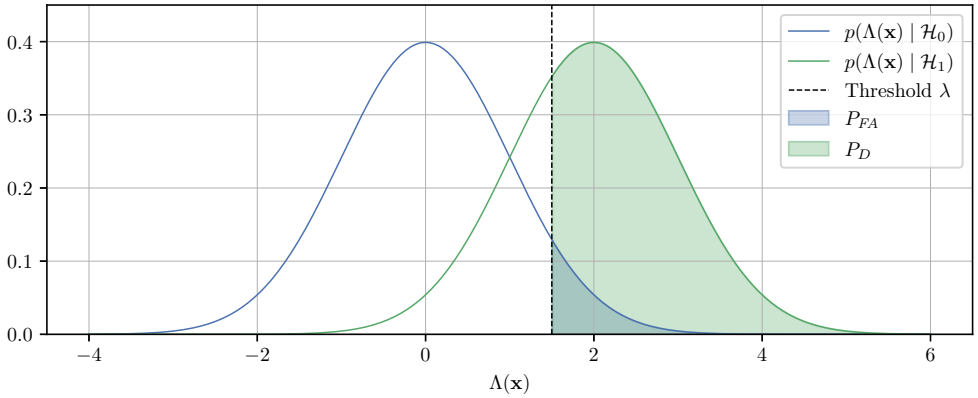

Figure 2.3.: Illustrative probability density functions (PDFs) of $\Lambda(\mathbf{x})$.

write $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}\right)$, where $\mathbf{0}$ and $\mathbf{I}$ are the zero vector and identity matrix of size N, respectively. The statistical model for $\mathbf{s}$ will be defined as needed, as different models aid in examining the diverse properties of the spectrum sensing algorithms investigated (see chapter 3 and appendix B).

There are mainly two strategies available for constructing signal detection algorithms. The first is the so-called *Bayes risk* criterion [26, 60, 66]. It consists of assigning a cost to each outcome of the decision process and subsequently minimizing the total expected cost, termed *risk*. However, this requires knowledge of the prior probabilities of the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. Since this information is not available (MR-01), the so-called *Neyman-Pearson* criterion will be used hereafter [26, 60]. It works by maximizing the *probability of detection* P_D under the constraint that the *probability of false-alarm* is upper bounded by P_{FA}. The proof in appendix A.2 shows that the Neyman-Pearson detector using the likelihood-ratio $\Lambda(\cdot)$ is optimal under the here given conditions. The resulting *test-statistic* (or *test-criterion*) is called the *likelihood-ratio test* (LRT). The detector decides $\mathcal{H}_0$ if $\Lambda(\mathbf{x}) < \lambda$ and $\mathcal{H}_1$ if $\Lambda(\mathbf{x}) > \lambda$. That is

$$\Lambda(\mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{H}_1)}{p(\mathbf{x} \mid \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda, \tag{2.6}$$

where $p(a \mid b)$ denotes the conditional probability[2] of a given b, and $\mathbf{x}$ is the observed data vector[3]. The *decision threshold* can be found by rearranging the equation from the Neyman-Pearson theorem (see appendix A.2)

$$P_{FA} = \int_{\{\mathbf{x}:\Lambda(\mathbf{x})>\lambda\}} p(\mathbf{x} \mid \mathcal{H}_0) d\mathbf{x} = \int_{\lambda}^{\infty} p(\Lambda(\mathbf{x}) \mid \mathcal{H}_0) d\Lambda \tag{2.7}$$

so that λ is isolated. This will become more apparent when applied to actual detection methods in chapter 3.

[2] One could similarly call $p(a \mid b)$ the likelihood of b given a. This is why $\Lambda(\mathbf{x})$ is called the likelihood-ratio.

[3] The problem presented here is a classical decision theory problem. The signal vector $\mathbf{x}$ can be thought of as a point in an N-dimensional signal vector space of all possible signals. Every sample x_n of $\mathbf{x}$ is a feature used in the decision process. One thus needs to find a decision rule that divides the signal vector space into two subspaces, also called *decision regions*, separated by an $N-1$-dimensional *decision boundary*. The detection algorithm simplifies this process, as it maps a vector from the N-dimensional signal space down to 1 dimension so that the decision boundary becomes a simple threshold λ.

	Actual condition true ($\mathcal{H}_1$)	Actual condition false ($\mathcal{H}_0$)
Predict true	Probability of detection (P_D)	Probability of false alarm (P_{FA})
Predict false	Probability of miss (P_M)	Probability of true negative

Figure 2.4.: Confusion matrix for signal detection.

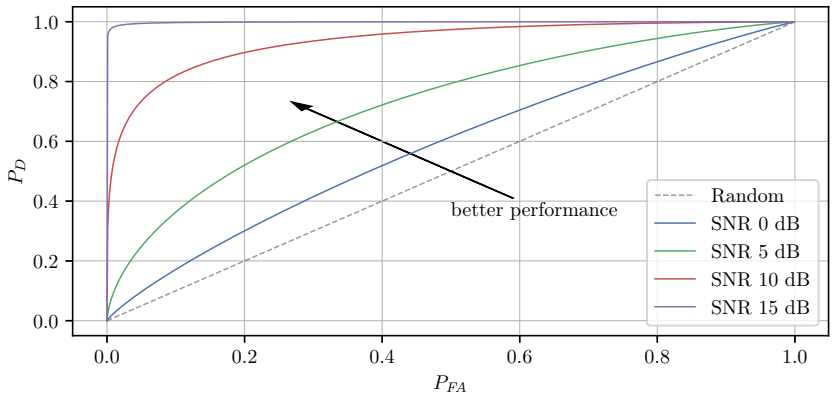

Figure 2.5.: Receiver operator characteristic (ROC) at different SNR values for an ideal energy detector with $N = 10$.

A detector does not always make correct decisions. To quantify the performance of the detector, it is necessary to formally introduce the above-mentioned probability of false alarm P_{FA} and probability of detection P_D. Together with the *probability of miss* P_M they are defined as [44, 60]

$$P_{FA} = p(\Lambda(\mathbf{x}) > \lambda \mid \mathcal{H}_0) = \int_{\lambda}^{\infty} p(\Lambda(\mathbf{x}) \mid \mathcal{H}_0) d\Lambda \tag{2.8}$$

$$P_D = p(\Lambda(\mathbf{x}) > \lambda \mid \mathcal{H}_1) = \int_{\lambda}^{\infty} p(\Lambda(\mathbf{x}) \mid \mathcal{H}_1) d\Lambda \tag{2.9}$$

$$P_M = p(\Lambda(\mathbf{x}) < \lambda \mid \mathcal{H}_1) = \int_{-\infty}^{\lambda} p(\Lambda(\mathbf{x}) \mid \mathcal{H}_1) d\Lambda = 1 - P_D. \tag{2.10}$$

A false alarm occurs when the detector predicts that a signal is present, even though there is no signal present. This is an unwanted error. The P_{FA} is usually chosen to be between 0.01 and 0.1 [45]. A (correct) detection occurs, as the name suggests, when the detector correctly detects that a signal is present. One aims to maximize the P_D, commonly required to be above 0.9 or 0.95 [45]. Their relation to one another can be seen in fig. 2.3, as well as in the so-called *confusion matrix* in fig. 2.4. These values can be traded off against each other by adjusting the threshold λ.

By keeping the sample size N constant, one can derive a function in the form of $P_D(P_{FA})$, which is commonly termed the *receiver operator characteristic* (ROC). It directly relates the probability of detection P_D to the probability of false alarm P_{FA} and is thus a powerful tool for visual performance comparison between detectors and models (see fig. 2.5). The diagonal in fig. 2.5, which represents a completely random detector, divides the ROC space. Curves in the upper left, above the diagonal, represent increasingly better classification results (better than random), while curves below the diagonal represent bad results (worse than random) [67].

The *signal-to-noise ratio* (SNR) [55] is defined as the ratio of the power of the signal $\mathbf{s}$ to the power of the noise $\mathbf{w}$

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} = \frac{\mathrm{E}[|\mathbf{s}|^2]}{\mathrm{E}[|\mathbf{w}|^2]} \overset{\mathrm{E}[\mathbf{w}]=0}{=} \frac{\mathrm{E}[|\mathbf{s}|^2]}{\sigma_w^2}, \tag{2.11}$$

where $\mathrm{E}[\cdot]$ denotes the expected value of a random variable. The last equality holds because the variance is defined as $\mathrm{Var}[Z] = \mathrm{E}[|Z|^2] - |\,\mathrm{E}[Z]|^2$ for a complex random variable Z [59]. In general, the higher the SNR, the easier it is to detect a signal. Because SNR values are usually expected to be very low (CT-01), detector design for space applications is challenging. Often it is more practical to express the power in a logarithmic form called *decibel* (dB) [64], which, for the power of signals, is defined as

$$P_{\mathrm{signal,dB}} = 10 \log_{10} \left(P_{\mathrm{signal}} \right) \tag{2.12}$$

$$P_{\mathrm{noise,dB}} = 10 \log_{10} \left(P_{\mathrm{noise}} \right) \tag{2.13}$$

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} \right) = 10 \log_{10} \left(\mathrm{SNR} \right). \tag{2.14}$$

3. Sensing Methods

This chapter introduces the most relevant state-of-the-art spectrum sensing methods by outlining their advantages and discussing their limitations. It will first focus on narrowband sensing, subsequently present wideband spectrum sensing methods, and conclude by introducing noise estimation techniques. Building on concepts presented in chapter 2, detection methods will be derived from a signal theoretic perspective. This is going to provide valuable insights into the strengths and weaknesses of detectors while allowing to analytically derive crucial performance metrics, which will later be verified through simulations in chapter 5. Only spectrum sensing methods applicable to SALSAT will be detailed. Less promising methods can be found in appendix B. A general overview of all methods will be provided at the end of each section.

While this chapter heavily relies on existing work in spectrum sensing, some of the here presented derivations and performance metrics are new and have not yet been described in related literature. These novel formulations allow for an analytical and quantitative approach to wideband spectrum sensing based on noise estimation, making it possible to model and predict spectrum sensing results in complex environments.

3.1. Narrowband Spectrum Sensing

Unlike wideband spectrum sensing, narrowband spectrum sensing is applied to a single spectral band. Subbands are not individually analyzed, and the detector does not differentiate between spectral components; the detection problem is thus entirely analogous to the one presented in section 2.3.

3.1.1. Energy Detection

The (narrowband) *energy detector* (ED), first conceived in [68], is one of the most popular detection schemes for spectrum sensing and related applications [69, 70]. It is used when the signal to be detected is completely unknown, and no assumptions about modulation type, communication protocol, or other signal properties can be made (MR-01). The ED is very susceptible to noise, hence the treatment of noise estimation methods in section 3.3. Because of its simplicity, attaining its performance metrics is relatively straightforward. However, since many different variants exist, one needs to be careful when using statistics available in literature. Thus, this section will first derive the ED equation in accordance with [60] using the LRT (see section 2.3), followed by the derivation of the performance statistics, using both exact (chi-square) and approximate methods (CLT).

Detector

Similar to the noise $\mathbf{w}$, which will be modeled as CWGN (see section 2.2), the signal $\mathbf{s}$ will also be characterized using a purely statistical model. It will be assumed that the signal is drawn from a complex Gaussian distribution with zero mean and covariance $\mathbf{C} = \sigma_s^2\mathbf{I}$. This assumption is, akin to Gaussian noise, motivated by the CLT. Hence, under $\mathcal{H}_0$ the received data samples will be modeled as $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \sigma_w^2\mathbf{I})$ and under $\mathcal{H}_1$ the received samples are $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, (\sigma_w^2 + \sigma_s^2)\mathbf{I})$, so that the PDF's under both hypotheses are

$$p(\mathbf{x} \mid \mathcal{H}_0) = \frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right) \tag{3.1}$$

$$p(\mathbf{x} \mid \mathcal{H}_1) = \frac{1}{\pi^N \left(\sigma_w^2 + \sigma_s^2\right)^N} \exp\left(-\frac{1}{\sigma_w^2 + \sigma_s^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right), \tag{3.2}$$

using the in appendix A defined complex multivariate Gaussian distribution. Constructing the LRT

$$\begin{aligned}\Lambda(\mathbf{x}) &= \frac{p(\mathbf{x} \mid \mathcal{H}_1)}{p(\mathbf{x} \mid \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda \\ &= \frac{\frac{1}{\pi^N \left(\sigma_w^2+\sigma_s^2\right)^N} \exp\left(-\frac{1}{\sigma_w^2+\sigma_s^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)},\end{aligned} \tag{3.3}$$

applying the natural logarithm $\ln(\cdot)$, and simplifying, results in

$$\begin{aligned}\ln\left(\Lambda\left(\mathbf{x}\right)\right) &= \ln\left(\frac{\frac{1}{\pi^N \left(\sigma_w^2+\sigma_s^2\right)^N} \exp\left(-\frac{1}{\sigma_w^2+\sigma_s^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}\right) \\ &= N \ln\left(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2}\right) - \left(\frac{1}{\sigma_w^2 + \sigma_s^2} - \frac{1}{\sigma_w^2}\right)\mathbf{x}^{\mathrm{H}}\mathbf{x} \\ &= N \ln\left(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2}\right) + \frac{\sigma_s^2}{\sigma_w^2(\sigma_w^2 + \sigma_s^2)}\mathbf{x}^{\mathrm{H}}\mathbf{x} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda'.\end{aligned} \tag{3.4}$$

After rearranging the terms, one has

$$\begin{aligned}\mathrm{T}(\mathbf{x}) &= \mathbf{x}^{\mathrm{H}}\mathbf{x} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \left(\lambda' - N \ln\left(\frac{\sigma_w^2}{\sigma_w^2 + \sigma_s^2}\right)\right) \frac{\sigma_w^2(\sigma_w^2 + \sigma_s^2)}{\sigma_s^2} \\ &= \mathbf{x}^{\mathrm{H}}\mathbf{x} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda'',\end{aligned} \tag{3.5}$$

which yields the final ED test-statistic

$$\mathrm{T}(\mathbf{x}) = \sum_{n=1}^{N} x_n^* x_n = \sum_{n=1}^{N} |x_n|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda'', \tag{3.6}$$

where the threshold λ'' will from here on just be referred to as λ. Because the data samples $\mathbf{x}$ are zero-mean, the detector estimates a scaled version of the variance $\mathrm{Var}[\mathbf{x}] = \sigma_x^2$, which can also be interpreted as the power of $\mathbf{x}$. Thus, the ED computes the energy of $\mathbf{x}$. Under $\mathcal{H}_0$ the variance is σ_w^2, while under $\mathcal{H}_1$ it increases to $\sigma_w^2 + \sigma_s^2$. Intuitively, if $\mathbf{s}$ is present, the energy of $\mathbf{x}$ increases, and when $\mathbf{s}$ is not present, it decreases. Therefore, unknown noise power levels σ_w^2 drastically influence the detector's performance.

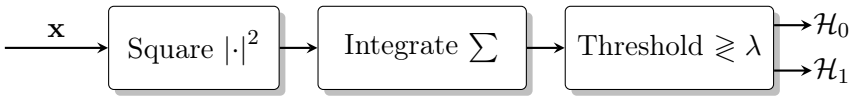

Figure 3.1.: ED block diagram (derived from [70]).

Chi-Square Statistics

Since the test-statistic $\mathrm{T}(\mathbf{x})$ (see eq. (3.6)) of the ED computes the sum of the squares of N IID complex Gaussian random variables, it follows a scaled chi-square distribution (see appendix A.1.2) with $2N$ degrees of freedom [60, 71]

$$\mathcal{H}_0: \quad \frac{2\,\mathrm{T}(\mathbf{x})}{\sigma_w^2} \sim \chi^2_{2N} \tag{3.7}$$

$$\mathcal{H}_1: \quad \frac{2\,\mathrm{T}(\mathbf{x})}{\sigma_w^2 + \sigma_s^2} \sim \chi^2_{2N}. \tag{3.8}$$

With its definition from section 2.3, one can derive the probability of false alarm P_{FA} for the ED as

$$\begin{aligned} P_{FA} &= p\left(\mathrm{T}(\mathbf{x}) > \lambda \mid \mathcal{H}_0\right) \\ &= \int_\lambda^\infty \chi^2_{2N}\left(\frac{2\,\mathrm{T}(\mathbf{x})}{\sigma_w^2}\right) d\,\mathrm{T} \\ &= 1 - F_{\chi^2_{2N}}\left(\frac{2\lambda}{\sigma_w^2}\right), \end{aligned} \tag{3.9}$$

where $F_{\chi^2_{2N}}$ is the CDF of χ^2_{2N}. Similarly, the probability of detection P_D is

$$\begin{aligned} P_D &= p\left(\mathrm{T}(\mathbf{x}) > \lambda \mid \mathcal{H}_1\right) \\ &= \int_\lambda^\infty \chi^2_{2N}\left(\frac{2\,\mathrm{T}(\mathbf{x})}{\sigma_w^2 + \sigma_s^2}\right) d\,\mathrm{T} \\ &= 1 - F_{\chi^2_{2N}}\left(\frac{2\lambda}{\sigma_w^2 + \sigma_s^2}\right). \end{aligned} \tag{3.10}$$

Given eq. (3.9), one can derive the threshold λ as a function of the P_{FA} simply as

$$\lambda = \frac{\sigma_w^2}{2} F^{-1}_{\chi^2_{2N}}\left(1 - P_{FA}\right). \tag{3.11}$$

To construct the ROC curve (see section 2.3) for the ED, one has to derive the following; starting with eq. (3.9)

$$\begin{aligned} P_{FA} &= 1 - F_{\chi^2_{2N}}\left(\frac{2\lambda}{\sigma_w^2}\right) \\ 1 - P_{FA} &= F_{\chi^2_{2N}}\left(\frac{2\lambda}{\sigma_w^2}\right) \\ F^{-1}_{\chi^2_{2N}}\left(1 - P_{FA}\right)\sigma_w^2 &= 2\lambda, \end{aligned} \tag{3.12}$$

which will then be plugged into eq. (3.10) to attain

$$\begin{aligned} P_D &= 1 - F_{\chi^2_{2N}}\left(\frac{F^{-1}_{\chi^2_{2N}}\left(1-P_{FA}\right)\sigma_w^2}{\sigma_w^2+\sigma_s^2}\right) \\ &= 1 - F_{\chi^2_{2N}}\left(\frac{F^{-1}_{\chi^2_{2N}}\left(1-P_{FA}\right)}{1+\frac{\sigma_s^2}{\sigma_w^2}}\right) \\ &= 1 - F_{\chi^2_{2N}}\left(\frac{F^{-1}_{\chi^2_{2N}}\left(1-P_{FA}\right)}{1+\mathrm{SNR}}\right) \end{aligned} \tag{3.13}$$

yielding a function of form $P_D(P_{FA})$, the ROC [69].

CLT Statistics

Although the test statistic $\mathrm{T}(\mathbf{x})$ has a chi-square distribution, the CLT suggests that the sum of IID random variables with finite mean and variance approaches a Gaussian distribution for large N [44, 60]. Modeling the test statistic as a Gaussian distribution makes the calculation of parameters more efficient (CT-06) while also allowing to analytically derive the number of samples N needed to achieve a certain P_{FA} and P_D, which is not possible using the chi-square distribution directly [69].

Because the test-statistic $\mathrm{T}(\mathbf{x})$ (see eq. (3.6)) of the ED has the following structure

$$\mathrm{T}(\mathbf{x}) = \sum_{n=1}^{N} |x_n|^2 = \sum_{n=1}^{N} x_{r,n}^2 + x_{i,n}^2 = \sum_{n=1}^{N} z_n \tag{3.14}$$

and z_n is an IID random variable with finite mean $\mu = \mathrm{E}[z_n] < \infty$ and finite variance $\sigma^2 = \mathrm{Var}[z_n] < \infty$, the CLT states that test statistic can be approximated by

$$\mathrm{T}(\mathbf{x}) = \sum_{n=1}^{N} z_n \approx \widehat{\mathrm{T}}(\mathbf{x}) = \mathcal{N}\left(\sum_{n=1}^{N}\mu_n, \sum_{n=1}^{N}\sigma_n^2\right) = \mathcal{N}\left(N\,\mathrm{E}[z], N\,\mathrm{Var}[z]\right). \tag{3.15}$$

One consequently needs to find $\mathrm{E}[z]$ and $\mathrm{Var}[z]$. Starting with the mean

$$\begin{aligned} \mathrm{E}[z] &= \mathrm{E}[x_r^2 + x_i^2] \\ &= \mathrm{E}[x_r^2] + \mathrm{E}[x_i^2] \\ &= \mathrm{Var}[x_r] - \mathrm{E}[x_r]^2 + \mathrm{Var}[x_i] - \mathrm{E}[x_i]^2 \\ &= \frac{1}{2}\sigma^2 - 0 + \frac{1}{2}\sigma^2 - 0 \\ &= \sigma^2, \end{aligned} \tag{3.16}$$

where $\mathrm{Var}[x] = \mathrm{E}[x^2] + \mathrm{E}[x]^2$ for any real random variable [59] and using the definition of the complex Gaussian from appendix A. Continuing with the variance

$$\begin{aligned}
\mathrm{Var}[z] &= \mathrm{Var}[x_r^2 + x_i^2] \\
&= \mathrm{Var}[x_r^2] + \mathrm{Var}[x_i^2] + \mathrm{Cov}[x_r^2 + x_i^2] \\
&= \mathrm{Var}[x_r^2] + \mathrm{Var}[x_i^2] + 0 \\
&= \mathrm{E}[(x_r^2)^2] - \mathrm{E}[x_r^2]^2 + \mathrm{E}[(x_i^2)^2] - \mathrm{E}[x_i^2]^2 \\
&= \mathrm{E}[x_r^4] - \left(\frac{1}{2}\sigma^2\right)^2 + \mathrm{E}[x_i^4] - \left(\frac{1}{2}\sigma^2\right)^2 \\
&= 3\,\mathrm{Var}[x_r]^2 - \frac{1}{4}\sigma^4 + 3\,\mathrm{Var}[x_i]^2 - \frac{1}{4}\sigma^4 \\
&= 3\left(\frac{1}{2}\sigma^2\right)^2 + 3\left(\frac{1}{2}\sigma^2\right)^2 - \frac{1}{2}\sigma^4 \\
&= \frac{6}{4}\sigma^4 - \frac{1}{2}\sigma^4 \\
&= \sigma^4,
\end{aligned} \tag{3.17}$$

using the same identity as above, and the fact that for any $x \sim \mathcal{N}(0, \sigma^2)$ the fourth moment is $\mathrm{E}[x^4] = 3\,\mathrm{Var}[x]^2$ [59]. The mean and the variance are thus

$$\mathrm{E}[z] = \begin{cases} \sigma_w^2 & \text{for } \mathcal{H}_0 \\ \sigma_w^2 + \sigma_s^2 & \text{for } \mathcal{H}_1 \end{cases} \tag{3.18}$$

$$\mathrm{Var}[z] = \begin{cases} \sigma_w^4 & \text{for } \mathcal{H}_0 \\ (\sigma_w^2 + \sigma_s^2)^2 & \text{for } \mathcal{H}_1 \end{cases} \tag{3.19}$$

and therefore the approximated test statistic, using the CLT, is distributed as

$$\widehat{\mathrm{T}}(\mathbf{x}) \sim \begin{cases} \mathcal{N}\left(N\sigma_w^2, N\sigma_w^4\right) & \text{for } \mathcal{H}_0 \\ \mathcal{N}\left(N(\sigma_w^2 + \sigma_s^2), N(\sigma_w^2 + \sigma_s^2)^2\right) & \text{for } \mathcal{H}_1. \end{cases} \tag{3.20}$$

Using this, one can derive the probability of false alarm and the probability of detection (similar to [70])

$$P_{FA} = p\left(\widehat{\mathrm{T}}(\mathbf{x}) > \lambda \mid \mathcal{H}_0\right) = \mathrm{Q}\left(\frac{\lambda - N\sigma_w^2}{\sqrt{N}\sigma_w^2}\right) \tag{3.21}$$

$$P_D = p\left(\widehat{\mathrm{T}}(\mathbf{x}) > \lambda \mid \mathcal{H}_1\right) = \mathrm{Q}\left(\frac{\lambda - N(\sigma_w^2 + \sigma_s^2)}{\sqrt{N}(\sigma_w^2 + \sigma_s^2)}\right), \tag{3.22}$$

where $\mathrm{Q}\left(\frac{\lambda-\mu}{\sigma}\right)$ is the normalized Q-function (see appendix A.1.1). Rearranging eq. (3.21) yields the threshold

$$\lambda = \sqrt{N}\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA}) + N\sigma_w^2 = (\sqrt{N}\,\mathrm{Q}^{-1}(P_{FA}) + N)\sigma_w^2 \tag{3.23}$$

and combining eq. (3.22) with eq. (3.23) results in the ROC [69]

$$\begin{aligned}
P_D &= \mathrm{Q}\left(\frac{\sqrt{N}\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA}) + N\sigma_w^2 - N(\sigma_w^2 + \sigma_s^2)}{\sqrt{N}(\sigma_w^2 + \sigma_s^2)}\right) \\
&= \mathrm{Q}\left(\frac{\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA}) - \sqrt{N}\sigma_s^2}{\sigma_w^2 + \sigma_s^2}\right).
\end{aligned} \tag{3.24}$$

In contrast to the chi-square statistics, it is now possible to derive an analytical term for the number of samples N required to achieve a certain P_{FA} and P_D pair [69]. Rearranging the P_{FA} and P_D equations so that the threshold is isolated, setting them equal, and simplifying yields the desired term

$$\begin{aligned}
\overbrace{\sqrt{N}(\sigma_w^2+\sigma_s^2)\,\mathrm{Q}^{-1}(P_D)+N\sigma_w^2+N\sigma_s^2}^{\text{from } eq.\ (3.22)} &= \overbrace{\sqrt{N}\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA})+N\sigma_w^2}^{\text{from } eq.\ (3.21)} \\
\sqrt{N}(\sigma_w^2+\sigma_s^2)\,\mathrm{Q}^{-1}(P_D)+N\sigma_s^2 &= \sqrt{N}\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA}) \\
N\sigma_s^2 &= \sqrt{N}(\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA})-(\sigma_w^2+\sigma_s^2)\,\mathrm{Q}^{-1}(P_D)) \\
\sqrt{N} &= \frac{\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA})-(\sigma_w^2+\sigma_s^2)\,\mathrm{Q}^{-1}(P_D)}{\sigma_s^2} \\
N &= \frac{(\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA})-(\sigma_w^2+\sigma_s^2)\,\mathrm{Q}^{-1}(P_D))^2}{\sigma_s^4}. \qquad (3.25)
\end{aligned}$$

Verification and numerical comparison between the chi-square and CLT statistics will follow in section 5.1.

Complexity

The ED requires N multiplications, followed by $N-1$ additions, resulting in an overall algorithmic complexity of $\mathcal{O}(N)$. This makes the ED easy to compute (CT-05) and not very resource-intensive (CT-06) when compared to other detectors.

Limitations

While EDs are simple, have well-understood performance, and require minimal computational resources, one major disadvantage arises: they require the precise knowledge of the background noise power. If noise power levels are unknown or imprecise, the resulting performance will rapidly degrade, coming nowhere near the above analytically derived metrics (see section 5.1.2). It is thus of paramount importance to precisely measure noise levels dynamically. To overcome this problem, noise estimation methods will be presented in section 3.3 and combined with ED.

3.1.2. Eigenvalue Detection

The *eigenvalue detector* exploits the fact that signals are (in general) correlated in order to discriminate them from noise [26, 61]. This is done by utilizing functions of the eigenvalues of the sample covariance matrix as test-statistics, leading to the design of a detector that works, similar to the ED, without any prior knowledge about the signal that is to be detected (MR-01).

Detector

The received discrete-time sample vector $\mathbf{x}$ with length N, under both hypotheses, is again given as

$$\mathcal{H}_0: \quad \mathbf{x} = \mathbf{w} \tag{3.26}$$
$$\mathcal{H}_1: \quad \mathbf{x} = \mathbf{s} + \mathbf{w}, \tag{3.27}$$

where $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_w^2\mathbf{I}\right)$ is the CWGN and $\mathbf{s} \sim \mathcal{CN}\left(\mathbf{0}, \mathbf{C}_s\right)$ is the signal vector with unknown covariance matrix $\mathbf{C}_s$. Constructing the following processing vectors [61]

$$\widetilde{\mathbf{w}}_n = [w_n, w_{n-1}, \ldots, w_{n-L+1}]^{\mathrm{T}} \tag{3.28}$$
$$\widetilde{\mathbf{s}}_n = [s_n, s_{n-1}, \ldots, s_{n-L+1}]^{\mathrm{T}} \tag{3.29}$$
$$\widetilde{\mathbf{x}}_n = [x_n, x_{n-1}, \ldots, x_{n-L+1}]^{\mathrm{T}}, \tag{3.30}$$

where L is called the smoothing factor (the processing vectors can be thought of as sliding windows with location n and length L), one has

$$\mathcal{H}_0: \quad \widetilde{\mathbf{x}}_n = \widetilde{\mathbf{w}}_n \tag{3.31}$$
$$\mathcal{H}_1: \quad \widetilde{\mathbf{x}}_n = \widetilde{\mathbf{s}}_n + \widetilde{\mathbf{w}}_n, \tag{3.32}$$

which is a direct consequence of the original signal model. The statistical covariance matrix, with respect to the processing vector $\widetilde{\mathbf{x}}_n$, can be found as [72]

$$\begin{aligned}\mathbf{R}_{\widetilde{\mathbf{x}}} &= [\mathrm{Cov}[\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{x}}_j]]_{1\leq i,j\leq L} \\ &= \mathrm{E}[(\widetilde{\mathbf{x}}_n - \mathrm{E}[\widetilde{\mathbf{x}}_n])(\widetilde{\mathbf{x}}_n^{\mathrm{H}} - \mathrm{E}[\widetilde{\mathbf{x}}_n^{\mathrm{H}}])] \\ &\overset{\mathrm{E}[\widetilde{\mathbf{x}}_n]=\mathbf{0}}{=} \mathrm{E}[\widetilde{\mathbf{x}}_n\widetilde{\mathbf{x}}_n^{\mathrm{H}}]\end{aligned} \tag{3.33}$$

and for $\widetilde{\mathbf{w}}_n$ and $\widetilde{\mathbf{s}}_n$ respectively

$$\mathbf{R}_{\widetilde{\mathbf{w}}} = \mathrm{E}[\widetilde{\mathbf{w}}_n\widetilde{\mathbf{w}}_n^{\mathrm{H}}] \tag{3.34}$$
$$\mathbf{R}_{\widetilde{\mathbf{s}}} = \mathrm{E}[\widetilde{\mathbf{s}}_n\widetilde{\mathbf{s}}_n^{\mathrm{H}}]. \tag{3.35}$$

The covariance matrices of the processing vectors express the auto-correlations of the samples over time; with time lag 0 to time lag $L-1$. Since CWGN is not correlated, the covariance matrix of the noise processing vector $\mathbf{R}_{\widetilde{\mathbf{w}}}$ has no correlation for any time lag, except 0. This fact collapses the noise covariance matrix into a simple diagonal matrix $\mathbf{R}_{\widetilde{\mathbf{w}}} = \sigma_w^2\mathbf{I}$. Because signals are generally correlated, the signal covariance matrix $\mathbf{R}_{\widetilde{\mathbf{s}}}$ has off-diagonal entries $\neq 0$ and can not be collapsed into a simple diagonal matrix. Thus, under the two hypotheses the statistical covariance matrices with respect to the processing vector $\widetilde{\mathbf{x}}_n$ are [61]

$$\mathcal{H}_0: \quad \mathbf{R}_{\widetilde{\mathbf{x}}} = \sigma_w^2\mathbf{I} \tag{3.36}$$
$$\mathcal{H}_1: \quad \mathbf{R}_{\widetilde{\mathbf{x}}} = \mathbf{R}_{\widetilde{\mathbf{s}}} + \sigma_w^2\mathbf{I}. \tag{3.37}$$

Under $\mathcal{H}_0$, the off-diagonal elements in $\mathbf{R}_{\widetilde{\mathbf{x}}}$ are all 0, while under $\mathcal{H}_1$ they are $\neq 0$ (because the signal samples are assumed to be correlated). It is important to note, that the exact structure of $\mathbf{R}_{\widetilde{\mathbf{x}}}$ under $\mathcal{H}_1$ is not known. This however, is not a problem, as one can still differentiate between the two hypotheses by taking all off-diagonal elements into account simultaneously.

From a practical perspective, the statistical covariance matrix must be estimated using a finite number of samples N. The so called *sample covariance matrix* can be found as follows [72]

$$\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}} = \mathrm{E}[\widetilde{\mathbf{x}}_n \widetilde{\mathbf{x}}_n^{\mathrm{H}}] = \frac{1}{N-L+1} \sum_{n=L}^{N} \widetilde{\mathbf{x}}_n \widetilde{\mathbf{x}}_n^{\mathrm{H}} \tag{3.38}$$

and converges to the actual covariance matrix of the underlying process for $N \to \infty$.

A slightly different but fundamentally identical viewpoint can be achieved by considering the sample auto-correlations of the received signal $\mathbf{x}$, defined as [73]

$$\mathrm{R}(l) = \mathrm{E}[x_n x_{n-l}^*] = \frac{1}{N} \sum_{n=1}^{N} x_n x_{n-l}^* \quad l = 0, 1, \ldots, L-1, \tag{3.39}$$

where N is again the number of samples available and L is the smoothing factor. The statistical covariance matrix can then be approximated by

$$\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}} = \begin{bmatrix} \mathrm{R}(0) & \mathrm{R}(1) & \cdots & \mathrm{R}(L-2) & \mathrm{R}(L-1) \\ \mathrm{R}(1) & \mathrm{R}(0) & \cdots & \mathrm{R}(L-3) & \mathrm{R}(L-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathrm{R}(L-2) & \mathrm{R}(L-3) & \cdots & \mathrm{R}(0) & \mathrm{R}(1) \\ \mathrm{R}(L-1) & \mathrm{R}(L-2) & \cdots & \mathrm{R}(1) & \mathrm{R}(0) \end{bmatrix}. \tag{3.40}$$

Here, one can clearly identify the symmetric Toeplitz structure of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ [61]. Under $\mathcal{H}_0$ all auto-correlations with $l \neq 0$ will be $\mathrm{R}(l) = 0$, since the noise $\mathbf{w}$ is not correlated. Under $\mathcal{H}_1$ some (likely all) auto-correlations will be $\mathrm{R}(l) \neq 0$ due to the signal $\mathbf{s}$ being correlated.

One can directly use the properties of the diagonal and off-diagonal elements and construct the following test $\mathrm{T}(\mathbf{x})$, which is called *covariance absolute value* (CAV) detector [73]. Here, $r_{n,m}$ denotes the n-th row and m-th column entry of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$

$$\gamma_1 = \frac{1}{L} \sum_{n=1}^{L} \sum_{m=1}^{L} |r_{n,m}| \tag{3.41}$$

$$\gamma_2 = \frac{1}{L} \sum_{n=1}^{L} |r_{n,n}| \tag{3.42}$$

$$\mathrm{T}(\mathbf{x}) = \frac{\gamma_1}{\gamma_2} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_{\mathrm{CAV}}. \tag{3.43}$$

Under $\mathcal{H}_0$ one expects that $\gamma_1 = \gamma_2$, since all off-diagonal elements are 0. Under $\mathcal{H}_1$ one expects that $\gamma_1 > \gamma_2$; thus making the test-statistic $\mathrm{T}(\mathbf{x}) > 1$. While this test does work, multiple authors have shown that there exists a more powerful test [52, 61, 63, 74].

Instead of using the entries of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ directly, one can examine its eigenvalues instead. Denoting them in descending order from largest to smallest, one has $\nu_1 \geq \nu_2 \geq \nu_3 \geq \ldots \geq \nu_L$. Since the covariance matrix under $\mathcal{H}_0$ is $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}} = \sigma_w^2 \mathbf{I}$, all L eigenvalues will be $\nu_1 = \nu_2 = \nu_3 = \ldots = \nu_L = \sigma_w^2$. The eigenvalues of $\hat{\mathbf{R}}_{\widetilde{\mathbf{s}}}$ from largest to smallest are defined as $\rho_1 \geq \rho_2 \geq \rho_3 \geq \ldots \geq \rho_L$. Under $\mathcal{H}_1$ one thus has $\nu_j = \rho_j + \sigma_w^2$. Because $\rho_1 > \rho_L$, it is possible to detect signals by computing the ratio of the largest to the smallest eigenvalue [61]

$$\mathrm{T}(\mathbf{x}) = \frac{\nu_{\max}}{\nu_{\min}} = \frac{\nu_1}{\nu_L} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_{\mathrm{MME}}. \tag{3.44}$$

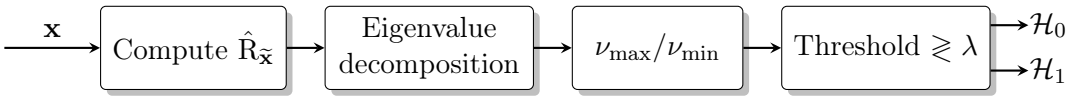

Figure 3.2.: Simplified MME block diagram.

The resulting detector is called the *maximum-minimum eigenvalue* (MME) detector. It is important to note that the ratio in eq. (3.44) eliminates the influence of the noise eigenvalues and is therefore independent of noise power. This is the major advantage of eigenvalue detection: it allows for blind spectrum sensing (MR-01) without the knowledge of background noise power (CT-02).

A related eigenvalue detector is the so-called *energy with minimum eigenvalue* (EME) detector. As the name suggests, it works by comparing the the normalized signal energy (signal power) with the minimum eigenvalue

$$\mathrm{T}(\mathbf{x}) = \frac{\xi(\mathbf{x})}{\nu_{\min}} = \frac{\xi(\mathbf{x})}{\nu_L} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_{\mathrm{EME}} \tag{3.45}$$

$$\text{with} \quad \xi(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} |x_n|^2. \tag{3.46}$$

According to [52], the EME performs very similarly to the MME detector. However, because the MME detector is more prevalent in research literature and because the statistics are simpler, only the MME detector will be considered from here on (consult [73] and [61] for more information on the CAV and EME detectors).

Approximate Statistics

To derive the statistics for the MME detector, one needs to examine the eigenvalue distribution of the sample covariance matrix $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$. Because of the limited number of available samples N, $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ deviates from its underlying statistical distribution (the statistical covariance matrix $\mathbf{R}_{\widetilde{\mathbf{x}}}$ is unknown). This makes it difficult to analytically determine the threshold and its related statistics. The mathematical process is very involved, relies on random matrix theory, and goes beyond this thesis's scope. Thus, only the results from [73, 74] will be presented here. However, the general procedure is similar to the one presented in section 3.1.1 (see [75, 76] for more information on the theory behind the statistical derivation).

One can formulate the probability of false alarm P_{FA} and probability of detection P_D for the MME detector with the help of

$$c_1 = (\sqrt{N-1} + \sqrt{ML})^2 \tag{3.47}$$

$$c_2 = (\sqrt{N_1} - \sqrt{ML})(\frac{1}{\sqrt{N-1}} + \frac{1}{\sqrt{ML}})^{1/3} \tag{3.48}$$

as the following approximate functions

$$P_{FA} \approx 1 - F_1\left(\frac{\lambda_{\text{MME}}(\sqrt{N}-\sqrt{ML})^2 - c_1}{c_2}\right) \tag{3.49}$$

$$P_D \approx 1 - F_1\left(\frac{\lambda_{\text{MME}}N + N(\lambda_{\text{MME}}\rho_L - \rho_1)/\sigma_w^2 - c_1}{c_2}\right), \tag{3.50}$$

where $F_1(\cdot)$ denotes the Tracy–Widom CDF of order 1 (see appendix A.1.3) and where M is the oversampling factor. The threshold λ_{MME} can be approximated by

$$\lambda_{\text{MME}} \approx \frac{(\sqrt{N}+\sqrt{ML})^2}{(\sqrt{N}-\sqrt{ML})^2}\left(1 + \frac{(\sqrt{N}+\sqrt{ML})^{-2/3}}{(NML)^{1/6}}F_1^{-1}(1-P_{FA})\right), \tag{3.51}$$

where $F_1^{-1}(\cdot)$ denotes the inverse Tracy–Widom CDF of order 1. In contrast to the threshold of the ED, the threshold of the MME detector does not depend on the noise power σ_w^2; making it very robust against changing and uncertain noise power levels (CT-02).

Complexity

The MME detector consists mainly of two parts: computing the sample covariance matrix $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}}$ and finding its eigenvalue decomposition. Noting that the covariance matrix is symmetric and Toeplitz, one only needs to compute the "top row"; that is all $\mathrm{R}(l),\ l = 0, \ldots, L-1$ (see eq. (3.40)). This equates to LN multiplications and $L(N-1)$ additions, resulting in an $\mathcal{O}(LN)$ algorithmic complexity. Eigenvalue decomposition generally has a $\mathcal{O}(L^3)$ complexity (where L is the matrix size). This results in an overall complexity of $\mathcal{O}(LN + L^3)$ [61].

Limitations

When compared to other spectrum sensing methods, the eigenvalue detector is relatively resource-intensive. Its algorithmic complexity, with increasing covariance matrix size L, grows cubically. This requires careful implementation, likely with the support of hardware acceleration (see section 6.2), especially when real-time requirements are to be met (CT-04). Additionally, the eigenvalue detector requires relatively many samples N in order for the sample covariance matrix to be sufficiently accurate. This further increases complexity and leads to high sensing latencies (CT-05). More accurate statistics, needing fewer samples, are an ongoing research topic [63, 77, 78].

The eigenvalue detector assumes that the signal samples are correlated. If this is not the case, the detector will not be able to detect the signal. However, signal samples can usually assumed to be correlated due to the following reasons [61]:

1. The signal is oversampled at the receiver. When the sampling frequency f_s is larger than the bandwidth B by some factor M, then M consecutive signal samples will be correlated. There is usually some form of oversampling present, especially in wideband sensing applications (see section 3.2).

Narrowband method	Blind	Complexity	Robustness
Matched filter*	✗	medium	high
Energy	✓	low	low
Covariance*	✗	medium	medium
Eigenvalue	✓	high	high
Cyclostationary*	(✓)	very high	high

Table 3.1.: Comparison of narrow band detection methods (*presented in appendix B.1).

2. The propagation channel $h(\tau, t)$ (see section 2.2) has time dispersion, meaning that the signal symbols are spread out in time. This results in correlation between adjacent signal samples as they travel through the channel. Time dispersion in the channel can, for example, be caused by multipath propagation, where the multipath delay exceeds the symbol period.

3. The signal symbols are correlated themselves. Due to redundant patterns in signal protocols and data, signals are almost always intrinsically correlated.

3.1.3. Conclusion

This section, and the accompanying appendix B, presented the most prevalent narrowband spectrum sensing methods in research literature [29–31]. The matched filter (appendix B.1.1) and covariance detector (appendix B.1.2) are not applicable to spectrum sensing on SALSAT, as they require a priori knowledge of the signal trying to detect (MR-01). While cyclostationary detection (appendix B.1.3) can be adapted to blind spectrum sensing, the resulting detector has very high complexity, severely degrading run time performance. Therefore, cyclostationary detection is also not applicable to SALSAT (CT-06).

Similarly to covariance and cyclostationary detection, eigenvalue detection uses properties of signals to differentiate them from noise. However, it does this by not assuming any specific signal type or covariance structure while at the same time offering reasonable complexity. Nonetheless, its run time performance is still not very good, making its implementation in low-power systems, with little computational resources, challenging (CT-06). Energy detection is the simplest of all presented algorithms; it has the lowest complexity and the best run time behavior. Its performance metrics are very well understood, enabling precise tuning of P_{FA} and P_D values. However, energy detection requires exact knowledge of background noise, making it not applicable to SALSAT by itself (CT-02). Nevertheless, with the help of noise estimation, energy detection becomes suitable for real-world applications (see section 3.3.3).

3.2. Wideband Spectrum Sensing

Thus far, the spectrum sensing methods presented are limited to the detection of signals in a single frequency band; they only provide a scalar occupancy value $\mathcal{H}_0/\mathcal{H}_1$ (see section 2.3). Wideband spectrum sensing, in contrast to narrowband spectrum sensing, involves observing multiple subbands simultaneously and identifying subbands that are occupied by a signal and those which are free (see

fig. 1.4 and MR-02). The overall spectrum utilization can be significantly improved by dividing the wideband into multiple parallel narrowbands and jointly sensing for spectrum opportunities on narrow subbands. This section provides a general overview of the most common wideband spectrum sensing methods [30, 79–81]. Methods relevant to this thesis will be presented in greater depth, while those not directly relevant will only briefly be considered in appendix B.2.

3.2.1. Wideband Signal Model

Under the assumption that the individual signal transmissions, spread over the spectrum band, are uncorrelated, the detection model generalizes to

$$\mathcal{H}_{0,l}: \quad \mathbf{x}_l = \mathbf{w}_l \tag{3.52}$$

$$\mathcal{H}_{1,l}: \quad \mathbf{x}_l = \mathbf{s}_l + \mathbf{w}_l, \tag{3.53}$$

where $l = 1, 2, \ldots, L$ denotes the l-th subband $\mathcal{B}_l$ of bandwidth B_l; the total observed band is $\mathcal{B} = \bigcup_{i=1}^{L} \mathcal{B}_l$ with bandwidth $B = \sum_{i=1}^{L} B_l$ [28, 81]. The detection problem then becomes one of extracting the L subband signals $\mathbf{x}_l$, each of bandwidth B_l, from the original signal $\mathbf{x}$ and subsequently applying the test statistic

$$\mathrm{T}(\mathbf{x}_l) \underset{\mathcal{H}_{0,l}}{\overset{\mathcal{H}_{1,l}}{\gtrless}} \lambda_l \tag{3.54}$$

to each of them, yielding a decision vector $\mathcal{H} = [\mathcal{H}_{0,1}/\mathcal{H}_{1,1}, \mathcal{H}_{0,2}/\mathcal{H}_{1,2}, \ldots, \mathcal{H}_{0,L}/\mathcal{H}_{1,L}]^{\mathrm{T}}$ where each entry either indicates a free or occupied spectrum subband. This requires L thresholds $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_L]^{\mathrm{T}}$ which directly determine the $P_{FA} = [P_{FA,1}, P_{FA,2}, \ldots, P_{FA,L}]^{\mathrm{T}}$ and $P_D = [P_{D,1}, P_{D,2}, \ldots, P_{D,L}]^{\mathrm{T}}$ vectors, which contain the desired probability of false alarm and probability of detection for every subband [79]. There exists a great deal of literature about jointly tuning these parameters to achieve optimal throughput, especially when one assumes correlation between individual subband signals [82–84]. The treatment of such topics would go beyond this thesis's scope and is therefore not further considered here.

3.2.2. Wideband Energy Detection

As the name suggests, the *wideband energy detector* (WED) expands the concept from narrowband energy detection to wideband signals. The incoming signal $\mathbf{x}$ is first split into L subband signals $\hat{\mathbf{x}}_l$ with the help of *spectral* estimation, where $\hat{\mathbf{x}}$ denotes the frequency domain representation of $\mathbf{x}$ (see appendix A.3). Then L narrowband EDs are applied in parallel to each $\hat{\mathbf{x}}_l$ [28, 79, 81].

Spectral Estimation

Splitting the wideband signal $\mathbf{x}$ into its subband signals $\hat{\mathbf{x}}_l$ requires *spectral* estimation, or more concretely, *power spectral density* (PSD) estimation (see appendix A.3.2). The PSD characterizes the signal's power content at each frequency, thereby allowing the analysis of spectral subbands and their energies [64].

There are a multitude of spectral estimation techniques available. They can mostly be subdivided into *parametric* and *non-parametric* methods [64, 85]. The parametric approaches assume some underlying statistical model in order to achieve better performance on fewer data samples. Because SALSAT requires blind spectrum sensing (MR-01), generally[1] no such model assumptions can be made, thus ruling out parametric spectral estimation techniques.

Non-parametric methods generally rely on the *Fourier transform*, which transforms signals from the time domain into the frequency domain, thereby exposing the signal's frequency components [64, 86]. The concept of Fourier transforms in the realm of spectral estimation can further be advanced by employing modern techniques, such as *multitaper power spectral density* (MT-PSD) estimation [87–89]. The MT-PSD estimator achieves superior estimation results by using carefully designed tapering functions to improve the estimated spectrum variance, without compromising on the level of incurred spectral leakage [79, 90]. Although the MT-PSD estimator has superior performance, and is robust against noise inaccuracies, it has notably higher computational and implementational complexities, when compared to simple Fourier transforms (CT-06) [79]. Additionally, many communication systems have built-in Fourier transform processors (more precisely FFT cores [91, 92]) that can be utilized to estimate the spectrum over wide frequency ranges, making the Fourier transform a prime candidate for spectral estimation. Thus, this thesis will only concern itself with Fourier transform-based spectral estimation.

Because the data vector $\mathbf{x}$ consists of discrete samples, one needs to employ the *discrete Fourier transform* (DFT). Implementations of the DFT rely on the *fast Fourier transform* (FFT) algorithm, which is a lower complexity implementation of the DFT, making it practically feasible. The FFT gets computed on short segments of the incoming discrete data samples and reduces the $\mathcal{O}(N^2)$ complexity of the DFT to $\mathcal{O}(N \log N)$ [64, 86]. Common FFT sizes are powers of two 2^n with $n = 8, \ldots, 14$, resulting in FFT lengths of 256 to 8192. To reduce spectral leakage coming from sharp edges at the beginning and the end of the data segments, one can employ *windowing*, which smoothes the segment edges. Prominent window types are the *Hann* and *Flattop* window (see fig. A.9). Employing no special window results in a *rectangular* window. This segmentation and windowing procedure is called *short-time Fourier transform* (STFT), as it computes a short windowed Fourier Transform [64, 86]. Appendix A.3 provides a more formal overview of the Fourier transform, the DFT, FFT, STFT, and their mathematical relations. There, one can also find the definition of *Parsevals theorem* [64], which is the pillar on which wideband energy detection is built. It states that the energy of signals is preserved through the Fourier transform, thus making wideband energy detection feasible.

Detection

After the time domain signal $\mathbf{x} = [x_1, \ldots, x_N]^{\mathrm{T}}$ has been converted into its frequency domain counterpart $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_K]^{\mathrm{T}}$ by the STFT, where k is now a discrete variable of frequency, the wideband test-statistic

$$\mathrm{T}(\hat{\mathbf{x}}_l) = \sum_{k \in \mathcal{B}_l} |\hat{x}_k|^2 \underset{\mathcal{H}_{0,l}}{\overset{\mathcal{H}_{1,l}}{\gtrless}} \lambda_l \tag{3.55}$$

[1] One could assume very general models, thereby making them applicable to a wide range of signal types. This, however, is not further studied here as the performance gains are, for the most part, not justified in comparison to non-parametric methods [85].

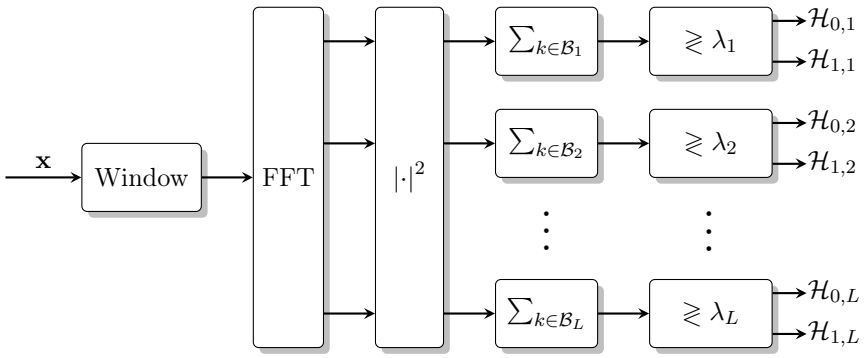

Figure 3.3.: Wideband energy detector block diagram (derived from [70]).

is applied; resulting in the decision vector $\mathcal{H} = [\mathcal{H}_{0,1}/\mathcal{H}_{1,1}, \mathcal{H}_{0,2}/\mathcal{H}_{1,2}, \ldots, \mathcal{H}_{0,L}/\mathcal{H}_{1,L}]^{\mathrm{T}}$ of length L, where each entry indicates a free or occupied spectrum subband [79, 81], and where $|\hat{x}_k|^2$ is the PSD of x_n. Summing over a subspace of the PSD, denoted here in terms of the subband $\mathcal{B}_l$, results in the energy contained in that PSD subspace (or subband). By adjusting the number of STFT samples going into a specific subband $\mathcal{B}_l$, one can dynamically adapt the resolution of the WED and, for example, increase the resolution in some subband while reducing the resolution in other frequency regions.

Statistics

Since wideband energy detection operates analogous to narrowband energy detection, the statistics derived in section 3.1.1 can generally be adapted. However, when applying these statistics to a single subband $\mathcal{B}_l$, one needs to take the reduced noise bandwidth into account. This is because the CWGN power σ_w^2 is spread uniformly across the total original bandwidth B [93]. The noise power in $\mathcal{B}_l$ is thus only

$$\sigma_{l,w}^2 = \frac{B_l}{B}\sigma_w^2. \tag{3.56}$$

When a signal is spread over multiple bands, the same concept applies to its power σ_s^2. However, since the signal power is usually not uniformly spread across its bandwidth, modeling this phenomenon becomes more difficult. Therefore, the signal power $\sigma_{l,s}^2$ will explicitly be given in terms of the power in a specific subband $\mathcal{B}_l$. Thus, the reduction in noise power, compared to the signal power, results in an effective boost of the SNR.

With these modifications in mind, one can derive the SNR in a given subband $\mathcal{B}_l$ as

$$\mathrm{SNR}_l = \frac{\sigma_{l,s}^2}{\sigma_{l,w}^2} = \frac{B}{B_l}\frac{\sigma_{l,s}^2}{\sigma_w^2} \tag{3.57}$$

and when all bands have the same bandwidth $B_1 = B_2 = \ldots = B_L$, then

$$\mathrm{SNR}_l = L\frac{\sigma_{l,s}^2}{\sigma_w^2}. \tag{3.58}$$

Another important factor that needs to be considered is the reduced number of samples N that underlie the computed energy in a subband $\mathcal{B}_l$. The number of samples are, similar to the contained power, proportional to the relative bandwidth of $\mathcal{B}_l$ [94]. When N is the total number of samples

used to compute the STFT, then the samples N_l underlying the energy estimate of band $\mathcal{B}_l$ are

$$N_l = \frac{B_l}{B} N \tag{3.59}$$

and for equal bandwidths

$$N_l = \frac{N}{L}. \tag{3.60}$$

This reduction in the number of samples results in reduced accuracy, or increased variance, of the spectrum band energy estimate. It also encompasses the absolute noise power reduction. Thus, one only needs to adjust the number of samples N_l and the SNR_l in a subband, but not the noise power $\sigma_{l,w}^2$. The SNR_l adaption will be done trough a relative increase of the signal power $\sigma_{l,s}^2$, resulting in

$$\widetilde{\sigma}_{l,s}^2 = \frac{B}{B_l} \sigma_{l,s}^2 \tag{3.61}$$

and for equal bandwidths

$$\widetilde{\sigma}_{l,s}^2 = L \sigma_{l,s}^2. \tag{3.62}$$

With the above presented relations, one can derive the wideband statistics for the chi-square and CLT models.

Chi-Square The probability of false alarm $P_{FA,l}$ and the probability of detection $P_{D,l}$ for a subband $\mathcal{B}_l$ are

$$P_{FA,l} = 1 - F_{\chi^2_{2N_l}} \left(\frac{2\lambda_l}{\sigma_w^2} \right) \tag{3.63}$$

$$P_{D,l} = 1 - F_{\chi^2_{2N_l}} \left(\frac{2\lambda_l}{\sigma_w^2 + \widetilde{\sigma}_{l,s}^2} \right). \tag{3.64}$$

The threshold λ_l and the ROC equate to

$$\lambda_l = \frac{\sigma_w^2}{2} F^{-1}_{\chi^2_{2N_l}} \left(1 - P_{FA,l} \right) \tag{3.65}$$

$$\begin{aligned} P_{D,l} &= 1 - F_{\chi^2_{2N_l}} \left(\frac{F^{-1}_{\chi^2_{2N_l}} \left(1 - P_{FA,l} \right)}{1 + \frac{\widetilde{\sigma}_{l,s}^2}{\sigma_w^2}} \right) \\ &= 1 - F_{\chi^2_{2N_l}} \left(\frac{F^{-1}_{\chi^2_{2N_l}} \left(1 - P_{FA,l} \right)}{1 + \mathrm{SNR}_l} \right). \end{aligned} \tag{3.66}$$

CLT The CLT approximated probability of false alarm $P_{FA,l}$ and probability of detection $P_{D,l}$ for a subband $\mathcal{B}_l$ are given as

$$P_{FA,l} = \mathrm{Q} \left(\frac{\lambda_l - N_l \sigma_w^2}{\sqrt{N_l} \sigma_w^2} \right) \tag{3.67}$$

$$P_{D,l} = \mathrm{Q} \left(\frac{\lambda_l - N_l (\sigma_w^2 + \widetilde{\sigma}_{l,s}^2)}{\sqrt{N_l} (\sigma_w^2 + \widetilde{\sigma}_{l,s}^2)} \right). \tag{3.68}$$

The threshold λ_l, the ROC and the required number of samples N_l are

$$\lambda_l = (\sqrt{N_l}\,\mathrm{Q}^{-1}(P_{FA,l}) + N_l)\sigma_w^2 \tag{3.69}$$

$$P_{D,l} = \mathrm{Q}\left(\frac{\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA,l}) - \sqrt{N_l}\widetilde{\sigma}_{l,s}^2}{\sigma_w^2 + \widetilde{\sigma}_{l,s}^2}\right) \tag{3.70}$$

$$N_l = \frac{(\sigma_w^2\,\mathrm{Q}^{-1}(P_{FA,l}) - (\sigma_w^2 + \widetilde{\sigma}_{l,s}^2)\,\mathrm{Q}^{-1}(P_{D,l}))^2}{\widetilde{\sigma}_{l,s}^4}. \tag{3.71}$$

Complexity

The complexity of the WED can be derived from the FFT complexity, which is $\mathcal{O}(N \log N)$, and L narrowband EDs running in parallel, which have a total complexity of $\mathcal{O}(NL)$. Thus, the overall complexity equates to $\mathcal{O}(N \log N + NL)$.

Limitations

Similar to narrowband energy detection, wideband energy detection requires the precise knowledge of background noise power levels. Noise uncertainty severely degrades the detection performance of the WED. It is thus crucial to precisely measure noise levels dynamically. Noise estimation methods will be presented in section 3.3 and subsequently combined with wideband energy detection.

3.2.3. Wideband Eigenvalue Detection

Eigenvalue detection can (theoretically) be adapted to wideband operation by merely splitting the signal $\mathbf{x}$ into its subband components $\mathbf{x}_l$ and applying a narrowband eigenvalue detector on each $\mathbf{x}_l$; yielding L eigenvalue detection results. Since the eigenvalue detector operates on time-domain data, it is beneficial to apply a time-domain filtering method. One could employ several bandpass filters in parallel. Every bandpass filter h_l would be tuned to only let the frequency components of its corresponding subband $\mathcal{B}_l$ through. This filter configuration is known as a *filter bank* [85]. However, when employing such a system, two main problems emerge.

Whitening

The first problem one encounters when combining eigenvalue detection (or correlation-based detection in general) with some form of signal preprocessing is the induced correlation by the preprocessing systems into the data. Any filtering, both analog and digital, induces correlation [26, 61]. Because most RF systems employ some form of preprocessing or filtering, this is a general problem when working with correlation or eigenvalue-based detection. Thus, the in section 3.1.2 presented narrowband eigenvalue detector also suffers from this problem.

This so-called *noise coloring* [85] can be resolved by multiplying the covariance matrix $\mathbf{R}_{\widetilde{\mathbf{x}}}$ with a *whitening* matrix $\mathbf{Q}$, which reveres (or neutralizes) the correlation induced by preprocessing [26, 52, 61]. The whitened covariance matrix can be found as

$$\mathbf{R}'_{\widetilde{\mathbf{x}}} = \mathbf{Q}^{-1}\mathbf{R}_{\widetilde{\mathbf{x}}}\mathbf{Q}^{-1}, \tag{3.72}$$

so that under $\mathcal{H}_0$, $\mathbf{R}'_{\widetilde{\mathbf{x}}}$ again becomes

$$\mathbf{R}'_{\widetilde{\mathbf{x}}} = \mathbf{Q}^{-1}\mathbf{R}_{\widetilde{\mathbf{x}}}\mathbf{Q}^{-1} = \sigma_w^2\mathbf{I}. \tag{3.73}$$

Because the whitening matrix $\mathbf{Q}$ is only related to filter and preprocessing systems, it can be precomputed in advance by recording the (sample) covariance matrix $\mathbf{R}_{\widetilde{\mathbf{x}}}$ when measuring only (uncorrelated) CWGN

$$\mathbf{Q} = \mathbf{R}_{\widetilde{\mathbf{x}}}^{\frac{1}{2}}. \tag{3.74}$$

In order to reduce the variance of $\mathbf{Q}$, one could average multiple whitening matrices together (see section 5.4).

Limitations

The second problem arising when employing L bandpass filters and L eigenvalue detectors is the amount of resources required to implement such a system. Real-time, software-only implementations would not be practically feasible. Even hardware-based implementations would quickly hit their limit, especially in a resource-constrained environment like SALSAT (CT-06, CT-05). One solution would be to reduce the bandpass filter bank to a *single* tunable bandpass filter that can dynamically adjust its center frequency (and ideally also its bandwidth). Feeding the bandpass filtered signal into a *single* eigenvalue detector that detects the presents of a signal in only a *single* subband at a time would significantly reduce the implementation cost. Some search algorithm would then look for an unoccupied subband of desired width by adjusting the bandpass filter's center frequency until the eigenvalue detector detects no signal present. As soon as the current subband becomes occupied, the detector would start searching again. This would induce severe latencies, as there is no guarantee on how long the search takes until an unoccupied subband is identified. This approach would also only provide occupancy information for one subband at a time. It would not provide general spectrum occupancy information and is thus not well suited for research applications, such as SALSAT

3.2.4. Wavelet Detection

PSD estimation was introduced in section 3.2.2. The estimated PSD entails the power of the received data samples with respect to frequency. Under the assumption that the power of a subband $\mathcal{B}_l$ occupied by some signal is greater than the power of noise only subbands, the WED is able to decide on the presence or absence of a signal in each spectrum subband. By assuming that the power spectral characteristic is smooth within each subband but exhibits a discontinuous change between adjacent subbands $\mathcal{B}_l$ and $\mathcal{B}_{l+1}$, one can identify and locate spectrum holes by analyzing the irregularities in the estimated PSD with the *wavelet transform* (WT) (see appendix A.3.6) [95–99].

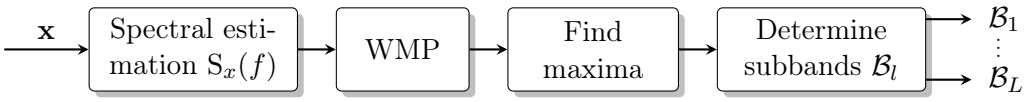

Figure 3.4.: Simplified wavelet detection block diagram.

A significant advantage of wavelet-based spectrum sensing comes from the fact that the aforementioned wideband spectrum sensing techniques have assumed that the boundaries f_l and widths B_l of subbands $\mathcal{B}_l$ are known a priori. However, this assumption is not practical, as signals in heterogeneous environments can have different bandwidths and occupy different parts of the predefined subbands [79, 81]. By defining the subbands based on information coming from the estimated PSD, wavelet-based spectrum sensing can overcome this problem and determine the boundaries of the subbands without prior knowledge of the locations and widths of subbands.

Detector

Wavelet-based spectrum sensing, similar to wideband energy detection, operates on the estimated PSD $S_x(f)$. Let $\psi_s(f)$ denote the discrete wavelet function

$$\psi_s(f) = \frac{1}{s}\psi\left(\frac{f}{s}\right) \tag{3.75}$$

dilated by the *dyadic* scaling factor s; which is defined using powers of two, i.e., $s = 2^m, m = 1, 2, \ldots, M$ [64, 86]. Commonly used wavelets include the Gaussian derivative wavelet family, as defined in appendix A.3.6. Of special interest to wavelet-based spectrum sensing is the first derivative Gaussian wavelet $\psi'_s(f)$. When applied to the PSD, the continuous wavelet transform (CWT)

$$\mathcal{W}'_s(f) = S_x(f) * \psi'_s(f) = \frac{1}{s}\int_{-\infty}^{\infty} x(\tau)\psi'(\frac{\tau - f}{s})d\tau \tag{3.76}$$

it is able to extract edges (also known as singularities) and detect gradient changes [97]. Edges of the subbands correspond to local maxima of the absolute value of the CWT when using the first derivative Gaussian wavelet. This method is known as the *wavelet transform modulus maxima* (WTMM) [97, 98]. One can thus extract subband boundaries by searching for local maxima

$$f_l = \arg\operatorname*{maxima}_f |\mathcal{W}'_s(f)|, \tag{3.77}$$

where a local maxima $\check{x}$ is generally defined as [100]

$$\operatorname*{maxima}_f \leftrightarrow \exists\delta > 0 : f(\check{x}) > f(x), x \in [\check{x} - \delta, \check{x} + \delta], \tag{3.78}$$

with more information about maxima finding (also known as peak detection) and the size of δ following in section 5.3. It is important to note that the CWT is carried out in the frequency domain on $S_x(f)$, while $S_x(f)$ relates to the received samples $\mathbf{x}$ via the STFT.

Due to the characteristics of the wideband spectrum and the PSD, strong noise power levels may negatively impact the detection of edges. To partially overcome this problem, a product of several first derivative Gaussian wavelet transforms is calculated, thus suppressing noise and sharpening

the edges. This technique is referred to as the *wavelet multiscale product* (WMP) [98] and expressed by

$$\mathcal{U}_M(f) = \prod_{M=1}^{M} \mathcal{W}'_{s=2^m}(f) \tag{3.79}$$

yielding the improved spectrum edge detection algorithm

$$f_l = \arg \underset{f}{\text{maxima}} \, |\mathcal{U}_M(f)|. \tag{3.80}$$

Increasing M results in more robust edge detection, with the incurred cost of being less sensitive. Typical values for M include $M = 3, \ldots, 6$.

The detected subband edges f_l can be used to reconstruct the subbands $\mathcal{B}_l$ with bandwidth B_l; the subbands not necessarily need to have the same bandwidth $B_1 \neq B_2 \neq \ldots \neq B_L$.

Complexity

A *filterbank* implementation of the WT has $\mathcal{O}(N)$ time complexity in certain cases (see appendix A.3.6). Many peak finding algorithms have $\mathcal{O}(N)$ complexity. With the complexity of the FFT being $\mathcal{O}(N \log N)$, the overall complexity of the wavelet-based edge detection equates to $\mathcal{O}(N(1 + M) + N \log N)$, making it relatively efficient. However, software implementations of the WT often have worse time complexity.

Limitations

While wavelet-based spectrum sensing provides a powerful and flexible tool for spectrum edge detection, it suffers from various problems. The first coming from the fact that sharp edges not merely arise at the boundaries of the subbands but also arise due to spurious signals or noise peaks. These undesired edges may degrade the boundary estimation [98, 99]. This phenomenon can be reduced by employing the WMP as mentioned earlier. Another deficiency of the wavelet-based edge detection algorithm comes from the fact that it requires relatively sharp edges at subband boundaries. When subbands transition smoothly between one another, wavelet-based edge detection has a hard time detecting edges. Lowering M increases the edge detection sensitivity while at the same time also increasing the probability for false edges to arise. As discussed below, one can further enhance wavelet-based spectrum sensing by introducing more rigorous edge criteria and by combining it with other spectrum sensing methods, such as wideband energy detection [99]. Energy detection helps to prevent false edges from occurring when employing sensitive wavelet-based edge detection.

3.2.5. Conclusion

This section outlined the most prominent wideband spectrum sensing methods and evaluated them for operation on SALSAT. While compressed wideband sensing offers attractive reductions in sampling rate demands, it is not yet mature enough for actual implementation (see appendix B.2).

Wideband method	Blind	Complexity	Robustness
Wideband energy	✓	medium	low
Wideband eigenvalue	✓	very high	high
Wavelet-based	(✓)	medium	low
Compressed*	(✓)	very high	low

Table 3.2.: Comparison of wideband detection methods (*presented in appendix B.2).

Analyzing the wideband spectrum with spectral estimation techniques enabled the adaption of narrowband energy detection to multiple bands and gave rise to STFT-based wideband energy detection. Combined with sensitive wavelet-based spectrum edge detection, wideband energy detection can operate in heterogeneous spectrum environments by dynamically adjusting the location and width of spectral subbands. Eigenvalue-based filter banks or tunable narrowband filters provide robust sensing performance by not needing information about background noise power levels. It was shown how adverse correlation, introduced by preprocessing and filtering, could be mitigated through whitening matrices. However, high complexity implementation and high latency operation make eigenvalue-based wideband sensing not applicable to constrained and real-time environments. Despite its shortcomings, the ability to operate independent of background noise levels, by extracting and separating signal from noise components through eigenvalues, will prove valuable, as presented in the following section.

3.3. Noise Estimation

In conjunction with wavelet-based spectrum edge detection, wideband energy detection proved to be the prime wideband spectrum sensing method in the previous section. However, uncertain knowledge of background noise power levels severely limits the performance of wideband energy detection. The exact knowledge of the background noise power σ_w^2 is of paramount importance to energy detection. Therefore, this section will concern itself with noise power estimation methods and their effects on wideband energy detection.

3.3.1. Free-Band Noise Estimation

One possible noise estimation algorithm consists of estimating the noise power from noise-only samples. Under the assumption that the noise samples $\mathbf{w}$ are IID and zero-mean Gaussian $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_w^2\mathbf{I}\right)$, the *maximum likelihood estimator* (MLE) of the background noise power is the ED

$$\hat{\sigma}_w^2 = \frac{1}{N}\sum_{n=1}^{N}|w_n|^2 \tag{3.81}$$

normalized by the number of samples N [94, 101]. Assuming that the noise level is approximately constant across the total observed band $\mathcal{B}$, and with the knowledge that some subband $\mathcal{B}_l$ is signal free, i.e., consists of noise samples only, one could estimate the noise power for all of $\mathcal{B}$ by applying the above-presented noise estimator to samples from $\mathcal{B}_l$. Intuitively, the more samples are available to the estimator, the lower the estimation variance and the better the estimate (this intuition is

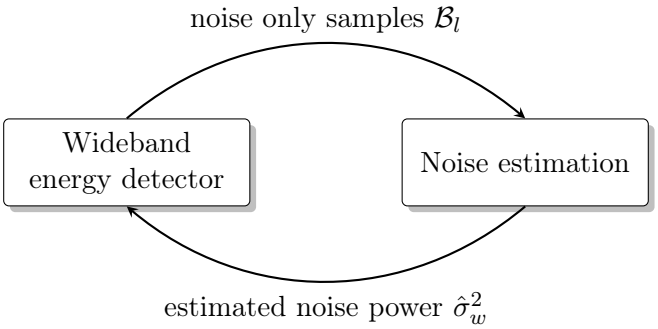

Figure 3.5.: Dependency loop in wideband energy detection and noise estimation.

supported by statics [101, 102]). Thus, one would ideally try to find some signal-free subband $\mathcal{B}_l$ that is as wide as possible and remains signal-free for the longest time possible, increasing the time-bandwidth product so that the maximum number of noise-only samples N can be used for the noise estimate.

Recalling the previously presented wideband detection model

$$\mathcal{H}_{0,l}: \quad \mathbf{x}_l = \mathbf{w}_l \tag{3.82}$$

$$\mathcal{H}_{1,l}: \quad \mathbf{x}_l = \mathbf{s}_l + \mathbf{w}_l, \tag{3.83}$$

where $l = 1, 2, \ldots, L$ denotes the l-th subband $\mathcal{B}_l$ of bandwidth B_l, the task of finding noise-only subbands directly emerges as the inverse task of signal detection. This is very convenient, as it allows relatively simple noise estimation using the results already available from spectrum sensing. One can use subbands classified as signal-free for noise estimation (hence the name *free-band noise estimation*).

Limitations

When using free-band noise estimation in combination with energy detection, multiple problems arise. Because energy detection relies on noise estimation and noise estimation, as just presented, relies on the results from energy detection, a dependency loop emerges (see fig. 3.5). Energy detection will inevitably make mistakes. More precisely, it is going to miss signals with the probability of miss P_M and wrongfully classify noise as signals with the probability of false alarm P_{FA}. Assuming some ED in the recent past predicted more signal samples to be noise than noise samples to be signals, then the energy contained in the supposed noise-only samples is higher than it would be if it only included noise-only samples. Consequently, the noise estimator will predict the estimated noise power $\hat{\sigma}_w^2$ to be higher than the actual noise power σ_w^2. Because the ED is going to base its future detection on $\hat{\sigma}_w^2$, which is higher than it should be, it will predict even more signal samples to be noise-only; thus closing the dependency loop. This can, depending on the system and the environment, lead to dramatic prediction errors. When starting the detection system, one encounters a similar situation: the ED requires a noise power estimate from the noise estimator, while the noise estimator needs signal-free samples from the ED. One somehow needs to break this dependency loop and *bootstrap* the system.

A solution to this problem could come from wideband eigenvalue detection (section 3.2.3). Because it does not rely on noise power estimates, eigenvalue detection breaks up the dependency loop presented above; while simultaneously providing very accurate spectrum sensing results. However, as already discussed, eigenvalue detection is not practically applicable to wideband sensing. Nevertheless, one could employ the *single bandpass filter eigenvalue sensing* method and use the energy detection results as a heuristic for the search algorithm that searches for free spectrum bands. This method is significantly slower than energy detection. However, because noise level changes are very slow phenomena (noise power can typically be assumed stationary for a couple of minutes [55, 59]) the speed of the wideband eigenvalue detection, using a tunable bandpass filter, appears to be sufficient for noise estimation. The here presented noise estimation method is novel, and it remains to be seen whether it is practically relevant.

3.3.2. Eigenvalue Noise Estimation

Due to the shortcomings of the above-presented noise estimation from noise-only samples, this section will introduce a method that can estimate noise from samples that contain noise *and* signals. Similar to eigenvalue detection, it utilizes the eigenvalue decomposition of the sample covariance matrix to separate noise and signal components in data samples [103, 104].

Sample Covariance Matrix

Since the signal $\mathbf{s}$ and the noise $\mathbf{w}$ are statistically independent, the covariance matrix $\mathbf{R}_{\widetilde{\mathbf{x}}}$ of size L can be decomposed into

$$\mathbf{R}_{\widetilde{\mathbf{x}}} = \mathbf{R}_{\widetilde{\mathbf{s}}} + \mathbf{R}_{\widetilde{\mathbf{w}}} = \mathbf{R}_{\widetilde{\mathbf{s}}} + \sigma_w^2 \mathbf{I}. \tag{3.84}$$

Denoting the eigenvalues of $\mathbf{R}_{\widetilde{\mathbf{x}}}$ in descending order from largest to smallest as $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_L$ and the eigenvalues of $\mathbf{R}_{\widetilde{\mathbf{s}}}$ from largest to smallest as $\rho_1 \geq \rho_2 \geq \ldots \geq \rho_L$, one finds that $\nu_j = \rho_j + \sigma_w^2 \ \forall j = 1, 2, \ldots, M$ and $\nu_j = \sigma_w^2 \ \forall j = M+1, M+2, \ldots, L$; where M is the number of signal-bearing eigenvalues in $\mathbf{R}_{\widetilde{\mathbf{x}}}$. The $L - M$ smallest eigenvalues of $\mathbf{R}_{\widetilde{\mathbf{s}}}$ are zero; therefore, all $L - M$ smallest eigenvalues of $\mathbf{R}_{\widetilde{\mathbf{x}}}$ contain only noise [61, 105]. If some portion B_l of the total bandwidth B is occupied by signals, one finds that $B_l/B = M/L$. Thus, the noise power σ_w^2 can theoretically be found by inspecting any of the $L - M$ smallest eigenvalues, as long as some portion of the wideband spectrum $\mathcal{B}$ is free from signals [103].

Because of the limited number of samples N, one can only compute the sample covariance matrix

$$\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}} = \frac{1}{N - L + 1} \sum_{n=L}^{N} \widetilde{\mathbf{x}}_n \widetilde{\mathbf{x}}_n^{\mathrm{H}}, \tag{3.85}$$

which diverges from the statistical covariance matrix for finite N (see section 3.1.2).

Minimum Description Length Eigenvalue Selection

The number of signal-bearing eigenvalues M of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ can be estimated by using the *minimum description length* (MDL) criterion developed in [106] and adapted to signal processing in [105]. MDL uses an information-theoretic approach to model selection, where the model of interest in the context of

noise estimation is one that best describes the eigenvalue distribution of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$. In other words: one is trying to find the minimum number of eigenvalues M needed to describe the correlation structure induced into $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ by the signals in $\mathbf{x}$. The general form of the MDL criterion is

$$\text{MDL} = -\log f(\mathbf{x}|\hat{\Theta}) + \frac{1}{2}k \log N, \tag{3.86}$$

where $f(\mathbf{x}|\hat{\Theta})$ is a parameterized family of probability densities from which the one that best fits the data needs to be selected. $\hat{\Theta}$ is the MLE of the model parameters, and k is the model size. The second term in eq. (3.86) can be thought of as a penalty term, penalizing larger model sizes. When observing data $\mathbf{x}$ the optimal model parameters are those that minimize the MDL.

When trying to detect the number of signals present in some observation vector $\mathbf{x}$ with the help of the eigenvalues ν_j of the sample covariance matrix $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$, the MDL was derived in [105] to be

$$\hat{M} = \arg\min_{M} -(L-M)N \log \frac{g(M)}{a(M)} + \frac{1}{2}M(2L-M)\log N, \tag{3.87}$$

where $g(M)$ and $a(M)$ are the geometric and arithmetic mean of the $L-M$ smallest sample eigenvalues respectively

$$g(M) = \prod_{j=M+1}^{L} \nu_j^{1/(L-M)} \tag{3.88}$$

$$a(M) = \frac{1}{L-M} \sum_{j=M+1}^{L} \nu_j. \tag{3.89}$$

Thus, after obtaining $\hat{M}$ from MDL, the noise related eigenvalues of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ can be assumed to be $\nu_j \; \forall j = \hat{M}+1, \hat{M}+2, \ldots, L$

Marchenko–Pastur Eigenvalue Fitting

Defining the *empirical distribution function* (EDF) of the eigenvalues of $\hat{\mathbf{R}}_{\widetilde{\mathbf{w}}}$ as

$$F^{\hat{\mathbf{R}}_{\widetilde{\mathbf{w}}}}(x) = \frac{\text{Number of eigenvalues of } \hat{\mathbf{R}}_{\widetilde{\mathbf{x}}} \leq x}{L} \tag{3.90}$$

for which it can be shown that $F^{\hat{\mathbf{R}}_{\widetilde{\mathbf{w}}}} \to F^{\mathbf{W}}$ almost surely for every x, as $N, L \to \infty$ with $c = L/N$ [107, 108]. $F^{\mathbf{W}}$ is known as the *Marchenko–Pastur* distribution (see appendix A.1.4). Its density will hereafter be denoted as $MP(c, \sigma_w^2)$, for some c and noise power σ_w^2. One needs to construct K linearly spaced values π_k, with $1, 2, \ldots, K$, in the range of $[\hat{\sigma}_{w,1}^2, \hat{\sigma}_{w,2}^2]$, where

$$\hat{\sigma}_{w,1}^2 = \frac{\nu_L}{(1+\sqrt{c})^2} \tag{3.91}$$

$$\hat{\sigma}_{w,2}^2 = \frac{\nu_{\hat{M}+1}}{(1+\sqrt{c})^2}, \tag{3.92}$$

and apply a goodness of fit test to find the optimal π_k

$$\hat{\sigma}_w^2 = \arg\min_{\pi_{\mathbf{k}}} \left\| dF^{\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}, L-\hat{M}}} - MP\left(\left(1 - \frac{\hat{M}}{L}\right)c, \pi_k\right) \right\|_2, \tag{3.93}$$

which yields the estimate of the noise power $\hat{\sigma}_w^2$ [103, 104]. $dF^{\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}, L-\hat{M}}}$ denotes the EDF density of the $L-\hat{M}$ smallest eigenvalues of $\hat{\mathbf{R}}_{\widetilde{\mathbf{x}}}$ and $\|\cdot\|_2$ is the L_2 norm.

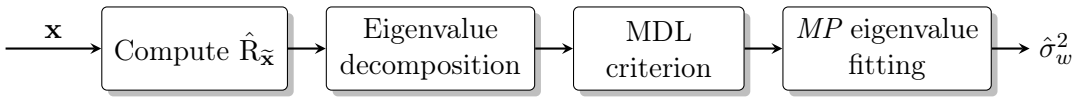

Figure 3.6.: Eigenvalue-based noise estimation block diagram.

Limitations

Similar to eigenvalue detection (section 3.1.2 and section 3.2.3), eigenvalue-based noise estimation relies on the eigenvalue decomposition of the sample covariance matrix. Thus, it is similarly computationally expensive and susceptible to correlations introduced by preprocessing and filtering. Because noise levels change relatively slowly and because only a single noise estimator is needed per WED, the increased complexity introduced by this estimator is manageable. The solutions introduced in section 3.2.3 to mitigate unwanted correlations are also directly applicable to the here presented eigenvalue noise estimation.

It is important to note that, while eigenvalue-based noise estimation involves two numerical optimizations, they are a lot less computationally expensive than the optimizations required by compressed sensing (see appendix B.2). The search space of possible values for the minimization consists of L values for eq. (3.87) and K values for eq. (3.93), with both $L, K \ll N$. Additionally, as shown in section 5.5, the fitting of eigenvalues to the Marchenko–Pastur distribution is generally not necessary; taking the average of the noise-only eigenvalues appears to be sufficient.

3.3.3. Wideband Energy Detection with Noise Estimation

Using the estimated noise power, the WED can be adapted to become independent from background noise. Normalizing the WED from section 3.2.2 with the estimated noise power $\hat{\sigma}_w^2$ and the number samples N_l from subband $\mathcal{B}_l$ results in

$$\mathrm{T}(\hat{\mathbf{x}}_l) = \frac{1}{\hat{\sigma}_w^2} \frac{1}{N_l} \sum_{k \in \mathcal{B}_l} |\hat{x}_k|^2 \underset{\mathcal{H}_{0,l}}{\overset{\mathcal{H}_{1,l}}{\gtrless}} \lambda_l. \tag{3.94}$$

Because the noise power $\hat{\sigma}_w^2$ is an estimated value, the statistics for the WED need to be adjusted to account for its probabilistic behavior. Seminal work in that regard was conducted in [101, 102, 109]. There, the energy detector statistics were modeled as the ratio of two scaled chi-square distributions, yielding a *Fisher–Snedecor* distribution $\mathcal{F}_{N_l,M_e}$, where N_l denotes the number of samples used by the energy detector itself and where M_e denotes the number of samples used for the noise estimate $\hat{\sigma}_w^2$; with N_l and M_e consisting of disjoint data samples. The exact statistics and their derivations would go beyond this work's scope and are thus not presented. Instead, only the approximate statistics will be shown. The combination of the *wideband* energy detector statistics (see section 3.2.2) with the noise estimation statistics from [101, 109] could not be found in related literature and thus appear to be novel.

CLT Statistics

The by the CLT approximated probability of false alarm $P_{FA,e,l}$ and probability of detection $P_{D,e,l}$ for a subband $\mathcal{B}_l$ with estimated noise variance are

$$P_{FA,e,l} = \mathrm{Q}\left(\frac{\lambda_{e,l} - 1}{\sqrt{\frac{N_l + M_e}{N_l M_e}}}\right) \tag{3.95}$$

$$P_{D,e,l} = \mathrm{Q}\left(\frac{\frac{\lambda_{e,l}}{1+\mathrm{SNR}_l} - 1}{\sqrt{\frac{N_l + M_e}{N_l M_e}}}\right). \tag{3.96}$$

The threshold $\lambda_{e,l}$ and the ROC can thus be written as

$$\lambda_{e,l} = \mathrm{Q}^{-1}(P_{FA,e,l})\sqrt{\frac{N_l + M_e}{N_l M_e}} + 1 \tag{3.97}$$

$$P_{D,e,l} = \mathrm{Q}\left(\frac{\mathrm{Q}^{-1}(P_{FA,e,l}) + \frac{1}{\sqrt{(N_l+M_e)/(N_l M_e)}}}{1 + \mathrm{SNR}_l} - \frac{1}{\sqrt{\frac{N_l+M_e}{N_l M_e}}}\right). \tag{3.98}$$

3.3.4. Conclusion

The here presented noise estimation methods make the WED applicable to real-world environments, where noise power levels are unknown. By using the adapted performance metrics it is possible to analytically model the WED with noise estimation. Because of its relative simplicity, when compared to free-band noise estimation, eigenvalue noise estimation will be chosen as the premier noise estimation method. Moreover, simulations conducted in section 5.5 show that it actually outperforms the free-band noise estimator in terms of estimation accuracy.

3.4. Summary

This chapter provided an overview of the most prominent narrowband and wideband spectrum sensing algorithms, as well as noise estimation methods; less relevant spectrum sensing methods were introduced in the accompanying appendix B. Wideband energy detection, combined with wavelet-based edge detection and noise estimation, emerged as the ideal wideband sensing method for application on SALSAT. It was introduced by first deriving appropriate narrowband detection methods and expanding them to wideband detection; while also introducing wavelet-based edge detection. The presented methods were evaluated and compared under the in section 1.4 presented requirements and constraints. The chapter ended with the presentation and evaluation of noise estimation methods. It was shown that the shortcomings of wideband energy detection could be compensated by complementing it with noise estimation and wavelet-based edge detection methods. The analytically derived detectors and their performance metrics will subsequently be verified through simulations in chapter 5.

4. Simulation Framework

The following chapter will present the simulation framework *SpecSens*, which has been developed in the course of this thesis. It consists of numerous functions and tools, aiding effective development and integration of spectrum sensing methods. It provides the foundation on which the spectrum sensing algorithms that were analytically derived in chapter 3 will be implemented and simulated (see chapter 5). With the integrated *Monte Carlo* simulation framework, algorithms can be simulated, verified, and compared.

SpecSens is written entirely in *Python* [110], utilizing several packages available through the Python package manager. Amongst others: *NumPy* [111], *SciPy* [112], *PyWavelets* [113] and *TracyWidom* [114]. More information regarding the software, such as version numbers, a complete package list, and installation procedures, can be found in the accompanying software repository [115]. The package structure is kept modular, thus easing development and simplifying software maintenance. In addition to SpecSens, numerous interactive *Jupyter Notebooks* [116] are provided in the software repository [115]. They demonstrate the operation of the developed algorithms and visualize the simulation results.

4.1. Python

Python has many advantageous features when compared to C/C++ or *Matlab*. In comparison to C/C++, Python, through its packages NumPy and SciPy, provides many powerful numerical data and signal processing functions. In combination with Jupyter Notebook and *Matplotlib* [117], Python becomes an interactive development environment, making scientific software development very effective. However, because Python is an interpreted language and relies heavily on multiple layers of abstraction, it is generally not as efficient as C/C++ [118]. While Python is a good choice for development and simulation, it is not well suited when implementing software in resource-constrained environments or developing real-time applications. Compared to Matlab, which is proprietary and closed source, Python and its packages are open source, freely available, and maintained, not by a single entity, but by a large community. Python is also a much richer language, providing many language features that Matlab does not have (e.g., namespaces, package manager, named arguments, etc.). These features make software development more productive, especially in large projects.

4.2. Test Data Generation

Simulating algorithms on data makes it possible to evaluate them and compare their performance. Because at the time of writing this thesis, no real data from SALSAT was available (see section 1.3.1), data samples were generated using the following methods.

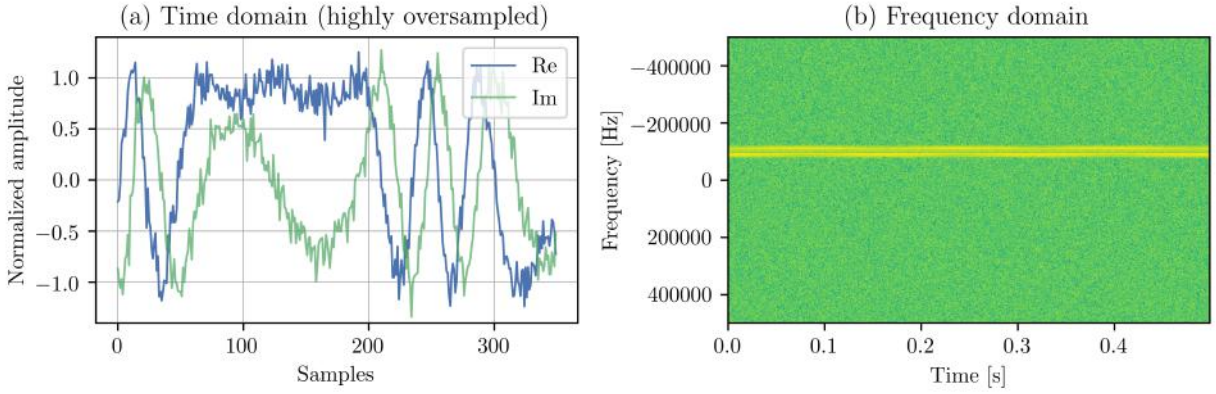

Figure 4.1.: Snapshot of an oversampled time (a) and frequency (b) representation of a randomly generated complex signal (eq. (4.1)) with additive CWGN at $\mathrm{SNR}_{\mathrm{dB}} = 0$. Signal parameters for (b): $f_c = -100\,\mathrm{kHz}$, $f_m = 3.9\,\mathrm{kHz}$, $f_d = 15\,\mathrm{kHz}$, $f_s = 1\,\mathrm{MHz}$.

Noise

Complex white Gaussian noise (CWGN) samples $\mathbf{w}$ are generated by drawing N 2-dimensional samples from a multivariate real Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, with $\boldsymbol{\mu} = \left(\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right)$ and $\mathbf{C} = \left(\begin{smallmatrix} \sigma_w^2/2 & 0 \\ 0 & \sigma_w^2/2 \end{smallmatrix}\right)$ (see appendix A) [60, 65]. The noise power σ_w^2 can either be provided in linear or decibel $\sigma^2_{w,\mathrm{dB}}$ form. If provided in decibels, power values are converted to linear values according to $\sigma_w^2 = 10^{\sigma^2_{w,\mathrm{dB}}/10}$. Besides the sample frequency f_s (in Hz), which constitutes the sample bandwidth, one can provide the desired length either in seconds s or in number of complex samples N.

It is important to note that power and energy (for both noise and signals), in the context of this simulation framework, do not have any real physical meaning or physical unit. They are simply mathematical concepts arising when transforming dimensionless numerical values in ways that would, if one were using units with actual physical dimensions, yield the dimensions normally associated with those mathematical transformations.

Narrowband Signal

Narrowband signals are generated according to [119], which provides common test signals for cognitive radio and spectrum sensing research, used by many authors [33, 52, 61, 73]. The signals from [119] are adapted to the complex I/Q domain (similar to the samples generated in [120]), yielding the following signal equation

$$x(t) = \exp\left(i\left(2\pi f_c t + \frac{f_d}{f_m}\sin(2\pi f_m t) + 2\pi\varphi\right)\right), \tag{4.1}$$

where $\varphi \in [0, 1)$ is the random phase, drawn from a uniform distribution. The time variable t (in s) is throughout this thesis referred to by its discrete-time counterpart n. f_c, f_m and f_d are the center frequency, the modulation frequency, and the deviation frequency (in Hz), respectively.

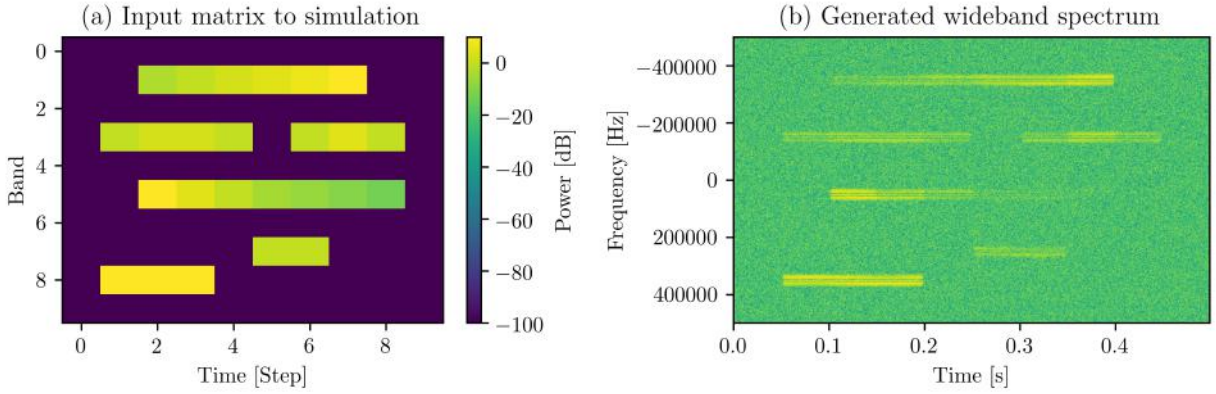

Figure 4.2.: (a) is a visualization of the in eq. (4.2) presented power matrix and (b) the resulting frequency representation of the complex wideband signal and additive CWGN with $\sigma^2_{w,\mathrm{dB}} = 0$ (also see fig. C.1).

The resulting discrete signal vector $\mathbf{x}$ is normalized and scaled by the signal power σ_s^2 (or $\sigma^2_{s,\mathrm{dB}}$), analogous to the above-presented noise data generation. One can also set the sample frequency f_s and signal length as desired.

Figure 4.1 (a) shows a exemplary complex narrowband signal with additive CWGN (which is highly oversampled for illustrative purposes). The frequency-domain visualization in fig. 4.1 (b), commonly known as a spectrogram (see appendix A.3.5), was produced by employing a STFT with a 1024-point flattop window.

Wideband Signal

Since wideband spectrum sensing methods need to be evaluated, wideband signals are required (see section 3.2). The here generated wideband signals consist of multiple narrowband signals (based on the narrowband signal from fig. 4.1), interleaved in frequency and time. To control the frequency, time, and power, with which a single narrowband signal appears in the wideband signal, a power-matrix $\mathbf{P}$ is supplied to the wideband signal generator. An exemplary power-matrix $\mathbf{P} \in \mathbb{R}^{10\times 10}$ is

$$\mathbf{P} = \begin{pmatrix} -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & -3 & 0 & 3 & 5 & 7 & 10 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 \\ -100 & 0 & 3 & 3 & 0 & -100 & 0 & 5 & 0 & -100 \\ -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & 10 & 5 & 0 & -5 & -7 & -10 & -13 & -100 \\ -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 & 0 & 0 & -100 & -100 & -100 \\ -100 & 10 & 10 & 10 & -100 & -100 & -100 & -100 & -100 & -100 \\ -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 & -100 \end{pmatrix}, \tag{4.2}$$

with each of the 10 rows resembling the power level (in dB) of a single subband over time, and where each column represents a snapshot of the wideband signal at a specific point in time. A power level of $-100\,\mathrm{dB}$ results in virtually no signal being present, while power levels of around $0\,\mathrm{dB}$ represent moderately powerful signals. For this example, the total signal bandwidth is $1\,\mathrm{MHz}$, and the signal length is $0.5\,\mathrm{s}$; the signal thus consists of 500×10^3 complex samples.

A graphical power level representation of $\mathbf{P}$ can be seen in fig. 4.2 (a). The resulting, generated wideband signal, with additive CWGN, is visualized in fig. 4.2 (b). Power level smoothing is required between different timesteps to avoid high-frequency peaks introduced by sharp power level changes. While this matrix-based wideband signal generation method is not very versatile, as it only allows for the generation of checkerboard-like wideband signal patterns, it is nonetheless adequate for this thesis's simulation purposes. In particular, the matrix representation allows for practical performance evaluation of wideband spectrum sensing methods by comparing the power-matrix with the detection-result-matrix (see section 5.2).

Doppler Shift Signal

Of particular interest to SALSAT, and space-related RF research, in general, are *Doppler-shifted* signals. Doppler shift is caused by the transmitter's relative motion to the receiver (see fig. 4.3 (a)). This results in shifted frequencies, proportional to the relative speed of the transmitter and the receiver. Because the relative speeds of objects in space can be very high, Doppler-shifted signals are a common occurrence in space. Thus, SpecSens incorporates a Doppler shift signal generator, utilizing the following relations

$$x = (0.5T - t)v \tag{4.3}$$

$$\alpha = \arctan(x/d) \tag{4.4}$$

$$\sigma^2_{s,\mathit{shift},\mathrm{dB}} = -\left|\frac{\sqrt{d^2+x^2}}{0.5Tv}\right|^3 \sigma^2_{s,\mathrm{dB}} \tag{4.5}$$

$$f_{c,\mathit{shift}} = \frac{2\alpha v}{c\pi} f_{\mathit{carr}} + f_c, \tag{4.6}$$

where c is the speed of light in vacuum (in m/s) and $t \in [0, T]$ the time (in s). v is the relative speed and d the minimum distance of the transmitter and receiver (in m/s and m, respectively). This results in scaled signal power levels $\sigma^2_{s,\mathit{shift},\mathrm{dB}}$ and shifted center frequencies $f_{c,\mathit{shift}}$. An exemplary Doppler shifted signal, with realistic physical values, is visualized in fig. 4.3 (b).

4.3. Monte Carlo Simulation

To verify the analytically derived statistics from chapter 3, such as the P_{FA} and P_D equations, quantitative *Monte Carlo* (MC) simulations are necessary [121]. They aid in surveying algorithms and comparing their performance. MC simulations rely on repeated random sampling to obtain numerical results that characterize the underlying statistical phenomenons. Because large numbers of simulations need to be conducted for MC to be effective, the here employed MC simulations are implemented using multithreading, employing Python's multiprocessing framework. This enables the parallel execution of multiple simulation environments on all available CPU cores and

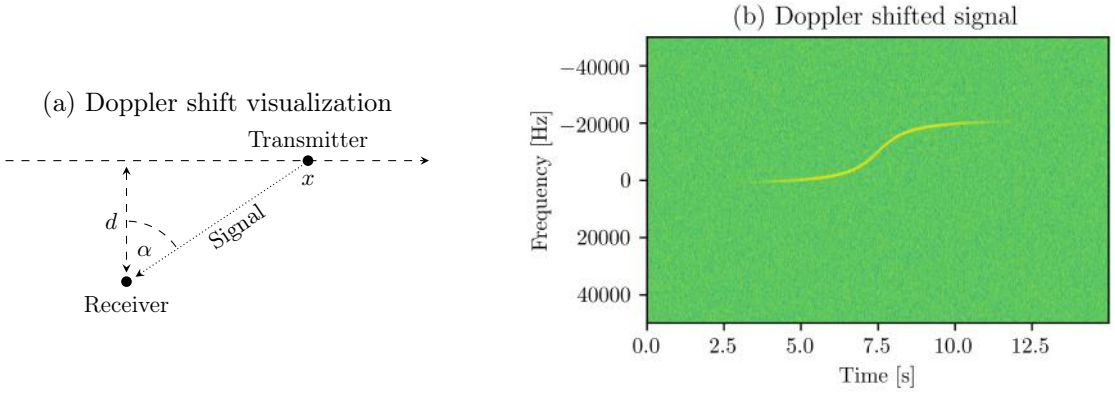

Figure 4.3.: Doppler signal with $v = 8\,\mathrm{km/s}$, $d = 6\,\mathrm{km}$, $f_{\mathrm{carr}} = 450\,\mathrm{MHz}$ and $\sigma^2_{s,\mathrm{dB}} = 0$. $f_c = -10\,\mathrm{kHz}$, $f_m = 32\,\mathrm{kHz}$, $f_d = 50\,\mathrm{Hz}$. $f_s = 100\,\mathrm{kHz}$, sampled over 15 s seconds, noise power at $\sigma^2_{w,\mathrm{dB}} = 5$ (also see fig. C.2).

thus yields a significant speedup compared to single-threaded implementations. Synchronization and data distribution are handled internally by the SpecSens package, providing a convenient and modular simulation interface. All simulation-related parameters can be passed as arguments to SpecSens, allowing for a clean programming interface. Thanks to a deterministic *tree-like* replication and distribution of entropy, a single *random seed* characterizes the complete multithreaded simulation, allowing for reproducible random simulations. With the help of NumPy's *seed sequencing*, correlation in parallel running random number generators is mitigated.

The MC simulation framework provides two levels of parallelization: *generations* and *iterations* (see appendix D). A generation can be thought of as a specific RF environment with a set of fixed parameters. Parameters are randomly generated using a prespecified range of possible parameter values and are generally drawn from a Gaussian distribution. Possible parameters are signal and noise power levels, as well as noise uncertainty. Every generation performs a specified number of iterations, where each iteration consists of generating data samples from the generation's RF environment, running the desired spectrum sensing method, and recording the detection outcome. After all iterations for a specific generation have been performed, the generation will store the simulation results. Once all generations have successfully executed, the resulting statistics are computed and visualized (see chapter 5).

5. Implementation and Simulation

Having introduced the theoretical concepts behind relevant spectrum sensing methods in chapter 3, this chapter will continue with their implementation and simulation. Following a similar path as chapter 3, the narrowband energy detector (ED) will be discussed first. Next, the concepts from narrowband energy detection will be expanded to wideband energy detection, followed by simulations of wavelet-based spectrum sensing. Afterwards, narrowband and wideband eigenvalue detection, utilizing *finite impulse response* (FIR) filters and whitening matrices, will be presented. These concepts lay the foundation for the then-following implementation and simulation of noise estimation methods. At last, wideband energy detection is combined with noise estimation to yield the *final* wideband spectrum sensing algorithm.

The following simulations are conducted with SpecSens, the simulation framework presented in chapter 4. They are run on an Intel Core i7-6700K processor at up to 4.20 GHz, with Ubuntu 18.04 installed. All 8 logical cores are utilized, with no other relevant tasks running during simulations. The software versions of the employed packages, and Python itself, are documented in the accompanying software repository. Because of the breadth of the here investigated spectrum sensing methods, only a small fraction of all conducted simulations and visualizations will be presented. Many more, accompanied by in-depth explanations, can be found in the form of Jupyter Notebooks in the mentioned software repository (see [115]).

5.1. Energy Detection

This section starts by verifying the analytically derived chi-square and CLT statistics from section 3.1.1. The ED used in the following simulations is implemented in Python according to eq. (3.6).

5.1.1. Chi-Square Statistics

The threshold for the chi-square ED is determined using eq. (3.11). The following parameters were used for the simulation: generations = 200, iterations = 300 (total iterations = 60×10^3), signal length = 0.001 s ($N = 1000$), $\mathrm{SNR}_{\mathrm{dB}} = -10$ ($\sigma^2_{w,\mathrm{dB}} = 10$, $\sigma^2_{s,\mathrm{dB}} = 0$), $P_{FA} = 0.1$. All other signal parameters were identical to the signal presented in fig. 4.1 (b). This resulted in the following values, which were analytically computed before starting the simulation: $\lambda = 13102.0687$, $P_D = 0.9577$. The simulation took 9.48 s and calculated the following numerical values: $P_{FA} = 0.0998$, $P_D = 0.9591$. Figure 5.1 (a) and (b) depict the convergence of the running averages of the per generation calculated P_{FA} and P_D values to their theoretical counterparts. They indicate that

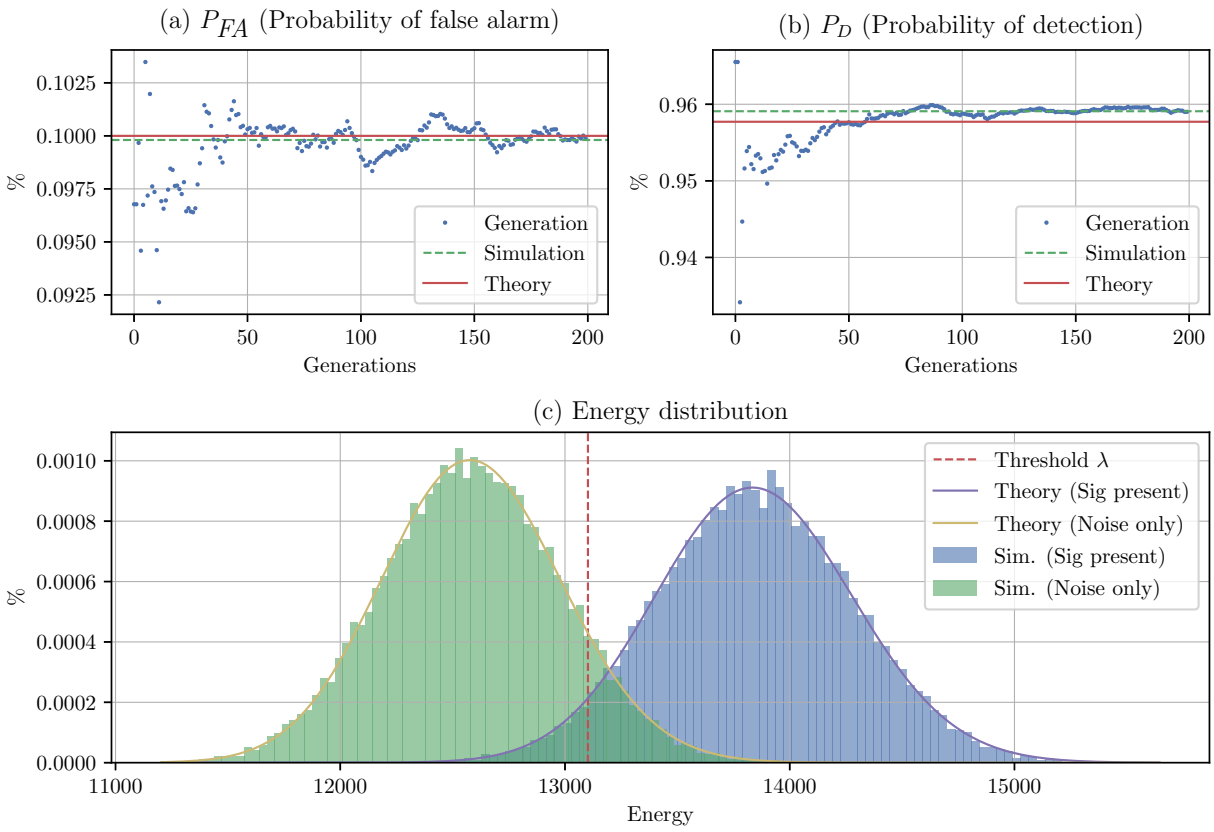

Figure 5.1.: (a) and (b) depict the convergence of the numerical P_{FA} and P_D values to their theoretical counterparts with an increasing number of MC simulation generations. The blue dots are the running average after each generation. In (c), the probability densities and histograms of the theoretical and numerical energies are shown. The red vertical line is the calculated threshold λ to achieve the desired $P_{FA} = 0.1$.

theory and simulation agree with one another. In fig. 5.1 (c) one can see how the histograms of the calculated energies from all simulation iterations overlap with the distributions underlying the theoretical derivations.

Figure 5.2 visualizes the impact of different sample sizes N and different $\mathrm{SNR}_{\mathrm{dB}}$ values on the ROC derived in eq. (3.13). One can see that (theoretically) any desired P_{FA} and P_D pair can be achieved by merely increasing the number of samples N. However, this comes at the cost of increased sensing time. Additionally, the impact of noise uncertainties results in the so called *SNR wall*: an uncertainty threshold below which a detector will fail to be robust, no matter how long it can observe [93, 94].

5.1.2. Noise Uncertainty

As already alluded to in chapter 3, EDs rely on the precise knowledge of background noise power levels. Introducing 1 dB of noise uncertainty, meaning the noise power is not known precisely but statistically distributed as $\sigma^2_{w,\mathrm{dB}} \sim \mathcal{N}(10, 1)$, renders the ED completely useless. The results of

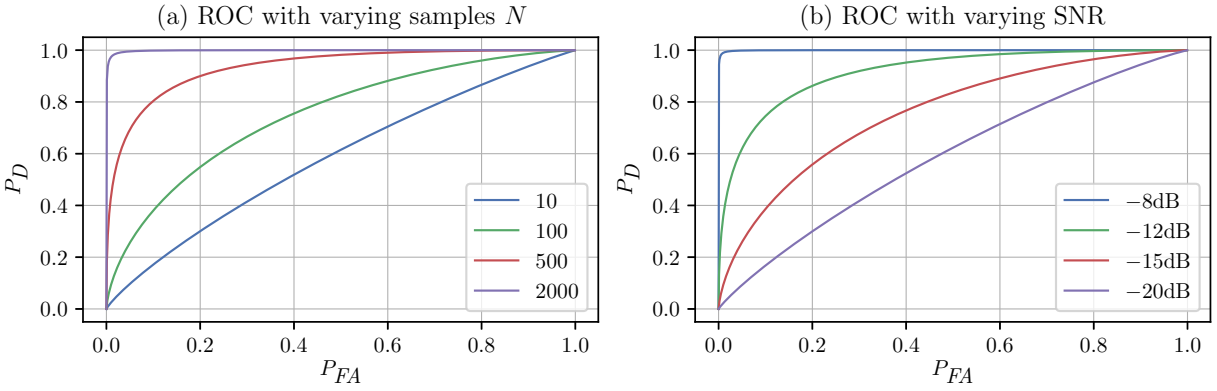

Figure 5.2.: ROCs with varying number of samples N and SNR (in dB) under ideal conditions. SNR is set to $-10\,\text{dB}$ for variable N, and $N = 1000$ for variable SNR.

the simulation with 1 dB of noise uncertainty can be seen in fig. 5.3, where all other simulation parameters are identical to the previous simulation (fig. 5.1). In fig. 5.3 (a) and (b) one can see how the numerically computed P_{FA} and P_D values converge to different values than analytically calculated, and that the energy distributions in fig. 5.3 (c) deviate significantly. This phenomenon is caused by the above mentioned SNR wall. Because noise power levels are not precisely known in real-world applications, the ED is not practically feasible if used by itself.

5.1.3. CLT Statistics

This section will compare the performance of the chi-square ROC (see eq. (3.13)) with the CLT ROC (see eq. (3.24)). The chi-square statistics are analytically precise, while the CLT is only an approximation that can be reliably used when the number of samples N is large. As can be seen in fig. 5.4 (a), with sample size $N = 500$, no significant difference between the two statistics emerges. The CLT ROC (almost) perfectly approximates the chi-square ROC. However, when the sample size is small, as in fig. 5.4 (b), the CLT approximation becomes very poor. Because sample sizes will almost always be in a range where these errors are negligible, only the CLT statistics will be used hereafter. They are simpler, allow for more parameters to be analytically derived, and require less computational resources to evaluate (see section 3.1.1).

5.2. Wideband Energy Detection

The following simulations of the wideband energy detector (WED) are generated using the theoretically derived results from section 3.2.2. More precisely, the spectral estimation is based on the in section 3.2.2 presented methods using a custom STFT utilizing the FFT provided by SciPy. The custom STFT employs a non-overlapping 1024-point windowing function. The accompanying WED is implemented in Python according to fig. 3.3 and eq. (3.55). The P_{FA}, P_D, λ, etc., are based on the in section 3.2.2 presented statistics.

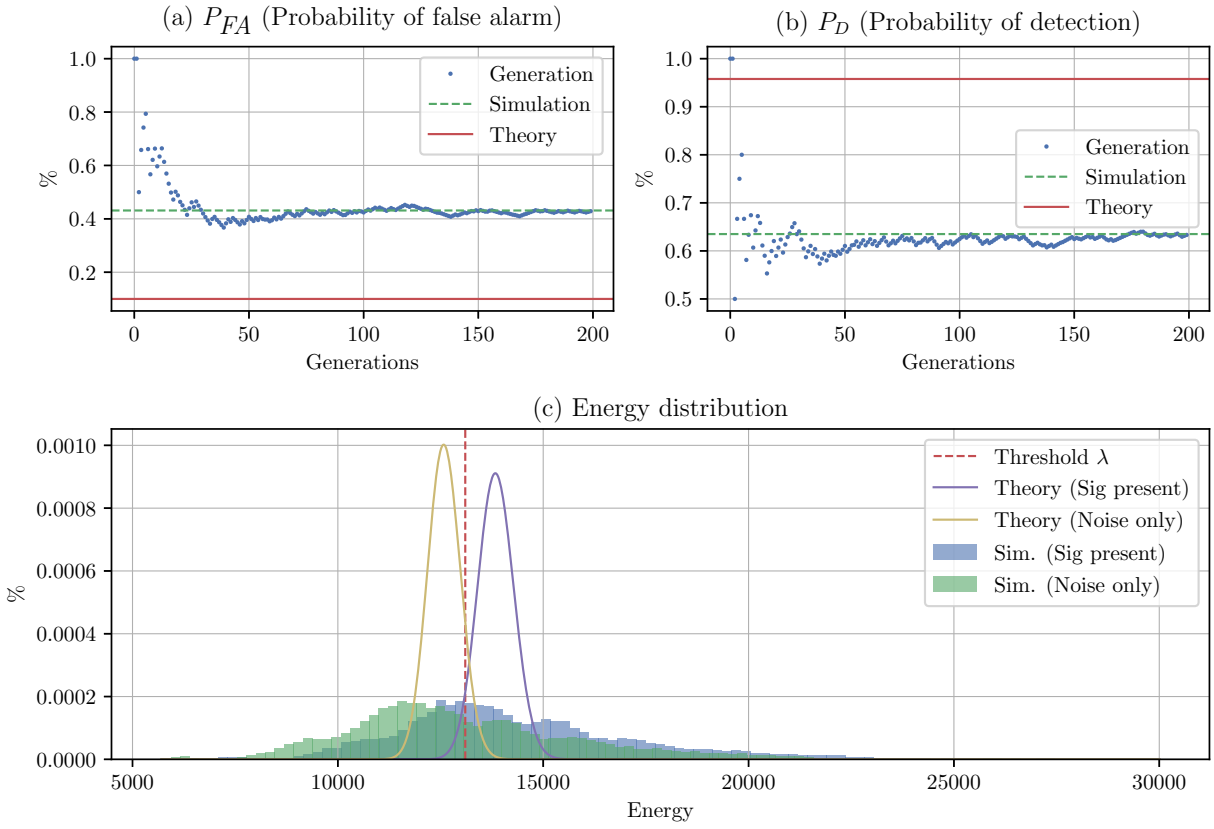

Figure 5.3.: Simulation result with noise uncertainty. One can see that the average simulation values do not converge to the theoretical values (a) (b) and that the numerical energy distributions strongly diverge from the theoretical ones (c).

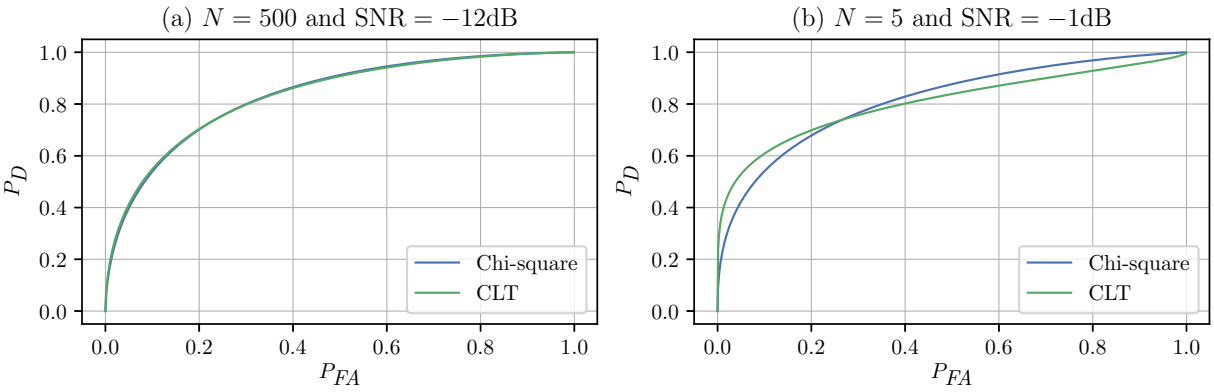

Figure 5.4.: Comparing the impact of sample size N on the accuracy of the CLT statistics.

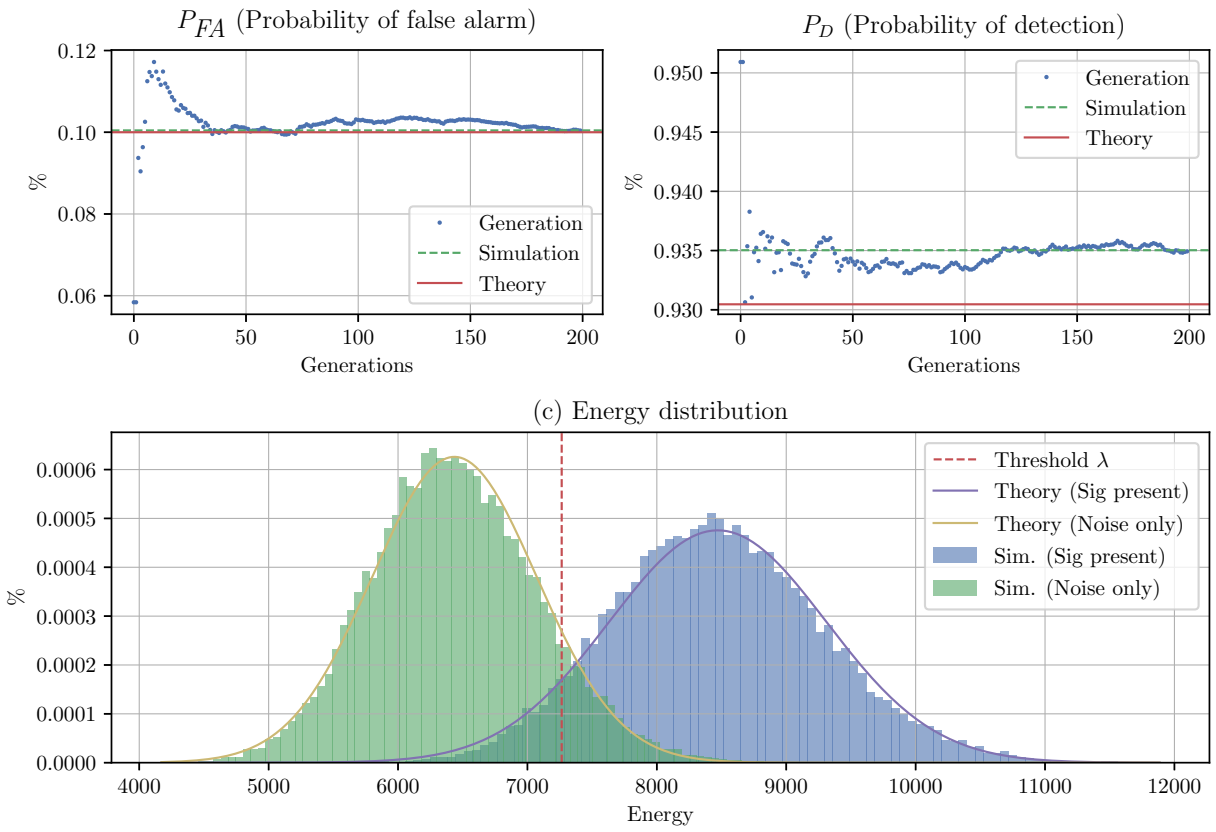

Figure 5.5.: Simulation results of the wideband energy detector using a rectangular window with $N = 1024$ and $P_{FA} = 0.1$.

5.2.1. Rectangular Window

The simulation shown in fig. 5.5 used the following parameters: generations = 200, iterations = 300 (total iterations = 60×10^3), signal length = 0.001 024 s ($N = 1024$), $\text{SNR}_{\text{dB}} = -15$ ($\sigma^2_{w,\text{dB}} = 18$, $\sigma^2_{s,\text{dB}} = 3$), $P_{FA} = 0.1$. The WED monitored $L = 10$ separate non-overlapping bands, each with a bandwidth of 100 kHz (see section 4.2). The simulation results were generated by observing band $l = 3$, where a signal with $f_c = -150$ kHz was randomly generated. All other signal parameters are identical to the signal presented in fig. 4.1 (b). The simulation took 14.6 seconds to execute. As can be seen in fig. 5.5 (a) and (b), the analytically derived $P_{FA} = 0.1$ and $P_D = 0.9305$, using $\lambda = 7264.7314$, were well met by the simulated $P_{FA} = 0.1005$ and $P_D = 0.9350$. Observing the energy distributions in fig. 5.5 (c), one can see that the histograms slightly deviate towards lower energy values when compared to the theoretical curves. This is likely due to a minuscule amount of the signal's energy leaking into other subbands $l \neq 3$. The curves still fit the histograms in the overlapping areas (close to the threshold λ) well, resulting in relatively accurate statistics.

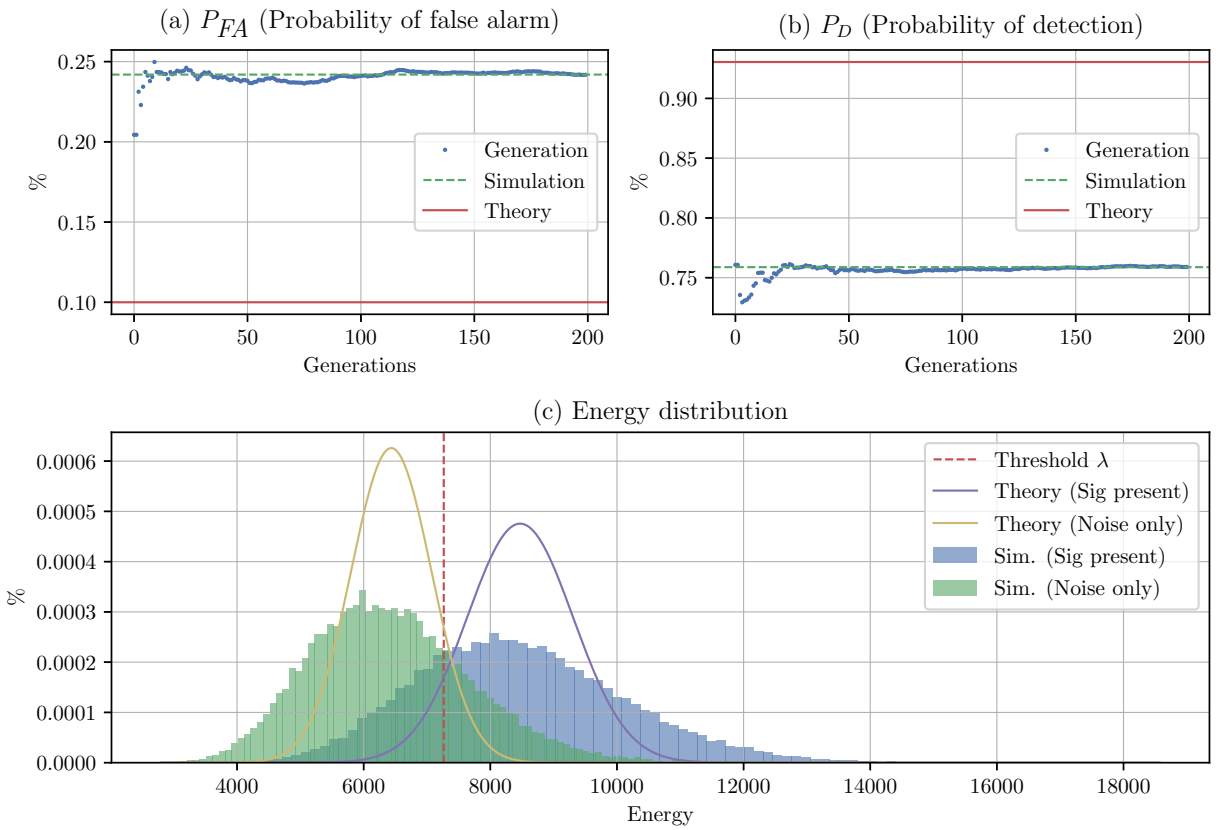

Figure 5.6.: Simulation results of the wideband energy detector using a $N = 1024$ flattop window (see appendix A.3.5). The simulated energy distribution (c) strongly deviates from the theoretically derived curves, leading to entirely wrong statistics in (a) and (b).

5.2.2. Other Windows

When using windowing functions other than the rectangular window, such as the flattop window, the statistics of the WED break down. The means μ of the energy distributions do not significantly change compared to the simulations employing the rectangular window. However, the standard deviations σ, depending on the type of window, increase by a factor of about 1.6× to 3.5×. This observation appears to be novel in the context of wideband energy detection, as no mention of this phenomenon could be found in related literature. The change in variance is likely caused by the change in the energy distribution of the PSD induced by the different window types. It is currently not possible to analytically describe this phenomenon, since there exists no literature about the impact of windowing functions on the performance statistics of WEDs. Thus, only rectangular windows will be employed hereafter.

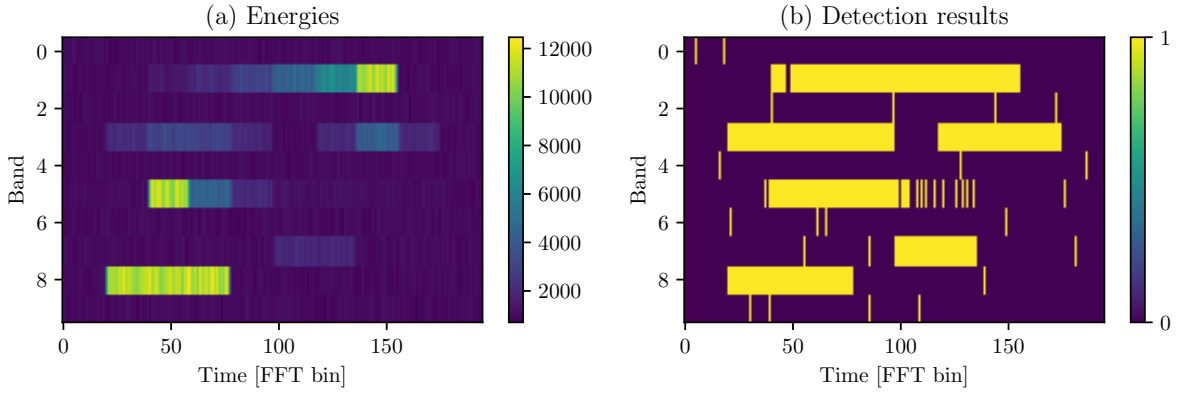

Figure 5.7.: Energy detection using a wideband signal generated with the power matrix from eq. (4.2), with $\sigma^2_{w,\mathrm{dB}} = 10$, $P_{FA} = 0.01$ and $L = 10$ subbands. The detector in (b) struggles to detect the low power signals in subband $l = 5$, starting with STFT bins at around 100.

5.2.3. Wideband Detection Visualization

To better comprehend and interpret wideband energy detection, this section introduces a visually appealing representation of the detection results. Based on the in eq. (4.2) depicted wideband signal power matrix, a 0.2 second long ($N = 200 \times 10^3$) wideband signal, otherwise identical to the one visualized in fig. 4.2, was generated and combined with additive CWGN at $\sigma^2_{w,\mathrm{dB}} = 10$. A $L = 10$ band WED, employing a 1024-point rectangular window, was applied, resulting in a 10×195 energy matrix visualized in fig. 5.7 (a). One can see the signal components in the corresponding subband at their respective energy levels. Following that, a threshold based on the in section 3.2.2 presented equations using $P_{FA} = 0.01$ was applied to the in fig. 5.7 (a) depicted energy levels, resulting in the in fig. 5.7 (b) illustrated detection results. The detector can almost perfectly reconstruct the presence of the originally modulated wideband signal. However, one can see that the *non-zero* P_{FA} value introduces spurious false detections. The detector is also not able to reconstruct the low power wideband signal in band $l = 5$ starting with STFT bins at around 100. As depicted in eq. (4.2) the signal power levels in these timesteps are ranging from $\sigma^2_{s,\mathrm{dB}} = -5$ to $\sigma^2_{s,\mathrm{dB}} = -13$. For $\sigma^2_{s,\mathrm{dB}} = -5$ the $P_D = 0.7$, while for $\sigma^2_{s,\mathrm{dB}} = -10$ it is only at $P_D = 0.096$; thus, resulting in poor sensing performance. The available samples $N = \lfloor 1024/10 \rfloor = 102$ per STFT bin and subband are not enough to discriminate between the low signal power levels and the relatively high noise power. One could increase the P_D by either increasing the allowed P_{FA}, which would result in more false detections, or by increasing the number of samples per subband through longer STFTs or fewer subbands (at the cost of higher latency and less resolution, respectively; see the ROC curves in fig. 5.2).

When using a higher resolution WED with $L = 100$ subbands, even very fine signals, such as the in fig. 4.3 presented Doppler signal, can be detected. Because the overall signal bandwidth is lower ($f_s = 100\,\mathrm{kHz}$), the STFT window length is reduced to 512. The resulting energies and detections can be seen in fig. 5.8 (a) and (b) respectively. Because of the relatively high SNR, the signal was well detected, despite the small STFT bin sizes.

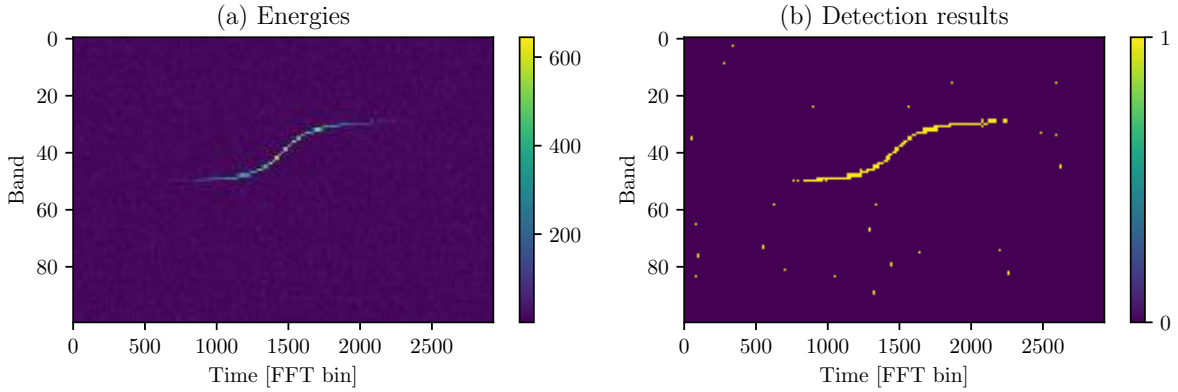

Figure 5.8.: Wideband energy detection using a 512-point rectangular STFT window applied to the Doppler signal from fig. 4.3, with $\sigma^2_{w,\mathrm{dB}} = 5$ and $\sigma^2_{s,\mathrm{dB}} = 0$. Using $P_{FA} = 0.01$ and $L = 100$ subbands, the Doppler signal is well detected by the wideband energy detector.

5.3. Wavelet Detection

This section presents the simulation results of the wavelet spectrum sensing method from section 3.2.4. Wavelet edge detection locates spectrum holes by analyzing irregularities in the estimated PSD using the WT (see appendix A.3.6).

5.3.1. Single PSD

The PSD of a random wideband signal is depicted in fig. 5.9 (a). It was generated using the SpecSens simulation framework by employing a 1024-point rectangular windowed STFT on a wideband signal with different subsignal power levels $\sigma^2_{s,\mathrm{dB}} = 0 \ldots 10$ and $\sigma^2_{w,\mathrm{dB}} = 0$ additive CWGN. To reduce the impact of the noise and to "sharpen" spectral edges, 10 consecutive PSDs were averaged together (similar to *Welch's method* [122]). The resulting "cleaner" PSD can be seen in fig. 5.9 (b). The number of consecutive averaged PSDs needs to be carefully tuned depending on signal shapes, the maximum tolerable detection latencies, and the SNR. Wavelet edge detection is applied to the PSD from fig. 5.9 (b) using the WMP of the first derivative Gaussian wavelet with $M = 4$ (see section 3.2.4). The resulting normalized spectral peaks, together with the peak detection results (red dots), are visualized in fig. 5.9 (c). The peak detector is based on SciPy's build-in peak detection. To reduce the detection of false peaks, a minimum required distance (denoted as δ in eq. (3.78)) and minimum height are set (see [99] for a similar procedure). If a peak is detected, any smaller peaks in the range of $\pm 20\,\mathrm{kHz}$ are dominated by the larger peak and ignored. Peaks are also only considered for detection when reaching above 0.1 in the normalized detection spectrum (fig. 5.9 (c)). This results in relatively stable detection of edges under reasonable SNRs. The detected spectral edges, together with the original PSD, are shown in fig. 5.9 (d).

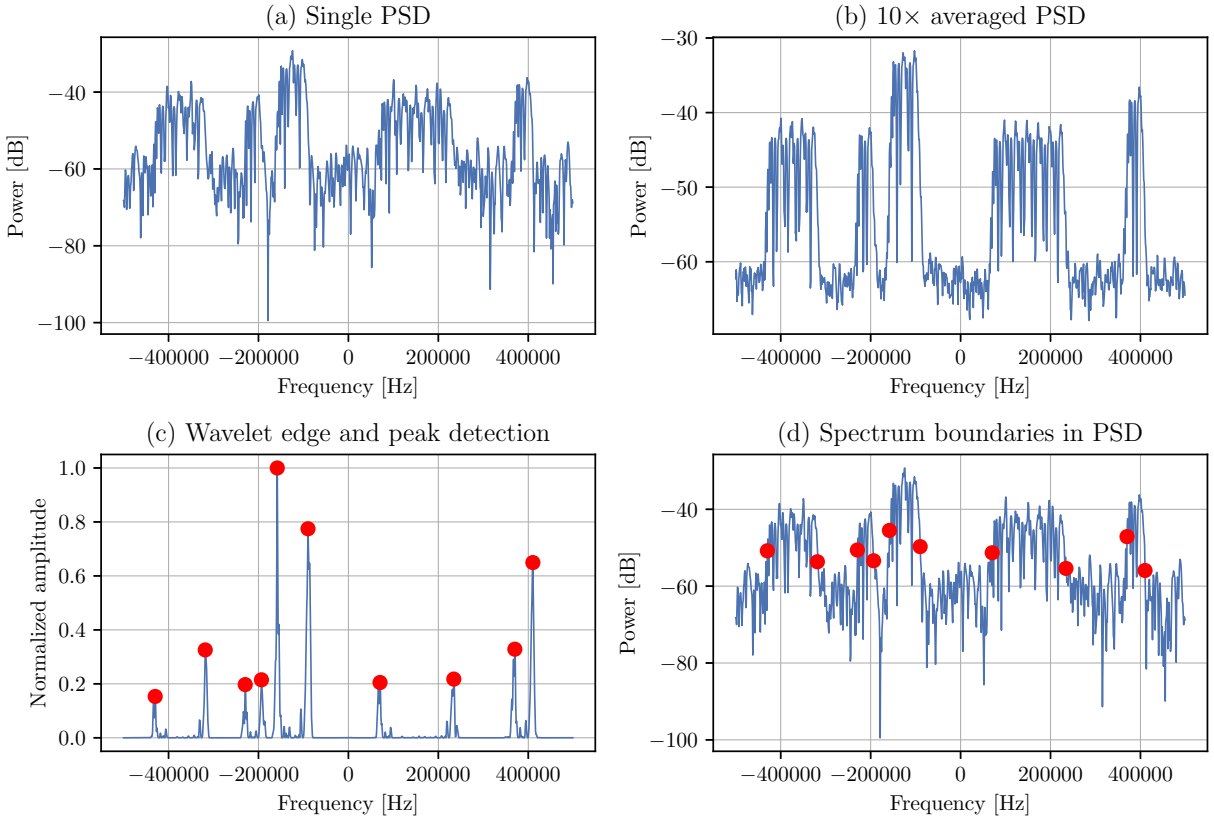

Figure 5.9.: Visualization of wavelet edge detection. Multiple PSDs, all similar to (a), are averaged to produce (b). Wavelet edge detection and peak detection are applied to (b), resulting in (c). The detected spectral edges in the original PSD are visualized in (d).

5.3.2. Spectrogram

Through SpecSens, the in fig. 5.10 (a) depicted random power matrix is converted into a wideband signal with additive CWGN as shown in fig. 5.10 (b). Applying many consecutive edge detections to each PSD results in spectral edges, as shown in fig. 5.10 (c). The wavelet edge detection was again applied to a 10× averaged PSD using a PSD history buffer. It is possible to identify the detected edges' *delayed start* on the left side of signals first appearing and *overhang* on the right side of fading signals. This phenomenon is introduced by the PSD averaging. Increasing the size of the PSD history buffer further exacerbates this effect while at the same time reducing the impact of noise and providing more stable detection results. When no signal in the wideband spectrum is present, the normalization of the resulting spectral peaks leads to a dramatic increase of false detections (see 0.4 s to 0.45 s in fig. 5.10 (c)). Wavelet edge detection itself is not able to reliably prevent these false detections.

However, when combining wavelet detection with wideband energy detection, false detections can be removed by comparing the corresponding subband energy content against an energy detection threshold. This leads in much cleaner detection results, as depicted in fig. 5.10 (d). Subbands not "passing" the energy detection threshold are discarded and not further considered. Incorporating wavelet edge detection into wideband energy detection reduces the required bandwidth needed to

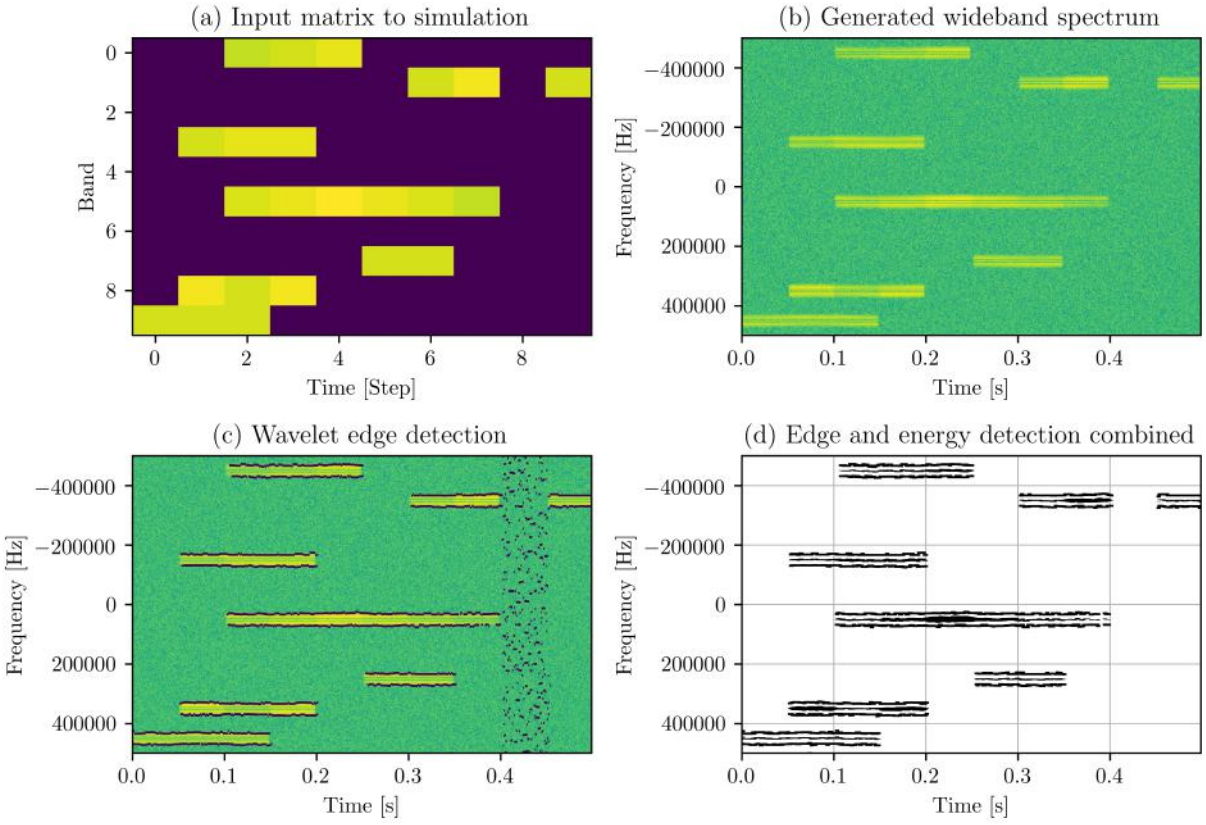

Figure 5.10.: The wideband spectrum in (b) contains the spectral representation of a wideband signal generated by SpecSens using the in (a) depicted power matrix. The wavelet edge detection results are shown in (c) and combined with energy detection in (d). The centerline of a detected signal indicates its relative strength: the stronger the signal, the thicker the centerline.

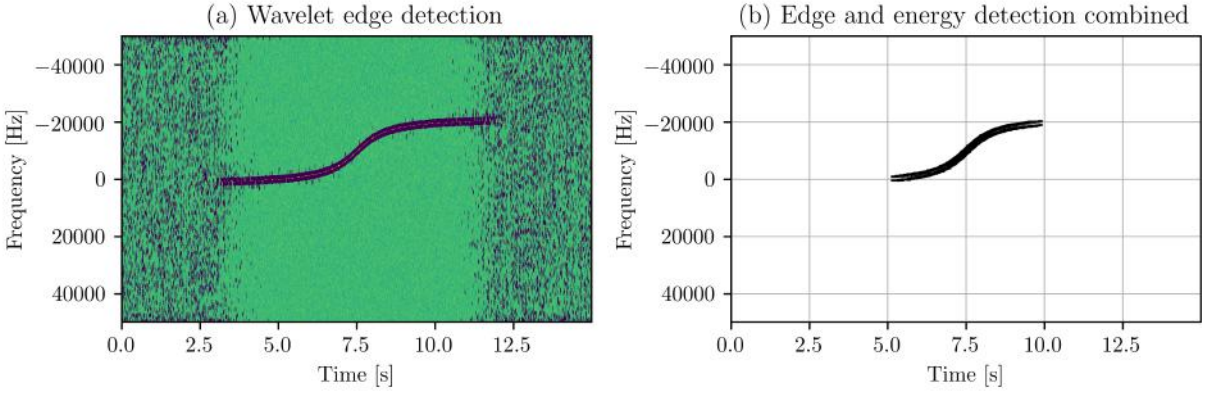

Figure 5.11.: Wavelet-based edge detection applied to the Doppler signal from fig. 4.3. Wavelet edge detection requires less bandwidth when running on sparse signals compared to wideband energy detection.

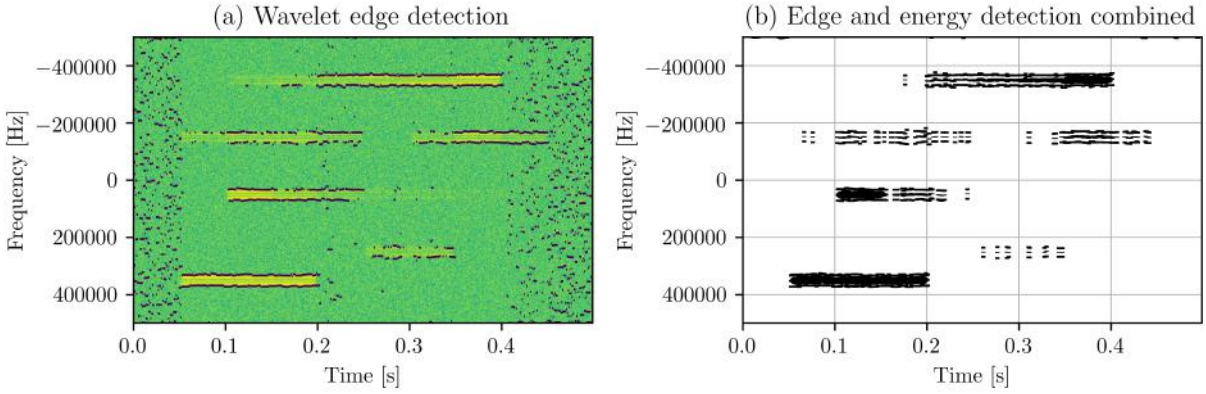

Figure 5.12.: When applied to low SNR signals, wavelet-based edge detection performance is worse than standalone wideband energy detection. Comparing the detection results with fig. 5.7 (b), one can see that wavelet edge detection misses more signals.

store or transmit the detection results as the sparsity of the wideband spectrum can be exploited. This enables high-resolution detection with minimal resource demands (CT-06), when compared to standalone WEDs. A prime example of this can be seen in fig. 5.11 (a) and (b). Wavelet edge detection combined with wideband energy detection was applied to the same Doppler shifted signal as shown in fig. 5.8. While the WED from fig. 5.8 returned a 100 entry long array for every detection step (mostly filled with zeros), the wavelet edge detector returned a linked list of *signal objects*[1]. When using wider bands or detecting with higher resolution, this reduction in required bandwidth becomes even more apparent. It is important to also note, that combining wavelet edge detection with wideband energy detection results in virtually no false detections (see fig. 5.10 (d) and fig. 5.11 (b)).

Applying wavelet-based edge detection to the same wideband spectrum as used in fig. 5.7 leads to fewer subband signals being detected when compared to the detection results of the WED (see fig. 5.12). Even moderately strong signals are not reliably detected. This is due to the fact that wavelet edge detection is more susceptible to low SNR when compared to standalone wideband energy detection. Nonetheless, when noise levels are comparatively low and detection latencies tolerable, wavelet edge detection combined with wideband energy detection offers resource-friendly and high-resolution detection results.

5.4. Eigenvalue Detection

In contrast to energy detection, eigenvalue detection does not rely on information about noise power levels. Thus, it can be reliably employed in environments with unknown and changing background noise (CT-02). Figure 5.13 (a) shows the sample covariance matrix $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}}$ of the eigenvalue detector (for $L = 20$) when applied to $N = 10 \times 10^3$ CWGN samples with $\sigma^2_{w,\mathrm{dB}} = 0$ ($\sigma^2_w = 1$). Because the

[1]The wavelet edge detector returns a linked list of signal objects for every detection step. At time 7 s in fig. 5.11, the edge detector returned a linked list with a single signal object with the following attributes: center frequency $f_c \approx -5.86\,\mathrm{kHz}$, bandwidth $B \approx 1.56 \times 10^3\,\mathrm{kHz}$, energy $= 1.58 \times 10^3$. At time 0.09 s in fig. 5.10, the detector returned a linked list with 3 entries. For more information, see [115].

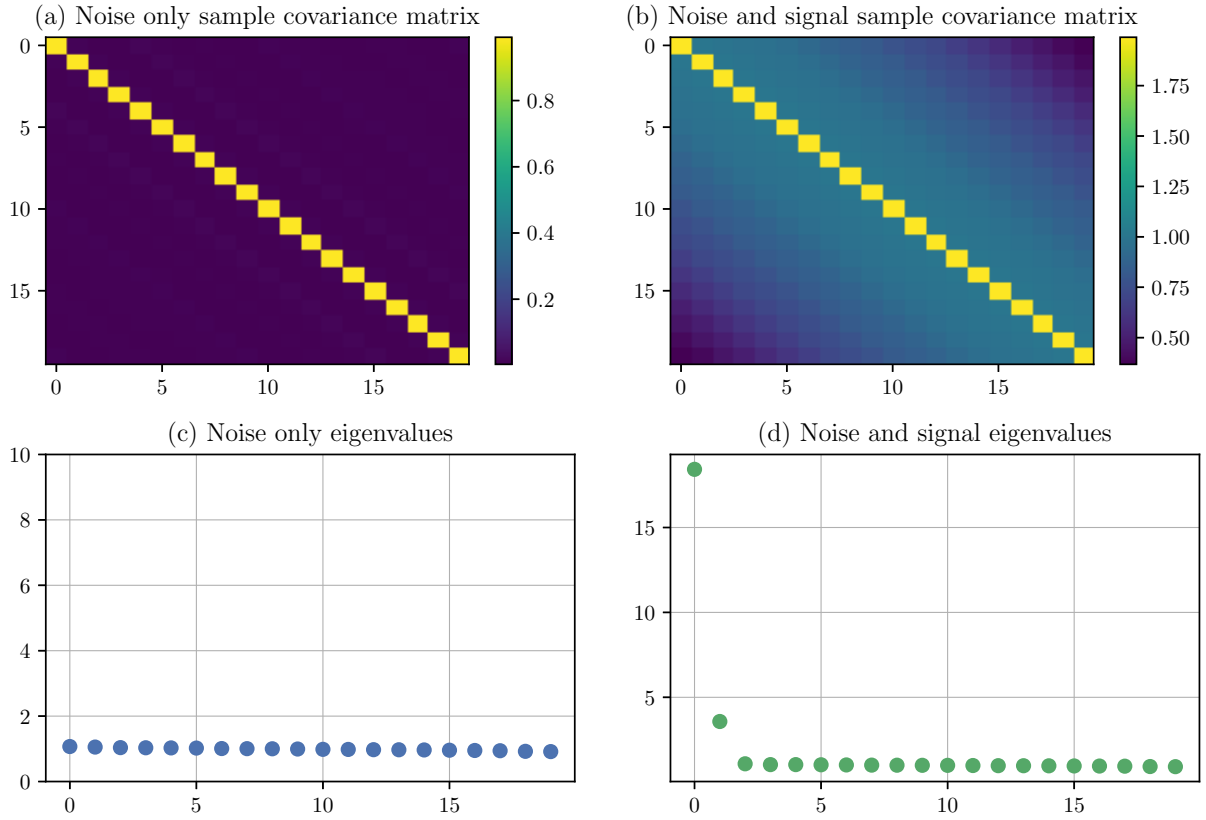

Figure 5.13.: Sample covariance matrices and eigenvalue distributions of the eigenvalue detector. (a) and (c) visualize the sample covariance matrix and eigenvalue distribution of the eigenvalue detector (with $L = 20$) when applied to only CWGN. When a signal is introduced, the covariance matrix and eigenvalue distribution transform to (b) and (d).

noise is uncorrelated, one can clearly see how the matrix has *ones* on the main diagonal and *zeros* elsewhere (refer to section 3.1.2 for information). All 20 eigenvalues are thus approximately equal to *one* (see fig. 5.13 (c)). When a signal with $\sigma^2_{s,\mathrm{dB}} = 0$ is introduced, the sample covariance matrix transforms to fig. 5.13 (b), where the non-diagonal entries are now also *non-zero*. This results in the eigenvalues being not evenly distributed anymore (see fig. 5.13 (d)). Comparing the largest eigenvalue $\nu_{\max}$ to the smallest eigenvalue $\nu_{\min}$ allows the eigenvalue detector to detect the presence of signals, independently of noise power levels (see eq. (3.44)).

5.4.1. Whitening

As already discussed in section 3.2.3, the eigenvalue detector, while being robust against unknown or changing noise power levels, is very sensitive to correlation introduced by signal preprocessing and filtering. As all RF systems employ some form of preprocessing, it is almost always necessary to introduce whitening. The characteristics of the induced correlations are described by the whitening matrix $\mathbf{Q}$, or its inverse $\mathbf{Q}^{-1}$. Using the in eq. (3.72) presented relations, it is possible to eliminate the correlations and whiten the noise.

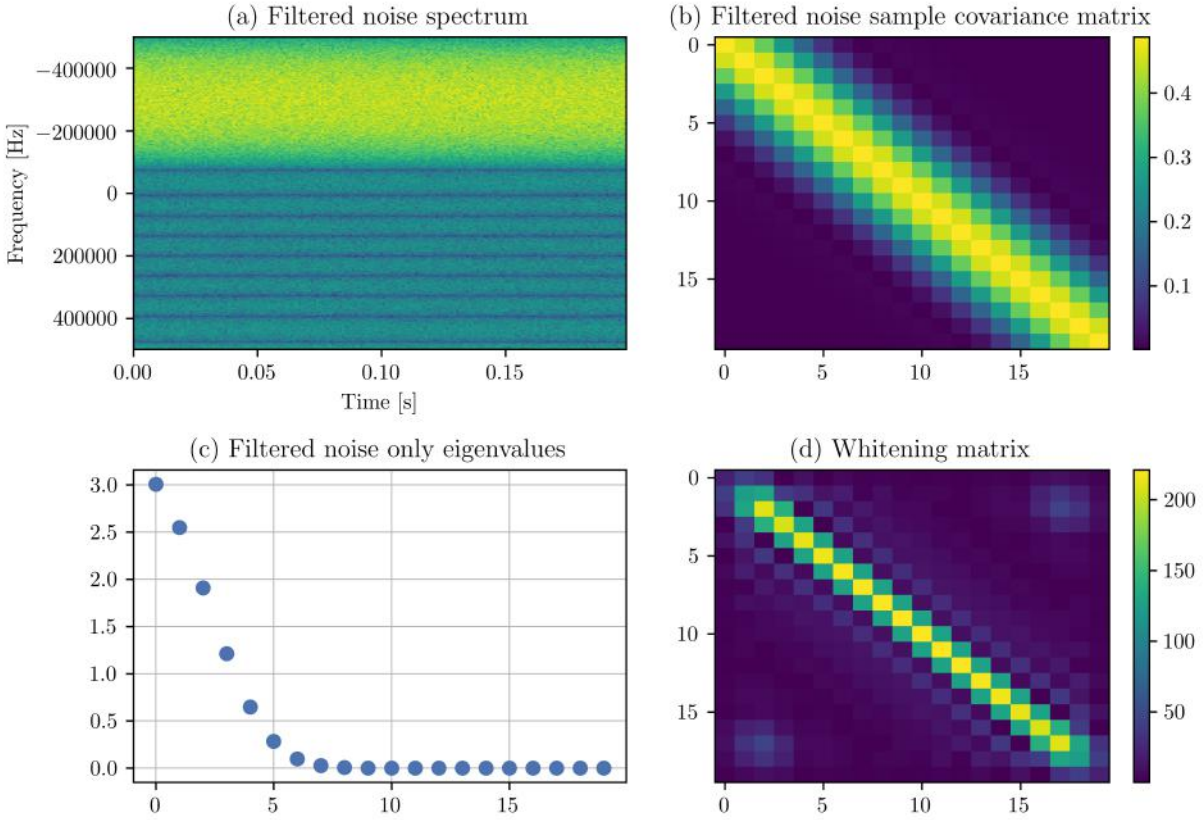

Figure 5.14.: Figure (a) shows the filtered wideband spectrum using a 16-tap FIR bandpass filter. This results in unwanted correlation as depicted in the sample covariance matrix (b) and eigenvalue distribution (c). The whitening matrix $\mathbf{Q}^{-1}$ in (d) mitigates the impact of the FIR filter by *whitening* the correlated noise.

The wideband eigenvalue detector requires a tunable bandpass filter (see section 3.2.3). The here presented detector employs a 16-tap FIR bandpass filter. The resulting wideband spectrum, filtered and noise only ($\sigma^2_{w,\mathrm{dB}} = 5.0$, with filter center frequency $f_{f,c} = -300\,\mathrm{kHz}$ and filter bandwidth $f_{f,B} = 200\,\mathrm{kHz}$), is depicted in fig. 5.14 (a). While one can see that the filter suppresses noise components outside its passband, it is also clearly visible how the filter does not attenuate all unwanted frequencies completely. The regions just outside the intended passband are only attenuated but not fully rejected. Employing larger FIR filters with more *taps* allows for sharper filter edges at the expense of higher phase shift and implementation cost. As can be seen in its sample covariance matrix (fig. 5.14 (b)) and eigenvalue distribution (fig. 5.14 (c)), the filtered noise is highly correlated. Applying the eigenvalue detector without any form of whitening would not result in good sensing performance. The inverse whitening matrix $\mathbf{Q}^{-1}$, visualized in fig. 5.14 (d), can be used to remove the induced correlations. The sample covariance matrix and its whitening counterpart must be at least as large as the correlation time-delay of the filtering or preprocessing (here 16, because the FIR filter has 16 taps).

To effectively suppress signals in neighboring subbands, the in fig. 5.15 presented eigenvalue detector simulation employed a 64-tap FIR bandpass filter. Because larger filters with even more correlation time-delay are a common occurrence (IIR filters are an extreme example), it is not practically feasible to rely on the whitening matrix alone to remove the effects of filtering and preprocessing. More

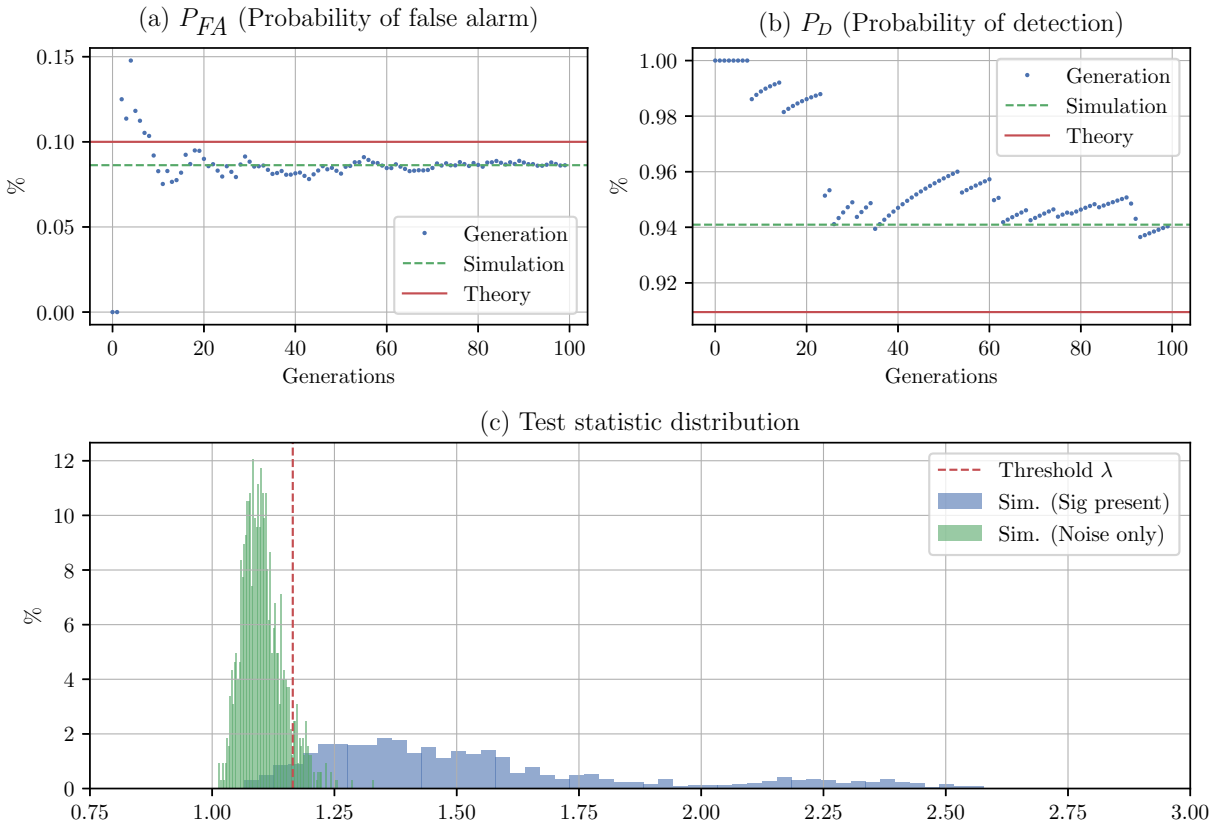

Figure 5.15.: Wideband eigenvalue detector simulation with a 64-tap FIR bandpass filter and 3 dB noise uncertainty. The detector used a $L = 10$ covariance and whitening matrix by employing the novel correlation error correction method.

advanced measures are required. Thus, this thesis introduces an additional pretraining step, in which the correlation-induced detection error is learned through several simulation trials with randomized wideband noise and signal samples. Similarly to the whitening matrix computation, the pretraining can be done in advance and does not influence the detectors real-time performance. The learning is done not at the covariance matrix, but at the detector level. This novel whitening technique allowed the in fig. 5.15 presented simulation to only employ a $L = 10$ covariance and whitening matrix. Because the eigenvalue detector's complexity grows with $\mathcal{O}(LN + L^3)$ (see section 3.1.2), this new method significantly reduces the required computation time when compared to a $L = 64$ covariance matrix; thereby reducing the implementation cost and bringing the eigenvalue detector with whitening in the realm of embedded applications (CT-06, CT-05). Only very basic simulations have been performed thus far, as there is a paucity of literature on decorrelation in the realm of eigenvalue spectrum sensing. More work on actual implementations, using real data samples, needs to be conducted (see section 6.2).

The simulation results shown in fig. 5.15 used the following parameters: $\sigma^2_{w,\mathrm{dB}} = 17.5$ with 3 dB of noise uncertainty ($\sigma^2_{w,\mathrm{dB}} \sim \mathcal{N}(17.5, 3)$), $\sigma^2_{s,\mathrm{dB}} = 1$ (average $\mathrm{SNR}_{\mathrm{dB}} = -16.5$), signal length 0.01 s ($N = 10 \times 10^3$)). The covariance matrix size was set to $L = 10$. All signal parameters were identical to the signal presented in fig. 4.1 (b), with the signal's center frequency set to $f_c = -300$ kHz. The

64-tap FIR bandpass filter center frequency was tuned to $f_{f,c} = -300\,\text{kHz}$ with a filter bandwidth of $f_{f,B} = 200\,\text{kHz}$. The probability of false alarm was set to $P_{FA} = 0.1$. This resulted in the following analytically derived values: $\lambda = 1.1648$ and $P_D = 0.9095$ (using the equations from section 3.1.2).

Training the whitening matrix over 256 steps took 2 minutes and 16 seconds. The actual simulation, using 100 generations and 20 iterations (total iteration $= 2 \times 10^3$) took about 25 seconds. It computed the following numerical values: $P_{FA} = 0.0863$ and $P_D = 0.9409$ (see fig. 5.15 (a) and (b)). Because their derivation is very complex, no theoretical distributions are presented in fig. 5.15 (c). The higher than theoretically computed P_{FA} and P_{D} values are likely due to some remaining correlation not completely being removed by the whitening procedure. However, through more simulations (which for reasons of space are not shown here, see [115]), it could be shown that the eigenvalue detector statistics (eq. (3.49), eq. (3.50) and eq. (3.51)) are not very accurate in general. They require manual tuning of some parameters, which is far from desirable. Additionally, the statistics are based on limiting approximations. Thus, it is reasonable to assume that they are somewhat inaccurate for a finite number of samples. Nonetheless, the presented simulations show that, in contrast to energy detection, eigenvalue detection can operate in uncertain noise environments. Moreover, the here discussed methods of removing unwanted correlations are directly applicable to eigenvalue noise estimation.

5.5. Noise Estimation

This section presents the simulation results of the in section 3.3 introduced noise estimation methods. It will start with noise estimation from signal free-bands (see section 3.3.1), continue with eigenvalue noise estimation (see section 3.3.2), and end by comparing various noise estimation methods. Because wideband energy detection relies on the precise knowledge of background noise power levels, the presented results are of paramount importance to the final WED (see section 5.6).

5.5.1. Free-Band Noise Estimation

As presented in section 3.3.1, free-band noise estimation uses signal-free subbands to extract noise information from data samples. Using eq. (3.81), it can estimate the noise power $\hat{\sigma}_w^2$. The in fig. 5.16 presented WED simulation was conducted in a similar environment as the simulation shown in fig. 5.5. However, to speed up and simplify the simulation, only four subbands were used ($L = 4$). The noise $\sigma^2_{w,\text{dB}} = 12.5$ and signal $\sigma^2_{s,\text{dB}} = 1$ power levels were adjusted accordingly. The noise uncertainty was set to $3\,\text{dB}$ ($\sigma^2_{w,\text{dB}} \sim \mathcal{N}(12.5, 3)$) and the noise estimation buffer length, used to smooth noise estimation results, was set to 10; meaning that the last 10 noise estimation results were used to produce the current noise estimate. The simulation, using the following analytically computed values: $\lambda = 1.0840$, $P_D = 0.9208$ (with $P_{FA} = 0.1$), took about $15.9\,\text{s}$ to complete. Previously conducted simulations showed that the (wideband) ED itself cannot handle noise uncertainties (see fig. 5.3, where only $1\,\text{dB}$ of noise uncertainty was present). However, the simulation results shown in fig. 5.16, using the in section 3.3.3 presented statistics, clearly indicate that the WED using noise estimation can operate in noise uncertain environments. The analytically computed values accurately approximate the numerical values: $P_{FA} = 0.1025$ and $P_D = 0.9205$ (see fig. 5.16 (a) (b)).

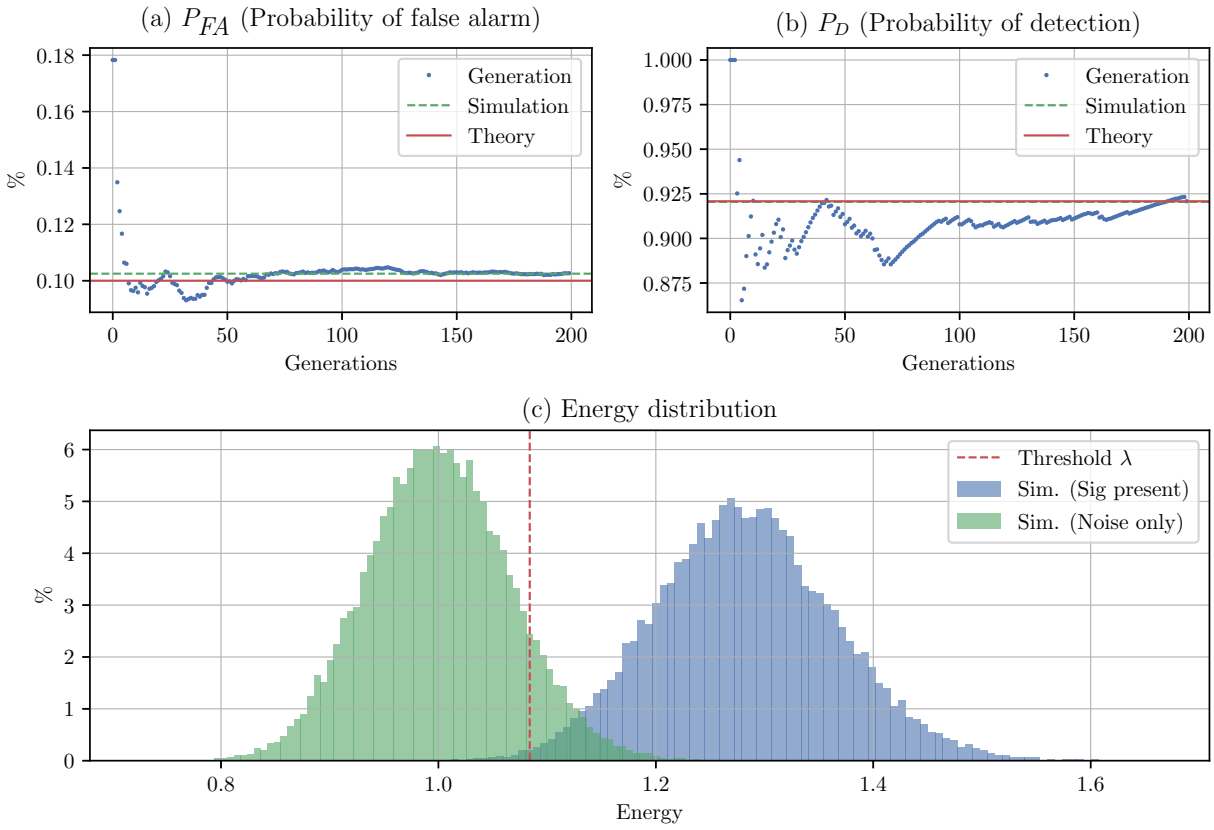

Figure 5.16.: Wideband energy detection with noise estimation from signal free-band subband at $\mathrm{SNR}_{\mathrm{dB}} = -11.5$ and 3 dB noise uncertainty using a 10× noise estimation buffer.

As already discussed in section 3.3.1, free-band noise estimation relies on the fact that the subband used for noise estimation is actually noise-free. While previously discussed methods (e.g., wideband eigenvalue detection) can be used to ensure the absence of signals in the subband used for noise estimation, the in fig. 5.16 presented simulation used prior knowledge of signal-free subbands. The energy detector was simulated in band $l = 1$, while noise estimation was conducted in a signal-free subband $l = 3$. This simplified the simulation, as no suitable dynamic free-band search and selection algorithm was available.

The here presented noise estimator and ED combination is the "de facto standard" in much of the available spectrum sensing literature [26, 65, 94, 101, 102, 109]. Most authors assume that one has prior knowledge of signal-free subbands, which, especially for space-related applications, is a false premise (see section 1.3.4).

5.5.2. Eigenvalue Noise Estimation

Compared to free-band noise estimation, eigenvalue-based noise estimation does not require any information about signal-free subbands. It uses the eigenvalue decomposition of the sample covariance matrix to extract noise information from wideband signals (see section 3.3.2), making it more robust and easier to use than free-band noise estimation. The only prior mention of this type of

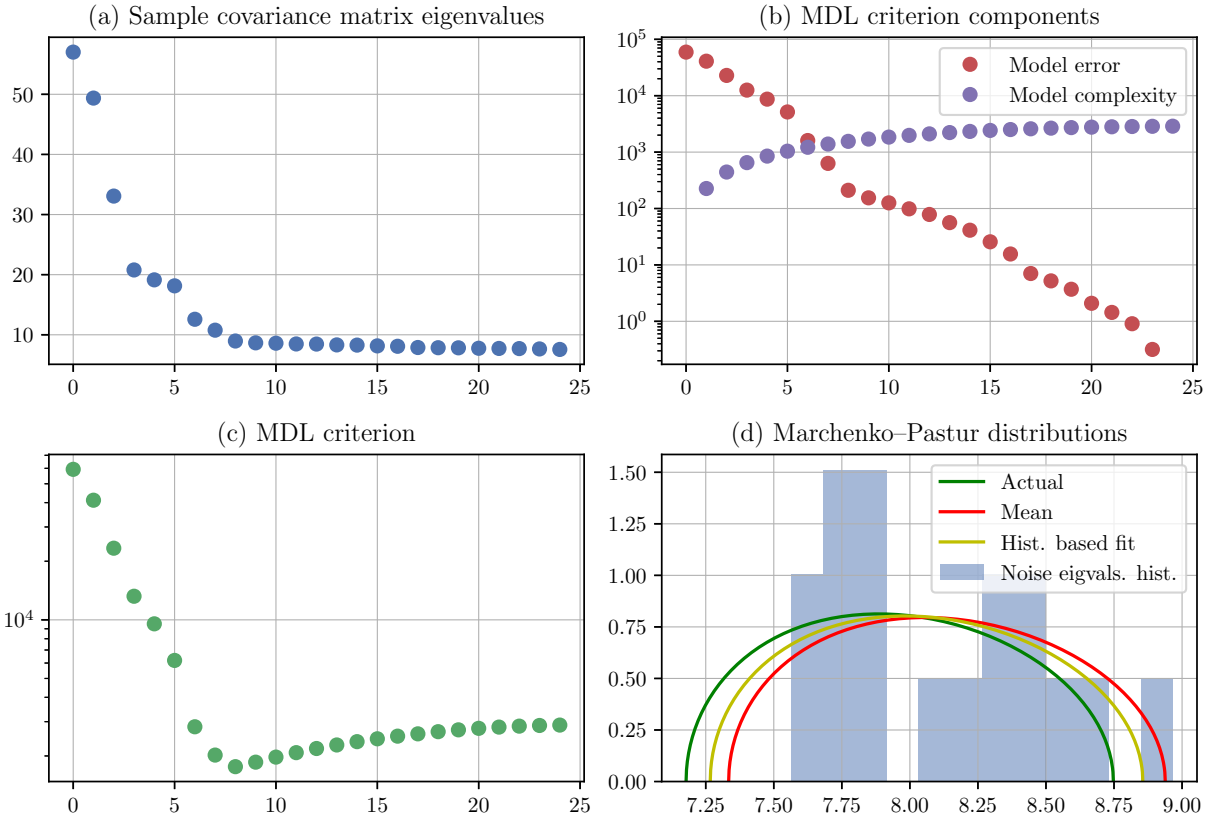

Figure 5.17.: Visulization of eigenvalue-based noise estimation: applying the MDL criterion on the eigenvalues from (a) results in (c). (b) presents the MDL criterion equation components separately. Using the estimated noise-only eigenvalues, one can fit the Marchenko–Pastur distribution to the eigenvalue histogram and obtain the final noise estimate (d).

noise estimation in the realm of spectrum sensing appears to be in [123]. The simulations conducted by the authors of [123] "only" employed a *narrowband* ED. Additionally, they did not give any analytical description of its performance. The remainder of this chapter will thus concern itself with eigenvalue noise estimation and its application to wideband energy detection.

The wideband signal used for the visualizations in fig. 5.17 consisted of four narrowband signals, each with $\sigma^2_{s,\mathrm{dB}} = 1$, and additive noise at $\sigma^2_{w,\mathrm{dB}} = 9$. The MDL criterion was applied to the in fig. 5.17 (a) depicted sorted eigenvalue distribution (of the sample covariance matrix). The *model error* and *model complexity* with increasing model size $\hat{M}$ are plotted in fig. 5.17 (b); while the overall MDL criterion is shown in fig. 5.17 (c) (see eq. (3.87)). Selecting the minimum value of the MDL criterion, here $\hat{M} = 8$, and discarding all eigenvalues larger than the 8th eigenvalue leaves only the noise-dependent eigenvalues. Using the remaining (noise only) eigenvalues to construct a histogram (also known as EDF (see eq. (3.90)) and fitting the Marchenko–Pastur distribution (see eq. (3.93)) results in fig. 5.17 (d). The Marchenko–Pastur distribution of the actual (correct) noise power and the distribution using the mean of the eigenvalues, are also shown in fig. 5.17 (d). One can see that in the here presented environment, the noise estimation using the histogram-based fit outperforms the eigenvalue average.

5.5.3. Noise Estimation Comparison

This section provides a quantitative comparison of different noise estimators and their estimation errors.

The noise estimation simulation for all six graphs in fig. 5.18 was conducted using the SpecSens simulation framework, running 5 iterations over 50 generations (250 total iterations) for every data point in each of the six graphs. If not stated otherwise, the following parameters were used: $\sigma^2_{w,\mathrm{dB}} = 10$ (with 10 dB of uncertainty, $\sigma^2_{w,\mathrm{dB}} \sim \mathcal{N}(10, 10)$), $\sigma^2_{s,\mathrm{dB}} = 1$, $N = 8192$, $L = 20$. Five narrowband signals (see fig. 4.1) occupied the wideband; resulting in 25% spectral occupancy. All estimation results were compared to the actual noise power value using the relative error in dB. The total simulation time for all six simulations was 1 hour and 26 minutes.

The blue curves in fig. 5.18 show the theoretically achievable estimation performance when using an optimal noise estimator. Generally, no unbiased, real-world estimator achieves better estimation performance in terms of the lowest possible *mean squared error* (MSE). The green curves show the free-band noise estimator. The red and the purple lines represent the average noise eigenvalue and histogram-based Marchenko–Pastur eigenvalue fitted noise estimate. Additionally, two novel Marchenko–Pastur eigenvalue noise estimators are depicted: the yellow curves show a kernel density estimator-based (KDE) Marchenko–Pastur eigenvalue noise estimator (using Gaussian kernels), while the light-blue curves represent a maximum likelihood estimate (MLE) of the Marchenko–Pastur distribution over the noise-only eigenvalues.

In fig. 5.18 (a) the length of the available data samples for noise estimation was varied using powers of two: $N = 2^k,\ k = 9, \ldots, 19$. This resulted in data vector lengths of 512 to $\approx 52410 \times 10^3$. The graphs show: the longer the sample data vector, the better the noise estimate, regardless of the estimation method used. While all methods performed very similarly on average, free-band noise estimation consistently performed worst. In fig. 5.18 (b) the spectrum occupancy was altered. Again, all eigenvalue-based noise estimators performed comparably; the estimation error increased with increasing spectrum occupancy. When approaching occupancy values near 100%, the estimation error rapidly increased. Until about 45% occupancy the free-band noise estimator was able to "keep up" with the eigenvalue-based estimators. However, at around 45% occupancy, the first signal appeared in its "free" subband, rendering it completely useless. Varying the SNR does not seem to impact any noise estimator's performance, except for the free-band noise estimator (see fig. 5.18 (c)). With SNR values above 0 dB, its performance significantly degrades. This is likely caused by an increasing signal power leakage into its "free" band, coming from neighboring signal subbands. The remaining three graphs, fig. 5.18 (d) (e) (f), show the detection errors over varying sample covariance matrix sizes under different spectrum occupancy values of 15%, 55%, and 85%, respectively. Examining fig. 5.18 (d), one can conclude that the covariance matrix size does not impact the noise estimator performance under low occupancy values. When increasing the occupancy to 55%, small covariance matrix sizes $L < 10$ perform poorly (see fig. 5.18 (e)). The performance difference between $L = 10$ and $L = 50$, however, is not noticeable. Even under high spectrum occupancy (see fig. 5.18 (f)), no real benefit from using covariance matrix sizes of more than $L = 20 \ldots 40$ can be inferred.

The presented results provide the first quantitative comparison of noise estimation methods in the context of wideband energy detection-based spectrum sensing. They suggest that the *average* noise eigenvalue estimation method suffices for the majority of noise estimation applications. With the exception of very small covariance sizes (below $L = 10$) under relatively high spectral occupancy, no

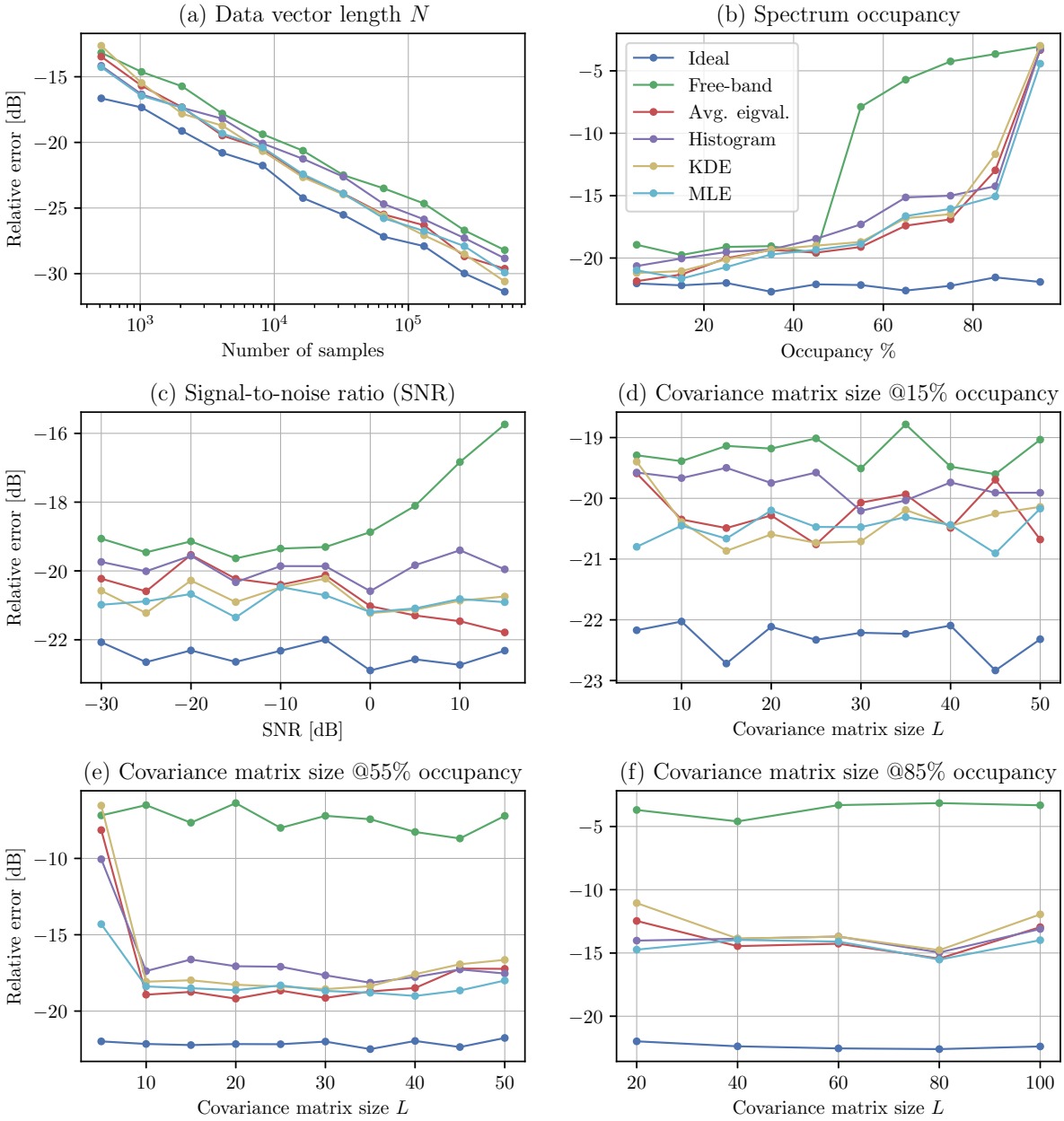

Figure 5.18.: Comparison of noise estimation methods under varying parameters, such as number of samples N (a), spectrum occupancy (b), SNR (c) and covariance matrix size L (d) (e) (f).

real advantage justifying the extra step of eigenvalue distribution fitting to the Marchenko–Pastur distribution could be found. In the particular case of small covariance sizes and high spectral occupancy, the MLE estimator emerged as the most promising noise estimation method. However, since covariance matrix sizes of $L = 10 \ldots 20$ can be realistically implemented and because spectral occupancy values above 50% are very rarely encountered, *average noise eigenvalue estimation* emerges as the premier noise estimation method for real-world applications in resource-constrained environments, like SALSAT (CT-06).

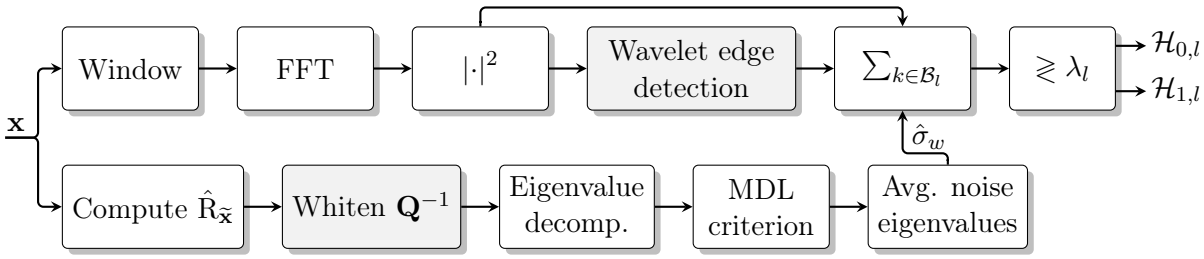

Figure 5.19.: Simplified block diagram of the final WED with noise estimation using average noise eigenvalues. The gray blocks (wavelet edge detection and noise whitening) are optional.

5.6. Wideband Energy Detection with Eigenvalue Noise Estimation

This section will detail the simulation results of the *final* detector (see fig. 5.19), combining wideband energy detection with average eigenvalue noise estimation to produce reliable detection results in noise uncertain-environments. It employs the WED (and its statistics) from section 3.3.3 and combines it with the noise estimation techniques presented in section 3.3.2 and section 5.5.3. Under reasonable SNR, one can additionally use the wavelet edge detector to detect subbands dynamically; otherwise, a prespecified number of uniform subbands must be selected. If the from filtering and preprocessing introduced correlation leads to poor detection performance, one can employ the above presented whitening techniques (see section 5.4.1).

For the in fig. 5.20 depicted simulation results, neither wavelet edge detection nor whitening was used. The number of subbands was set to four ($L = 4$), generations = 200, iterations = 300 (total iterations = 60×10^3), signal length = 0.001 024 s (rectangular window with $N = 1024$), avg. $\mathrm{SNR}_{\mathrm{dB}} = -10$ ($\sigma^2_{w,\mathrm{dB}} = 10$ with 3 dB of uncertainty, $\sigma^2_{w,\mathrm{dB}} \sim \mathcal{N}(10, 3)$, and $\sigma^2_{s,\mathrm{dB}} = 0$), $P_{FA} = 0.1$. The signal parameters were similar to the ones used in fig. 5.16. The average eigenvalue noise estimator employed a $L = 20$ sample covariance matrix and a $10\times$ estimation history buffer. This resulted in the following parameters which were analytically computed before starting the simulation: $\lambda = 1.0811$, $P_D = 0.9711$. The simulation took 20.1 s and calculated the following numerical values: $P_{FA} = 0.0934$, $P_D = 0.9534$. The results presented in fig. 5.20 show that the noise estimator is able to accurately approximate the actual noise power levels, thus yielding good detection performance. The wideband detector is able operate independently of noise power levels and without the knowledge of signal-free subbands. The slightly better than expected sensing performance (in terms of lower P_{FA}) is likely caused by the more precise noise estimation of the eigenvalue noise estimator when compared to the free-band estimator. The in section 3.3.3 depicted statistics were derived for asymptotically Gaussian estimators (such as the free-band noise estimator). Through the covariance matrix eigenvalue decomposition, the eigenvalue noise estimator is able to more effectively extract the underlying noise power level and thus has a slightly different distribution and performance (see section 5.5.3).

5.7. Summary

This chapter presented the implementation and simulation of the in chapter 3 introduced spectrum sensing methods by utilizing the novel simulation framework SpecSens. With the help of MC simulations, it was shown that the proposed chi-square and CLT statics are generally able to

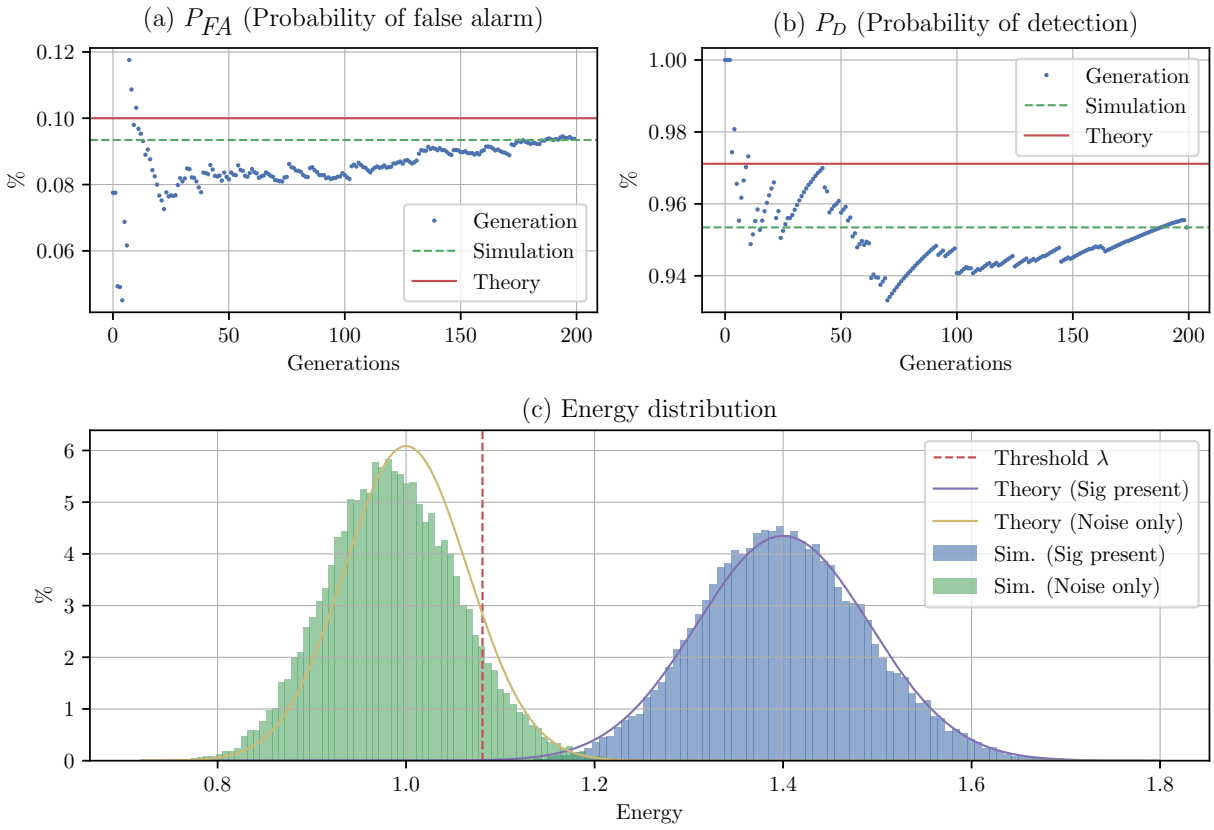

Figure 5.20.: Wideband energy detection with noise estimation using the average noise eigenvalues under 3 dB of noise uncertainty.

describe the ED's operation. The concept of energy detection was subsequently applied to wideband detection and simulated. The WED's performance could be accurately modeled with the help of analytically derived equations, resulting in reliable detection of wideband and Doppler-shifted signals. Windowing functions, with the exception of the rectangular window, resulted in very unpredictable sensing behavior. This observation appears novel, as it is not yet detailed in spectrum sensing-related literature. Wavelet edge detection was able to find spectral edges and subbands dynamically. While it efficiently uses the sparsity in detection results, it is not applicable to low SNR scenarios. In contrast to energy detection, eigenvalue detection does not rely on the precise knowledge of background noise power levels. It can reliably detect weak signals, even in unknown and changing noise environments. By employing novel whitening methods, undesirable correlations caused by preprocessing and filtering could be removed.

In noise uncertain environments the WED's performance deviates significantly from the analytically derived metrics. With knowledge of noise-free subbands, free-band noise estimation mitigated the impact of uncertain noise levels on the WED. Because of the shortcomings of free-band noise estimation, simpler and more robust noise estimation techniques, relying on the eigenvalue decomposition of the sample covariance matrix, were introduced. Simulations showed that no eigenvalue fitting to the Marchenko–Pastur distribution is necessary; taking the average of the noise eigenvalues suffices in most noise estimation scenarios. Finally, wideband energy detection was combined with average eigenvalue noise estimation. Simulations showed that it produces reliable detection results, even in noise uncertain environments.

6. Conclusions and Future Work

6.1. Requirements Verification

The wideband energy detector presented in section 5.6 combines research results from recent literature with novel methods developed in this thesis to reliably (CT-03) produce detection results over a wide range of frequencies simultaneously (MR-02). It can operate without any knowledge of the signal trying to detect (MR-01). Through its eigenvalue noise estimation algorithm, the detector is able to operate in environments with uncertain and changing noise power levels (CT-02). Using analytically derived performance metrics, desired operating characteristics can be dynamically adjusted (MR-03). Under the constraint of blind sensing, the WED provides an excellent trade-off between sensing performance in low SNR scenarios (CT-01) and implementation complexities (CT-06), while at the same time requiring relatively few data samples to produce detection results (CT-05). It operates using a single antenna (IR-02) and I/Q data stream (IR-01) without requiring any external information (CT-07).

While the general performance and function of the WED could be verified, no *target simulations* using real data were possible (see section 1.4 and section 1.3.1). Thus, the here presented verification is based solely on simulated data samples, resulting in only preliminary requirement verification.

6.2. Future Work

As already indicated in chapter 1, the presented work is just a first step towards a fully functional spectrum sensing implementation. It provides the necessary theoretical foundation on which further simulations and tests (using real I/Q data) need to be conducted. Simulated data can by no means replace thorough testing in real, complex, and heterogeneous spectral environments. To further verify the in section 1.4 presented requirements, prerecorded data samples (coming from SALSAT) could be injected into the SpecSens simulation framework and its MC simulation infrastructure. New signal-sources, replaying the labeled I/Q data samples, could augment the existing signal generators. Thanks to Python's extensive signal processing support and SpecSens's modular architecture, no external tools or systems would be required.

Python is an interpreted programming language that is generally not well suited for embedded, resource-constrained, and real-time environments. While some of its packages (e.g., NumPy) offer very efficient precompiled functions, Python can, for the most part, not compete with compiled, low-level programming languages, such as C or C++. They offer smaller memory footprints and better embedded software tool support, making them far more suitable for platforms like SALSAT. Thus, the next step would be to implement the presented algorithms in one of the mentioned languages. Ideally, one would employ specialized software packages (written in C/C++), which provide

Req.-Number	Addressed / Verified in	Comment
MR-01	section 3.1.1, section 5.1	The detector is capable of detecting all signal types.
MR-02	section 3.2.2, section 5.2	The detector is capable of detecting subband signals in wideband data.
MR-03	section 3.3.3	Through analytical equations, desired performance metrics can be dynamically adjusted.
CT-01	section 5.6	The detector is capable of operating in low SNR environments.
CT-02	section 3.3.2, section 5.6	Using eigenvalue noise estimation the detector is capable of operating under unknown and varying background noise.
CT-03	section 5.2, section 5.6	The detector is capable of reliably detecting signals in heterogeneous environments.
CT-04	section 3.2.2, section 5.2	The detectors complexity is relatively low, allowing for high-speed operation.
CT-05	section 3.2.2, section 5.2	Detection latency is relatively low, reducing the time between actual spectrum occupancy change and detection.
CT-06	section 3.2.2, section 5.2	The detectors complexity is relatively low, allowing for operation in resource constrained environments.
CT-07	section 3.3.3, section 5.6	The detector can operate without the need for networking or cooperation.
IR-01	section 3.1.1	The detector and the noise estimator operate using only I/Q data.
IR-02	section 3.1.1, section 3.3.2	The detector and the noise estimator require a single antenna for operation.

Table 6.1.: Verification of formal requirements defined in table 1.2.

high-performance signal processing functions (see appendix A). One can also utilize specialized vector instructions available on the respective target platform to increase computational performance further. SALSAT, through its NXP i.MX 7 ARM processor (see 1.3.1) provides NEON SIMD vector instructions, running up to 1 GHz. These specialized instruction set extensions increase the DSP processing capability of high-performance software while offering the low-power consumption required by resource-constrained applications.

Recent advances in hardware accelerators, fueled by machine learning and neural network research, lead to the resurrection of *systolic arrays* [124]. Systolic arrays expand the concept of vectorization into two dimensions, enabling high-performance linear algebra computations, such as matrix multiplication or solving systems of linear equations. FPGAs can achieve superb processing performance using systolic array architectures, even in real-time environments [125]. The FFT algorithm used throughout this thesis is already implemented as a hardware-accelerator onboard SALSAT's Intel Cyclone IV FPGA (see section 1.3.1) [92]. Similar accelerators, using the above-mentioned systolic array architecture, could be devised.

Eigenvalue noise estimation, being a relatively computationally expensive algorithm, is an excellent target for hardware acceleration. Its three main steps are the computation of the covariance matrix, multiplication with a whitening matrix, and eigenvalue decomposition (see fig. 3.2). Computing the covariance matrix requires the evaluation of many auto-correlations. Seminal work on the fast computation of auto-correlations using hardware acceleration was conducted in [126, 127] and [128]. The general task of matrix multiplication, needed for whitening, is further detailed in [129]. Finally, the eigenvalue decomposition of a square matrix can be generalized to any arbitrary matrix with *singular value decomposition* (SVD) [86, 100]. SVD implementations for FPGAs are investigated in [130], with some work on related decompositions presented in [131].

6.3. Conclusions

This work introduced a novel spectrum sensing method that combines wideband energy detection with eigenvalue-based noise estimation. Its properties make it a prime candidate for spectrum sensing on small and resource constraint satellites, such as SALSAT. It is able to operate in heterogeneous and complex environments. Even under low signal-to-noise ratios, the algorithm can reliably detect subband signals in wideband data and report on the underlying radio spectrum's occupancy.

SpecSens, a new simulation framework for spectrum sensing methods, was presented. Thanks to its modular design and its tight integration with the Python programming language, SpecSens lends itself to rapid algorithm prototyping and development. Through the parallel Monte Carlo simulation infrastructure, SpecSens can efficiently run simulations, giving quick and direct feedback to algorithm designers.

Because the presented work relies solely on simulated data it can only be seen as a preliminary step towards a functional implementation. While simulating data simplifies and streamlines algorithm development, it does not replace simulations using *real* data. More work on verification using RF samples from space is necessary. Additionally, all simulated algorithms were implemented in Python, making them not applicable to embedded and real-time operation. They need to be translated to high-performance and low-latency implementations, possibly using support from specialized hardware accelerators.

A. Mathematical Background

A.1. Probability Distributions

Performance evaluation of signal detection algorithms requires the ability to determine the probability distributions of the data samples and the subsequent test-statistic. Wherever possible, they are modeled using standard probability density functions. This section provides the definitions of the most relevant distributions used throughout this thesis.

A.1.1. Gaussian

The *real Gaussian probability density function* (PDF) for a scalar random variable $x \in \mathbb{R}$ is defined as [132]

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right), \tag{A.1}$$

where $\mathrm{E}[x] = \mu \in \mathbb{R}$ is called the mean and $\mathrm{Var}[x] = \sigma^2 \in \mathbb{R}$ the variance of x. As Gaussian distributions are very prevalent, a shorthand notation exists $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$; where $\sim$ means *distributed as*. Its *cumulative distribution function* (CDF) is denoted by

$$\Phi(x) = \int_{-\infty}^{x} p(t)dt = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t-\mu)^2\right) dt. \tag{A.2}$$

Because no closed-form expression for the Gaussian CDF exists, one needs to resort to tables or numerical solutions. The *complementary cumulative distribution function* (CCDF), also known as the *Q-function*, describes the right-tail probability and is defined as

$$\mathrm{Q}(x) = 1 - \Phi(x) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t-\mu)^2\right). \tag{A.3}$$

To express distributions over many data samples, it is necessary to introduce the *real multivariate Gaussian probability density function.* For a random vector $\mathbf{x} \in \mathbb{R}^N$ with $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ it is characterized by [60]

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{A.4}$$

where $\boldsymbol{\mu} = \mathrm{E}[\mathbf{x}] \in \mathbb{R}^N$ is the mean vector and $\mathbf{C} \in \mathbb{R}^{N\times N}$ is the covariance matrix, which is defined as $\mathbf{C} = \mathrm{E}\left[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\right]$. Since this thesis makes use of complex signals, it is necessary to review the *complex multivariate Gaussian distribution.* The complex sample vector $\mathbf{x} \in \mathbb{C}^N$ can be thought

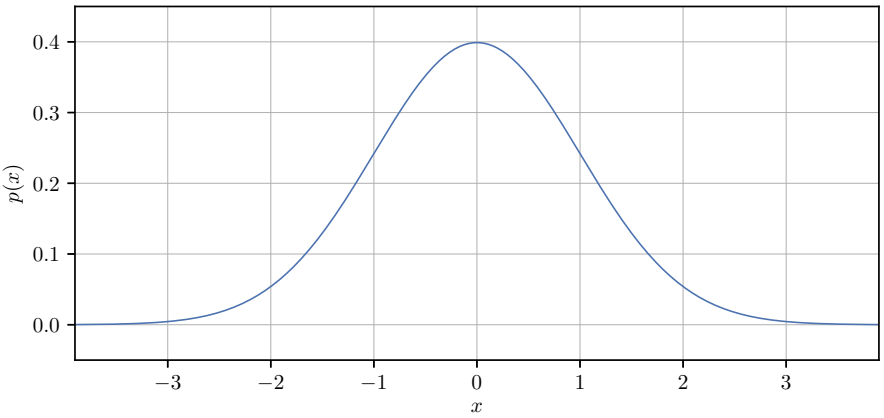

Figure A.1.: PDF of a Gaussian distribution for $x \sim \mathcal{N}(0, 1)$.

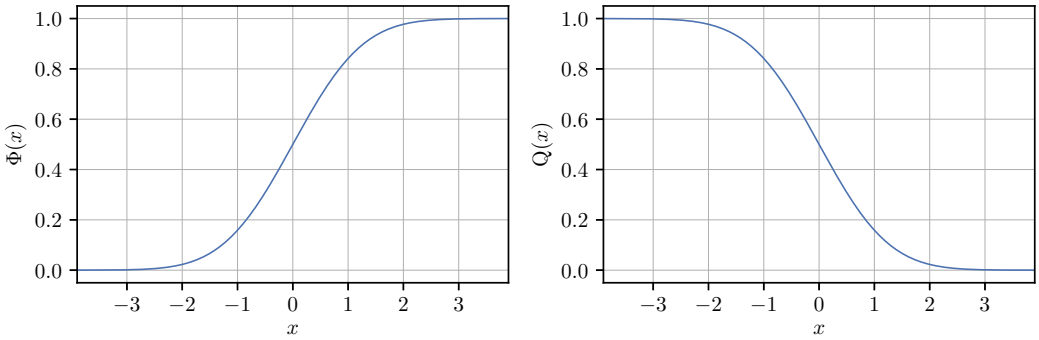

Figure A.2.: CDF and CCDF of a Gaussian distribution for $x \sim \mathcal{N}(0, 1)$.

of as a composition of two real sample vectors, where $\mathbf{x} = \mathbf{x}_r + i\mathbf{x}_i$. The complex mean vector is defined as $\mu = \mathrm{E}[\mathbf{x}] \in \mathbb{C}^N$; the complex covariance matrix $\mathbf{C} \in \mathbb{C}^{N \times N}$ is $\mathbf{C} = \mathrm{E}\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{H}}\right]$[1]. Thus, the PDF becomes

$$p(\mathbf{x}) = \frac{1}{\pi^N \det(\mathbf{C})} \exp\left(-(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{H}} \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \tag{A.5}$$

which can also be written as $\mathbf{x} \sim \mathcal{CN}(\boldsymbol{\mu}, \mathbf{C})$ [60]. Of particular interest to signal processing applications is the so-called *zero-mean circularly symmetric complex Gaussian distribution*, denoted as $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ and where σ^2 is the variance of the signal $\mathbf{x}$.

A.1.2. Chi-Square

A *chi-square* PDF with $N \in \mathbb{N}^+$ degrees of freedom is defined as

$$\chi^2_N(x) = \begin{cases} \frac{1}{2^{\frac{N}{2}}\Gamma(\frac{N}{2})} x^{\frac{N}{2}-1} \exp(-\frac{1}{2}x) & x > 0 \\ 0 & x < 0 \end{cases} \tag{A.6}$$

[1] $\mathbf{A}^{\mathrm{H}} = (\mathbf{A}^{\mathrm{T}})^*$ denotes the conjugate transpose for some complex matrix $\mathbf{A}$.

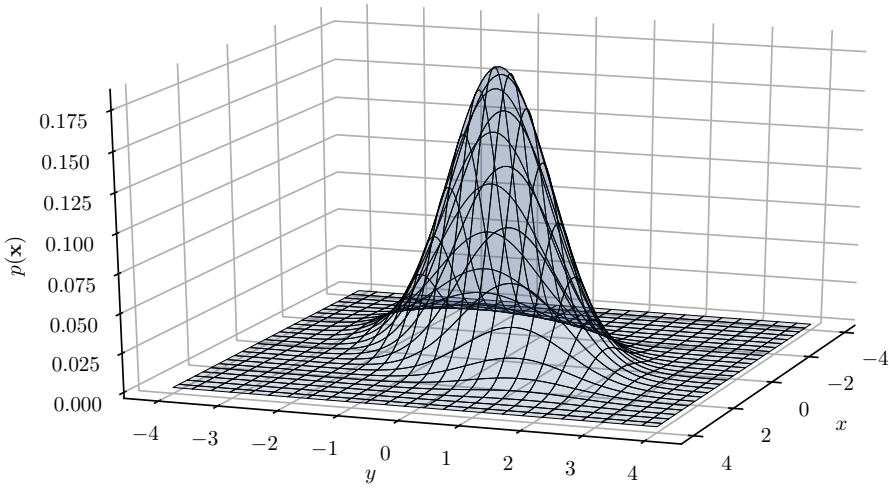

Figure A.3.: Multivariate Gaussian PDF with $N = 2$ and $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ where $\mathbf{C} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$.

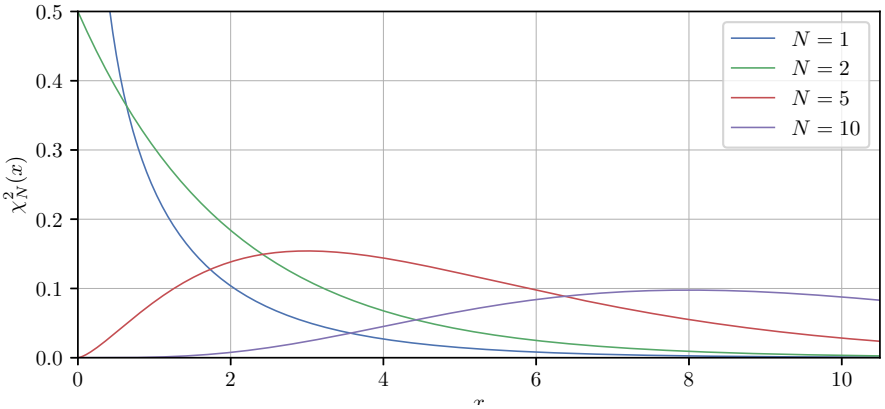

Figure A.4.: Multiple chi-square PDFs with different degrees of freedom.

for $x \in \mathbb{R}$ [60]. The Gamma function $\Gamma(\cdot)$ is defined as

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) dt \tag{A.7}$$

with $\mathrm{Re}(x) > 0 \in \mathbb{R}$. If $x \in \mathbb{N}$, it can be expressed using the factorial

$$\Gamma(x) = (x-1)!. \tag{A.8}$$

The chi-square distribution with N degrees of freedom arises as the sum of N squared independent and identically distributed (IID) random Gaussian variables. That is: if $\mathbf{y} \sim \mathcal{N}(0, \mathbf{I})$, then

$$\sum_{n=0}^{N} y_n^2 \sim \chi_N^2. \tag{A.9}$$

The CDF $F_{\chi^2_N}(\cdot)$ of the chi-square distribution is defined as

$$F_{\chi^2_N}(x) = \int_{-\infty}^{x} \chi^2_N(t)dt = \frac{\gamma(\frac{N}{2}, \frac{x}{2})}{\Gamma(\frac{N}{2})}, \tag{A.10}$$

where $\Gamma(\cdot,\cdot)$ is the upper incomplete Gamma function [60]

$$\Gamma(s,x) = \int_{x}^{\infty} t^{s-1}\exp(-t)dt. \tag{A.11}$$

The lower incomplete Gamma $\gamma(\cdot,\cdot)$ function is defined as

$$\gamma(s,x) = \int_{0}^{x} t^{s-1}\exp(-t)dt. \tag{A.12}$$

The three gamma functions relate to one another as follows

$$\Gamma(s) = \gamma(s,x) + \Gamma(s,x), \tag{A.13}$$

with $x > 0$. When $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ is a complex Gaussian random vector of length N, then the resulting chi-square distribution has $2N$ degrees of freedom

$$\sum_{n=0}^{N} |z_n|^2 \sim \chi^2_{2N}. \tag{A.14}$$

A.1.3. Tracy-Widom

The *Tracy–Widom* distribution describes the limiting cumulative probability distribution (CDF) of the normalized largest eigenvalue of a random Hermitian[2] matrix [61]. For $\beta = 1$ (which is the only relevant β to this thesis), it is defined as

$$F_1(t) = \exp\left(-\frac{1}{2}\int_{t}^{\infty}(q(u) + (u-t)q^2(u))du\right), \tag{A.15}$$

where $q(u)$ arises as the solution to the nonlinear *Painleve* equation of type II

$$q'' = uq(u) + 2q^3(u). \tag{A.16}$$

Because it is generally challenging to evaluate, and neither SciPy nor NumPy includes it, this thesis made use of the *TracyWidom* Python package [114], which in turn uses the interpolation tables from [133] and the asymptotics in [134].

A.1.4. Marchenko–Pastur

The *Marchenko–Pastur* distribution describes the asymptotic behavior of singular values of large rectangular random matrices [107]. Its probability density function is defined as

$$dF^W(\nu) = \frac{\sqrt{(\nu - \sigma_z^2(1-\sqrt{c})^2)(\sigma_z^2(1+\sqrt{c})^2 - \nu)}}{2\pi\sigma_z^2\nu c}d\nu, \tag{A.17}$$

[2] A Hermitian matrix $\mathbf{A}$ is a complex square matrix that is equal to its conjugate transpose. That is $\mathbf{A} = \mathbf{A}^{\mathrm{H}}$.

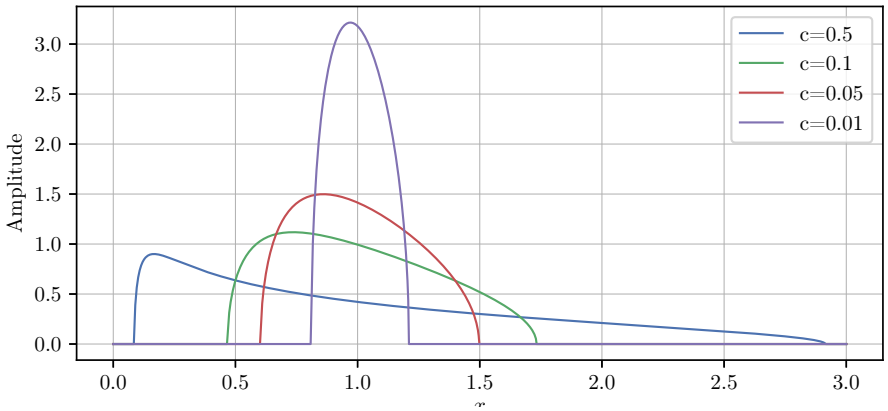

Figure A.5.: Multiple Marchenko–Pastur PDFs under different c.

where

$$\sigma_z^2(1-\sqrt{c})^2 \leq \nu \leq \sigma_z^2(1+\sqrt{c})^2 \tag{A.18}$$

for a Gaussian matrix of size $N \times L$; with $N, L \to \infty$ and $c = L/N$ [103, 104]. As no available Python implementation exists, a custom implementation was added to SpecSens.

A.2. Neyman-Pearson Theorem

This section formulates the proof of the *Neyman-Pearson theorem* using the method of *Lagrange multipliers* and is based on a slightly modified version of the proof presented in [60]. Considering an observation vector $\mathbf{x}$ in observation space R the two possible hypotheses are $\mathcal{H}_0$ and $\mathcal{H}_1$. Using Lagrange multipliers, one can maximize the probability of detection P_D for a given fixed probability of false alarm $P_{FA} = \alpha$. Forming the Lagrangian

$$\begin{aligned}\mathcal{L} &= P_D - \lambda(P_{FA} - \alpha) \\ &= \int_{R_1} p(\mathbf{x} \mid \mathcal{H}_1) d\mathbf{x} - \lambda \left(\int_{R_1} p(\mathbf{x} \mid \mathcal{H}_0) d\mathbf{x} - \alpha \right) \\ &= \int_{R_1} \left(p(\mathbf{x} \mid \mathcal{H}_1) - \lambda p(\mathbf{x} \mid \mathcal{H}_0) \right) d\mathbf{x} + \lambda\alpha. \end{aligned} \tag{A.19}$$

To maximize $\mathcal{L}$, one includes any x in R_1 for which the integrand is positive. Then R_1 is given as

$$R_1 = \left\{ x \in R \mid p(x \mid \mathcal{H}_1) > \lambda p(x \mid \mathcal{H}_0) \right\} \tag{A.20}$$

and one thus decides $\mathcal{H}_1$ when

$$\frac{p(x \mid \mathcal{H}_1)}{p(x \mid \mathcal{H}_0)} > \lambda \tag{A.21}$$

which directly yields the *likelihood-ratio test* (LRT)

$$\Lambda(\mathbf{x}) = \frac{p(x \mid \mathcal{H}_1)}{p(x \mid \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda. \tag{A.22}$$

It is now possible to determine the threshold λ with

$$P_{FA} = \int_{\{\mathbf{x}:\Lambda(\mathbf{x})>\lambda\}} p(\mathbf{x} \mid \mathcal{H}_0) d\mathbf{x} = \int_{\lambda}^{\infty} p(\Lambda(\mathbf{x}) \mid \mathcal{H}_0) d\Lambda = \alpha. \tag{A.23}$$

A.3. Spectral Estimation

The goal of *spectral estimation* (or *spectral density estimation*) is to estimate the spectral density of a signal from a sequence of samples. The spectral density characterizes the frequency content of the signal (over time) [135]. The following section is mostly based on [64, 85, 86, 136].

A.3.1. Fourier Series

The *Fourier series* in engineering contexts is a method to decompose a periodic signal into its frequency components. It is intimately related to the geometry of infinite-dimensional function spaces, also known as *Hilbert spaces*. Hilbert spaces generalize the notion of vector spaces to functions with infinitely many dimensions [64, 86]. The inner product in Euclidean vector spaces for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ is defined as

$$\mathbf{a} \cdot \mathbf{b} = \langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{N} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_N b_N. \tag{A.24}$$

The notion of inner products can be extended to functions. With complex functions $x(t)$ and $y(t)$ defined on domain $t \in L$ [64, 136]

$$\langle x(t), y(t) \rangle = \int_L x(t) y^*(t) dt. \tag{A.25}$$

Just as in Euclidean vector spaces, the inner product in function spaces may be used to project a function into a new coordinate system. Instead of projecting a vector into orthogonal basis vectors, the function is projected into an orthogonal basis of functions spanning the function space [86]. In the case of infinite-dimensional function spaces, the basis one is projecting into is infinite. This observation directly results in the Fourier series for complex T-periodic functions on $[0, T)$

$$\begin{aligned} x(t) &= \sum_{k=-\infty}^{\infty} c_k \cdot e^{i\frac{2\pi k t}{T}} \\ &= \sum_{k=-\infty}^{\infty} (\alpha_k + i\beta_k)(\cos(\frac{2\pi k t}{T}) + i \sin(\frac{2\pi k t}{T})), \end{aligned} \tag{A.26}$$

where $c_k = \alpha_k + i\beta_k$ and $e^{i\gamma} = \cos(\gamma) + i \sin(\gamma)$. Here, the function x is the sum of orthogonal sine and cosine functions. The complex coefficients c_k can be found by applying the above mentioned inner product with the basis function $y_k(t) = e^{i\frac{2\pi k t}{T}}$

$$c_k = \frac{1}{T} \langle x(t), e^{i\frac{2\pi k t}{T}} \rangle = \frac{1}{T} \int_T x(t) \cdot e^{-i\frac{2\pi k t}{T}} dt. \tag{A.27}$$

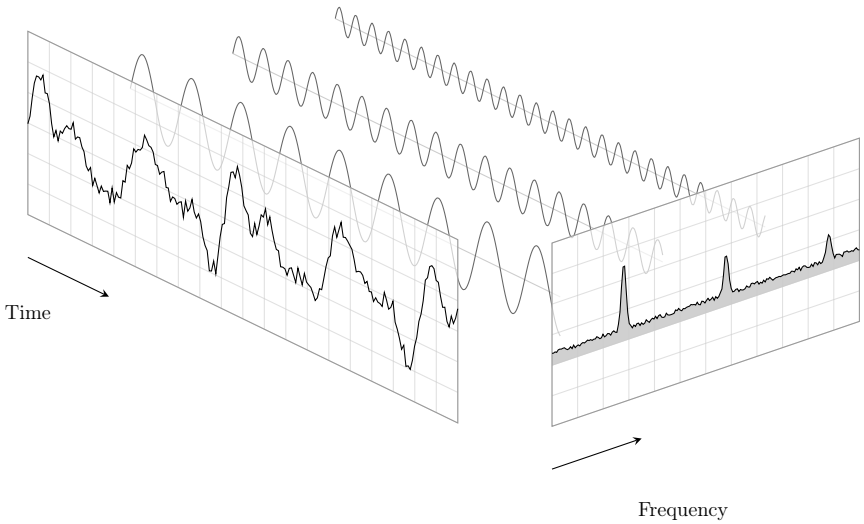

Figure A.6.: Visual representation of Fourier analysis [137].

A.3.2. Fourier Transform

The Fourier series is defined only for periodic functions. To also study non-period functions, the *Fourier transform* is introduced. It is an extension of the Fourier series that results when the period of the function is allowed to approach infinity [64, 136]. The Fourier transform integral is essentially the limit of a Fourier series as the domain's length goes to infinity [64, 86]. Letting $T \to \infty$ and defining $\omega_0 = \frac{2\pi}{T}$ yields

$$\begin{aligned} x(t) &= \lim_{T\to\infty} \sum_{k=-\infty}^{\infty} c_k e^{i\frac{2\pi kt}{T}} \\ &= \lim_{T\to\infty} \sum_{k=-\infty}^{\infty} \left(\frac{1}{T} \int_T x(t) \cdot e^{-i\frac{2\pi kt}{T}} dt \right) e^{i\frac{2\pi kt}{T}} \\ &= \lim_{w_0\to 0} \sum_{k=-\infty}^{\infty} \frac{\omega_0}{2\pi} \left(\int_{-\frac{\pi}{\omega_0}}^{\frac{\pi}{\omega_0}} x(t) \cdot e^{-ik\omega_0 t} dt \right) e^{ik\omega_0 x} \\ &= \lim_{w_0\to 0} \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \hat{x} \cdot e^{ik\omega_0 t} \omega_0. \end{aligned} \tag{A.28}$$

The discrete product $k\omega_0$ becomes a continuous frequency variable $k\omega_0 \to \omega$, and the summation becomes a Riemann integral. This results in

$$x(t) = \mathcal{F}^{-1}\{\hat{x}(\omega)\}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{x}(\omega) e^{i\omega t} d\omega \tag{A.29}$$

$$\hat{x}(\omega) = \mathcal{F}\{x(t)\}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt, \tag{A.30}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the Fourier transform and inverse Fourier transform, respectively. Both functions converge as long as $\int_{-\infty}^{\infty} |x(t)| dt < \infty$ and $\int_{-\infty}^{\infty} |\hat{x}(\omega)| d\omega < \infty$ [64, 86]. In engineering terms, this is equivalent to saying that the energy of the function (or the signal) needs to be finite.

Parsevals theorem states that the energy of signals is preserved up to a constant when applying the Fourier transform [64, 136]

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{x}(\omega)|^2 d\omega, \tag{A.31}$$

which is an important property when dealing with energy detectors in the frequency domain, e.g., wideband energy detection (see section 3.2.2).

The *power spectral density* (PSD) $S(f)$ of $x(t)$, with $\omega = 2\pi f$, describes the power distribution over the frequency and can be estimated by taking the square of the absolute value of the Fourier transform

$$\mathrm{S}_x(f) = |\hat{x}(f)|^2. \tag{A.32}$$

The PSD can also be expressed in terms of the Fourier transform of the autocorrelation function

$$\mathrm{R}_{xx}(\tau) = \mathrm{E}[x(t)^* x(t-\tau)] \tag{A.33}$$

$$\mathrm{S}_x(f) = \int_{-\infty}^{\infty} \mathrm{R}_{xx}(\tau) e^{-i2\pi f\tau} dt = \hat{\mathrm{R}}_{\mathrm{xx}}(f), \tag{A.34}$$

where x is assumed to be *wide-sense stationary* (WSS); this relation is known as the *Wiener–Khinchin theorem* [136].

A.3.3. Discrete Fourier Transform

The *discrete Fourier transform* (DFT) is a discretized version of the Fourier transform, which operates on vectors of data $\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ that are obtained by sampling continuous functions (or signals) [86].

Closely related and of paramount importance to signal discretization is the so-called *Nyquist–Shannon* sampling theorem. It states that any signal sampled at frequency f_s must contain no sinusoidal component at exactly or above half the sample frequency f_s. Strictly speaking, any signal with bandwidth B has to be sampled at $f_s > 2B$, where the threshold $2B$ is called *Nyquist rate* [136].

One can transform N discrete samples of function x from the time domain to an N-point discrete spectral representation with the DFT, defined as

$$\hat{x}_k = \sum_{j=1}^{N} x_j e^{-i\frac{2\pi jk}{N}}. \tag{A.35}$$

where k is a variable of discrete frequency. Its inverse, the iDFT, is given by

$$x_n = \frac{1}{N} \sum_{j=1}^{N} \hat{x}_j e^{i\frac{2\pi jn}{N}}. \tag{A.36}$$

Comparing these transforms to the above presented inner products of vectors shows that the DFT is a linear transformation that maps the data points of x (time-domain) to $\hat{x}$ (frequency-domain). Thus, this transformation can be computed by matrix multiplication [86]

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \vdots \\ \hat{x}_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \tag{A.37}$$

with the DFT matrix of size $N \times N$, where $\omega_N = e^{\frac{-2\pi i}{N}}$.

The discrete version of Parseval's theorem, which states that the total signal energy is preserved by the DFT, is

$$\sum_{n=1}^{N} |x_n|^2 = \frac{1}{N} \sum_{k=1}^{N} |\hat{x}_k|^2. \tag{A.38}$$

The discrete PSD is defined as

$$S_{x,k} = \frac{1}{N} |\hat{x}_k|^2, \tag{A.39}$$

where, depending on the context, $1/N$ is sometimes left out.

A.3.4. Fast Fourier Transform

Multiplying by the DFT matrix involves $\mathcal{O}(N^2)$ operations, making it too slow to be practically applicable. This is where the *fast Fourier transform* (FFT) comes in [138]. Popularized by J. W. Cooley and John Tukey, it scales with $\mathcal{O}(N \log N)$, enabling it to be used in many real-time applications [100]. The fundamental idea of the FFT is to subdivide the large DFT matrix recursively into smaller matrices. This can be done most efficiently if N is a power of 2, which yields the so-called *radix-2 algorithm*. The even and odd entries of the N-long input get split into two smaller $N/2$-long DFTs

$$\begin{aligned} \hat{x}_k &= \sum_{j=1}^{N} x_j e^{-i\frac{2\pi jk}{N}} \\ &= \sum_{j=1}^{N/2} x_{2j} e^{-i\frac{2\pi 2jk}{N}} + \sum_{j=1}^{N/2} x_{2j+1} e^{-i\frac{2\pi(2j+1)k}{N}} \\ &= \underbrace{\sum_{j=1}^{N/2} x_{2j} e^{-i\frac{2\pi jk}{N/2}}}_{N/2 \text{ DFT}} + \underbrace{e^{-i\frac{2\pi k}{N}}}_{\omega_N^n} \underbrace{\sum_{j=1}^{N/2} x_{2j+1} e^{-i\frac{2\pi jk}{N/2}}}_{N/2 \text{ DFT}} \end{aligned} \tag{A.40}$$

which for $N = 8$ can be visualized at the highest recursion level as seen in fig. A.7.

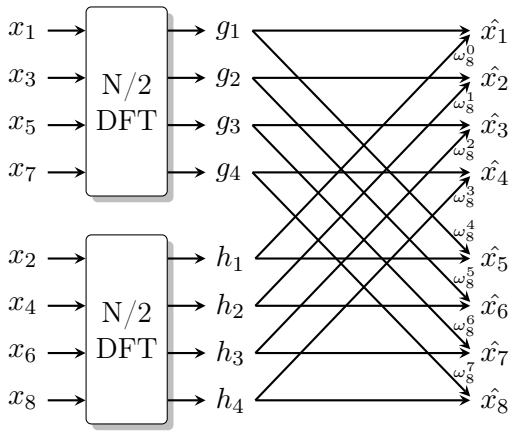

Figure A.7.: 8-point Radix-2 FFT high-level signal flow.

Repeating this process recursively for any input size down to $N = 2$, computing all DFTs on the way down, and then bringing the results back together yields the improved run-time FFT. It is important to remember that the FFT and the DFT return the same result; the FFT simply improves the computational complexity for large N.

The FFT is implemented in various software packages. The most relevant implementations are the FFTPACK implementation, written in FORTRAN [139], which is used by NumPy [111] and SciPy [112]. There also exists the FFTW implementation, which is written in C [140]. The FFTW uses advanced heuristics to achieve superb performance, making it faster than the FFTPACK implementation. However, it uses a more strict GPL license and is therefore not used by NumPy or SciPy. Python bindings for the FFTW are nonetheless available through pyFFTW [141].

Using hardware implementations makes the FFT even more efficient. In the context of SALSAT, Alexander Maaß has already implemented an Intel FFT IP-Core on the Cyclone IV FPGA [92]. Therefore, the details of FFT FPGA implementations will not be presented here.

A.3.5. Short-Time Fourier Transform

While the Fourier Transform provides detailed information on the frequency content of a given signal, it does not yield any information about when in time those frequencies occur. Thus, is it only able to characterize truly periodic and stationary signals. However, for signals with non-stationary frequency content, it is important to simultaneously characterize the frequency and its evolution over time [86].

The *short-time Fourier transform* (STFT) computes windowed DFTs using a moving window $g(t)$. This results in a series of single DFTs, each having the length of the window function. The STFT provides localization in both the temporal domain, as well as in the frequency domain. It assumes that the signal is stationary during the application of the window function to each chunk of the signal, a property called *quasi-stationary*. While this assumption does generally not hold, it results

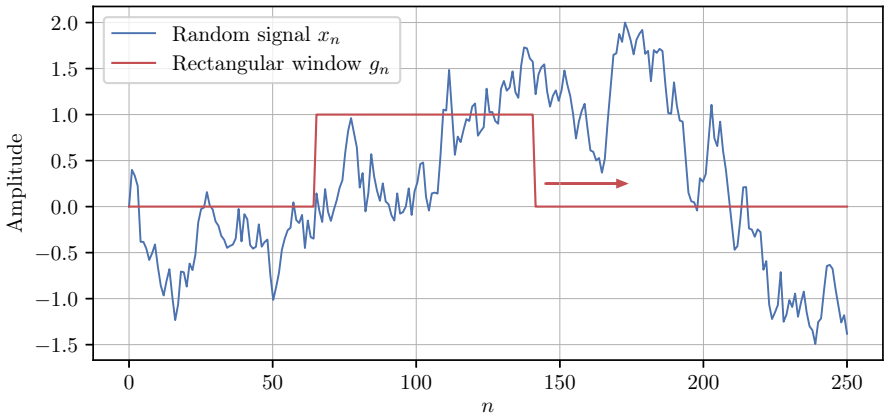

Figure A.8.: Illustration of the STFT with rectangular sliding window.

in sufficiently good spectral estimation results. The continuous STFT pair can be written as

$$\hat{x}(\omega, t) = \int_{-\infty}^{\infty} x(\tau) e^{-i\omega\tau} g^*(\tau - t) d\tau \tag{A.41}$$

$$x(t) = \frac{1}{2\pi \|g\|^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{x}(\omega, \tau) e^{i\omega t} g(\tau - t) d\omega d\tau, \tag{A.42}$$

where $\|g\|^2$ is the energy of the window function [64, 136]. While the summation, as stated above, goes from $-\infty$ to ∞, the finite window reduces the summation range to the window size. It is important to note that the STFT $\hat{x}(\omega, t)$ now has a frequency ω *and* a time t variable.

The discrete counterpart of the STFT is defined as

$$\hat{x}_{m,k} = \sum_{n=m-L/2}^{m+L/2} x_{n+mL}\, e^{-i\frac{2\pi k n}{L}} g_n \tag{A.43}$$

with discrete window function g_n of even size L, so that $k \in [-L/2, L/2]$ becomes the discrete-frequency variable and $m \in [0, \lfloor N/L - 1 \rfloor]$ the discrete-time variable with which one selects the DFT-bin of interest [64]. Thus, for an N entry discrete sample vector x, one has $\lfloor N/L \rfloor$ DFT-bins.

The simplest window function is the *rectangular window*. It is equivalent to replacing all but L values of the data samples in x by zeros, making it appear as though the signal suddenly turns *on* and *off* [135]. It is an example of a window with high-frequency resolution but low dynamic range. Of particular interest to this thesis (next to the rectangular window) are the *Hann window* and the *flattop window*. Both present a compromise between dynamic range and resolution. While the Hann window is versatile and well suited for general application, the flattop window is well suited for PSD estimation, as it has a better amplitude accuracy in the frequency domain compared to the Hann window [142]. The board topic of window functions would go beyond the scope of this thesis. For more information about window functions, in conjunction with the important topic of spectral leakage, consult [92, 135, 142, 143].

An important tuning factor is the size of the window function; and likewise the length of DFT. One cannot simultaneously localize a signal x in both the time domain and frequency domain. The shorter the DFT, the finer the time resolution. At the same time, the frequency resolution decreases.

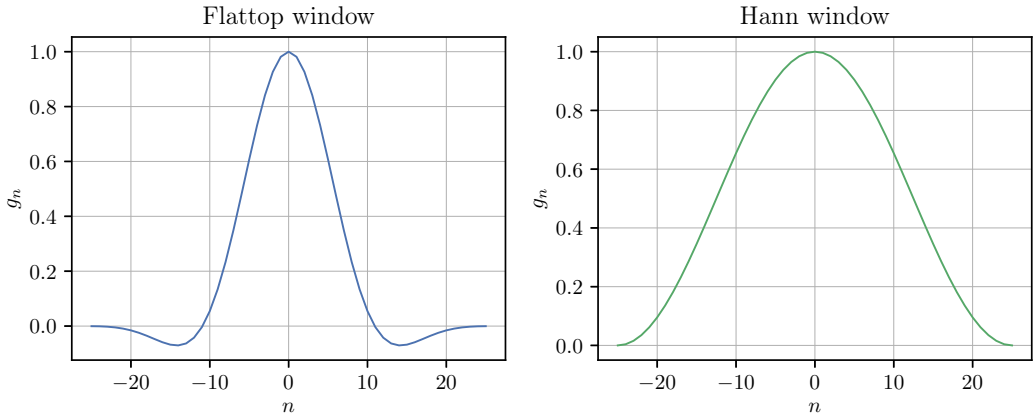

Figure A.9.: Flattop and Hann window with size $n = 50$.

When increasing the DFT length, the time resolution decreases, and the frequency resolution increases. This observation leads to the fundamental uncertainty principle of time-frequency analysis that limits the ability to simultaneously attain high resolution on both the time and frequency domains. Stated mathematically: for the continuous function $x(t)$, the time-frequency uncertainty principle can be written as

$$\left(\int_{-\infty}^{\infty} t^2|x(t)|^2dt\right)\left(\int_{-\infty}^{\infty} \omega^2|\hat{x}(\omega)|^2d\omega\right) \geq \frac{1}{16\pi^2}. \tag{A.44}$$

if both $tx(t)$ and $x'(t)$ are square integrable [86].

The *spectrogram* (or sometimes called *waterfall plot*) is a two-dimensional representation of the squared magnitude of the STFT. It is essentially a series of PSD absolute values indexed by time. The discrete spectrogram is defined as

$$\mathrm{S}_{m,k} = |\hat{x}_{m,k}|^2 \tag{A.45}$$

with m as the discrete-time variable and k as the index of discrete frequency. A typical format is a two-dimensional graph; one axis represents time, and the other axis represents frequency. The amplitude of a specific frequency at a given time is expressed by the strength or color of each point in the graph (see fig. A.10 and appendix C for examples) [144].

A.3.6. Wavelet Transform

Wavelets (or *wavelet transforms*) extend the concept of Fourier analysis to more general orthogonal basis functions and partially overcome the uncertainty principle discussed above by leveraging multiresolution decomposition [64, 85, 86, 96, 145]. This multiresolution approach allows for distinct time and frequency fidelities in different frequency bands. Starting with the *mother* wavelet $\psi(\tau)$, a family of scaled and translated wavelet functions is formed [86]

$$\psi_{a,t}(\tau) = \frac{1}{\sqrt{a}}\psi\left(\frac{\tau - t}{a}\right). \tag{A.46}$$

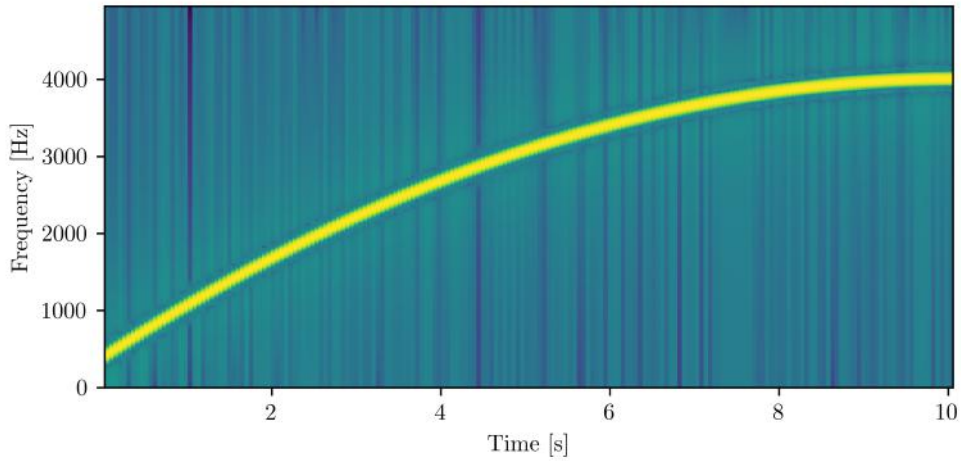

Figure A.10.: Spectrogram over 10s of a quadratic chirp signal from 400Hz to 4000Hz.

The parameters a and t are responsible for scaling and translating the wavelet $\psi(t)$ [86]. A key advantage over Fourier transforms is the adaptive temporal and frequency resolution: wavelets capture both time and frequency information, just as the STFT does, but with varying resolution. In low-frequency regimes, the *wavelet transform* (WT) provides good frequency resolution, and in high-frequency regimes, as frequency resolution gets less important, it provides good time resolution. This phenomenon is visualized in fig. A.11. The WT

$$\begin{aligned}\mathcal{W}_\psi\{x\}(a,t) = \langle x, \psi_{a,t}\rangle &= \int_{-\infty}^{\infty} x(\tau)\psi^*_{a,t}(\tau)d\tau \\ &= \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(\tau)\psi^*\left(\frac{\tau - t}{a}\right)d\tau \end{aligned} \tag{A.47}$$

can, analogous to the Fourier transform, be thought of as a projection into the orthogonal wavelet basis of scaled wavelets all derived from the same mother wavelet $\psi(\tau)$ [64, 86].

Two important wavelet types are the *Gaussian derivative* and *Mexican hat* wavelets (fig. A.12), which are extensively used in edge detection applications. The equation for the Gaussian wavelet family are [113]

$$\begin{aligned}\psi &= e^{-n^2} \\ \psi'(n) &= -2ne^{-n^2} \\ \psi''(n) &= (4n^2 - 2)e^{-n^2}.\end{aligned} \tag{A.48}$$

When working with *digital* data, it is necessary to discretize the WT. The scaling is usually chosen to be *dyadic*, i.e., powers of two: $a_k = 2^k$. With m as the time variable, the discrete wavelet family then is

$$\psi_{m,k}(\tau) = \frac{1}{\sqrt{2^k}}\psi(\frac{\tau - m}{2^k}), \tag{A.49}$$

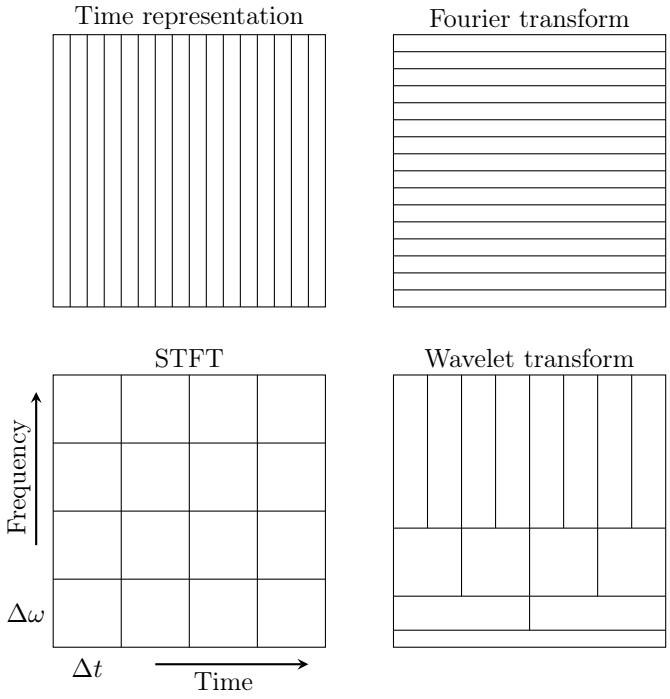

Figure A.11.: Illustration of resolution limitations and uncertainty in time-frequency analysis [86].

where the scaling factor is often chosen without the square-root so that the *discrete wavelet transform* (DWT) is given by

$$\begin{aligned}\mathcal{W}_{m,k}\{x\} = \langle x, \psi_{m,k}\rangle &= \int_{-\infty}^{\infty} x(\tau)\psi^*_{m,k}(\tau)d\tau \\ &= \frac{1}{2^k}\int_{-\infty}^{\infty} x(\tau)\psi^*(\frac{\tau - m}{2^k})d\tau. \end{aligned} \tag{A.50}$$

The discrete wavelet and the DWT can also be expressed in a more compact form with the help of the convolution operator, which has to be interpreted carefully to avoid confusion

$$\psi_s(t) = \frac{1}{s}\psi\left(\frac{t}{s}\right) \tag{A.51}$$

$$\mathcal{W}_s(t) = x(t) * \psi_s(t) = \frac{1}{s}\int_{-\infty}^{\infty} x(\tau)\psi^*(\frac{\tau - t}{s})d\tau, \tag{A.52}$$

where $s = 2^m, m = 1, 2, \ldots, M$ is called *dilation factor* [98]. This is the notation used in spectrum sensing literature and thus will also be used in this thesis.

The DWT has a similarly fast implementation as the FFT, called the *fast wavelet transform* (FWT). It uses cascading high-pass $h(n)$ and low-pass $g(n)$ filter pairs, known as *quadrature mirror filters*. After each filter level, the filter results are down-sampled by a factor of 2. The down-sampled results of the high-pass filter represent the coefficients for that level, while the down-sampled low-pass filter results are used for the next level. These filters are commonly implemented using *perfect reconstruction filter banks*. The topic of filter banks and quadrature mirror filters would go beyond the scope of this quick summary. For more information, see [145, 146].

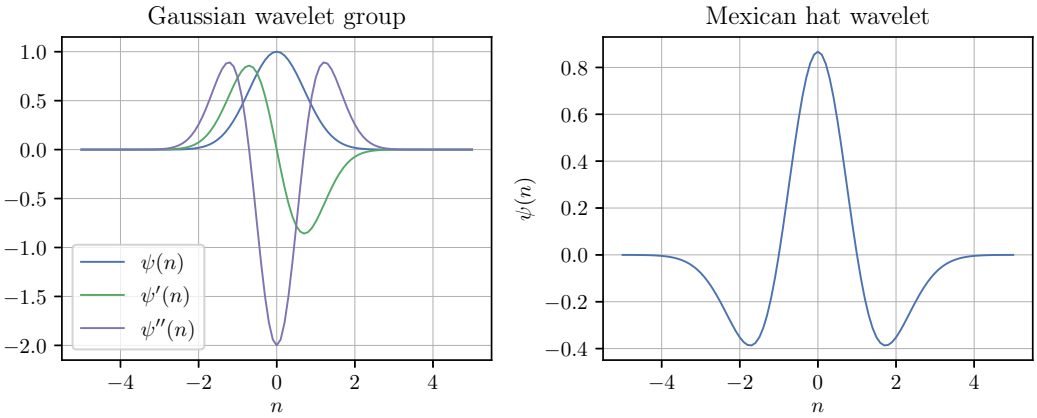

Figure A.12.: Real Gaussian derivative and mexican hat wavelets.

Because SciPy does not support wavelets well, the PyWavelets package was chosen for this work. It provides a wide range of WTs, as well as decent documentation [113]. As is common in the area of wavelet application, the terms *continuous wavelet transform* and *discrete wavelet transform* are used somewhat inconsistently. The DWT is often called CWT, and the DWT is then used to describe a different type of WT where some number of samples are skipped. It is important to note that both the CWT and DWT become point-by-point digital transformations in this case. However, for this thesis, the *special* DWT is not relevant. To stay consistent with the available literature on wavelet-based spectrum sensing, the DWT presented above will nonetheless be called CWT in this thesis.

An important application of WTs is in singularity or edge detection. Because the WT characterizes the local regularity of signals by decomposing them into elementary building blocks that are well localized, one can use it to detect gradient changes even on very noisy data samples. The effect of noise can be reduced by adjusting the scale of the wavelet: wavelets of large scales are more effective at removing noise components, but at the same time increase the uncertainty of the location of edges, while wavelets of small scales preserve the exact location of edges but cannot distinguish between noise and *actual* edges [147]. The wavelet of particular interest to edge detection in this thesis is the Gaussian derivative wavelet (see fig. A.12). It is important to note that the WT, when performing edge detection, is not necessarily operating on time-domain signals. Instead, it can be applied to all types of data or signals; in this thesis it will be applied to the PSD.

B. More Sensing Algorithms

B.1. Narrowband

B.1.1. Matched Filter Detection

When the signal trying to detect is (partially) known, one can use the *matched filter* detector (sometimes also known as *coherent detector* or *replica correlator*) [44, 60, 65]. It is the optimal linear detector for detecting a known signal in the presence of Gaussian noise and works by correlating the known signal with the incoming measurement samples. Prime application examples are radar and sonar, since the signal to detect is known: the signal is sent out, reflected by some object, and received as an attenuated, phase-shifted, and noise polluted version of the signal transmitted moments before.

Detector

As the signal $\mathbf{s} = [s_1, \ldots, s_N]^{\mathrm{T}} \in \mathbb{C}^N$ is assumed known and received together with additive CWGN (see chapter 2), the two PDFs under both hypothesis are

$$p(\mathbf{x} \mid \mathcal{H}_0) = \frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right) \tag{B.1}$$

$$p(\mathbf{x} \mid \mathcal{H}_1) = \frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}(\mathbf{x}-\mathbf{s})^{\mathrm{H}}(\mathbf{x}-\mathbf{s})\right). \tag{B.2}$$

The PDF under $\mathcal{H}_0$ is simply the complex multivariate Gaussian distribution of the noise $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \sigma_w^2\mathbf{I})$. Under $\mathcal{H}_1$ this distribution is shifted by the received signal $\mathbf{s}$, so that $\mathbf{x} \sim \mathcal{CN}(\mathbf{s}, \sigma_w^2\mathbf{I})$. Using these PDFs it is possible to construct the detector based on the LRT [60]

$$\begin{aligned}\Lambda(\mathbf{x}) &= \frac{p(\mathbf{x} \mid \mathcal{H}_1)}{p(\mathbf{x} \mid \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda \\ &= \frac{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}(\mathbf{x}-\mathbf{s})^{\mathrm{H}}(\mathbf{x}-\mathbf{s})\right)}{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}\end{aligned} \tag{B.3}$$

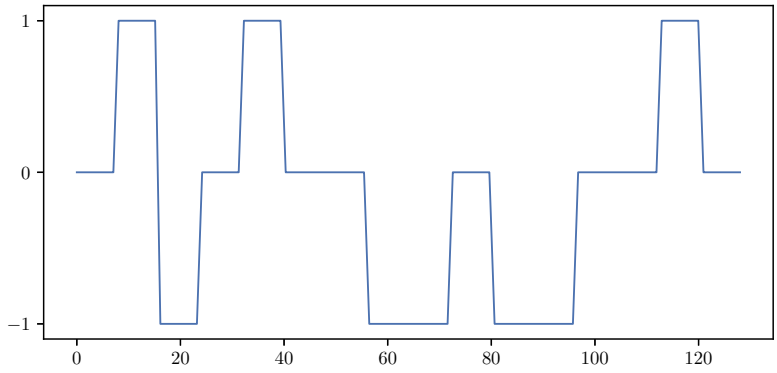

Figure B.1.: Simple exemplary signal $\mathbf{s} = [0, 1, -1, 0, 1, 0, 0, -1, -1, 0, -1, -1, 0, 0, 1, 0]^{\mathrm{T}}$ sampled over $N = 128$ data points.

and after utilizing the the natural logarithms monotonic property to simplify, one has

$$\begin{aligned}
\ln\left(\Lambda\left(\mathbf{x}\right)\right) &= \ln\left(\frac{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\left(\mathbf{x}-\mathbf{s}\right)^{\mathrm{H}}\left(\mathbf{x}-\mathbf{s}\right)\right)}{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}\right) \\
&= -\frac{1}{\sigma_w^2}\left(\left(\mathbf{x}-\mathbf{s}\right)^{\mathrm{H}}\left(\mathbf{x}-\mathbf{s}\right) - \mathbf{x}^{\mathrm{H}}\mathbf{x}\right) \\
&= -\frac{1}{\sigma_w^2}\left(-\mathbf{x}^{\mathrm{H}}\mathbf{s} - \mathbf{s}^{H}\mathbf{x} + \mathbf{s}^{\mathrm{H}}\mathbf{s}\right) \\
&= \frac{1}{\sigma_w^2}\left(2\,\mathrm{Re}\left(\mathbf{s}^{\mathrm{H}}\mathbf{x}\right) - \mathbf{s}^{\mathrm{H}}\mathbf{s}\right),
\end{aligned} \tag{B.4}$$

because $\mathbf{x}^{\mathrm{H}}\mathbf{s} + \mathbf{s}^{\mathrm{H}}\mathbf{x} = (\mathbf{s}^{\mathrm{H}}\mathbf{x})^* + \mathbf{s}^{\mathrm{H}}\mathbf{x} = 2\,\mathrm{Re}(\mathbf{s}^{\mathrm{H}}\mathbf{x})$ [60]. Since $\mathbf{s}$ and σ_w^2 are assumed known, one can move the corresponding known parts into a new threshold λ'

$$\mathrm{T}\left(\mathbf{x}\right) = \mathrm{Re}\left(\mathbf{s}^{\mathrm{H}}\mathbf{x}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda', \tag{B.5}$$

which yields the final test statistic $\mathrm{T}(\mathbf{x})$ [26, 30, 60]. This can also be written as

$$\mathrm{T}\left(\mathbf{x}\right) = \mathrm{Re}\left(\sum_{n=1}^{N} x_n s_n^*\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda'. \tag{B.6}$$

Shifting the known signal $\mathbf{s}$ through the data stream while constantly multiplying and summing leads to the resulting scalar value being largest when the received data stream $\mathbf{x}$ most closely matches the known signal $\mathbf{s}$. This can also be thought of as an optimal filter, tuned to *only* let the known signal pass through; hence the name: *matched filter*.

Example

To better understand the operation of the matched filter, it is beneficial to briefly visit a simple example. Assuming one is trying to detect the simplified signal from fig. B.1 in a set of noise observations. Constructing a sequence in which the signal will only be present at certain times

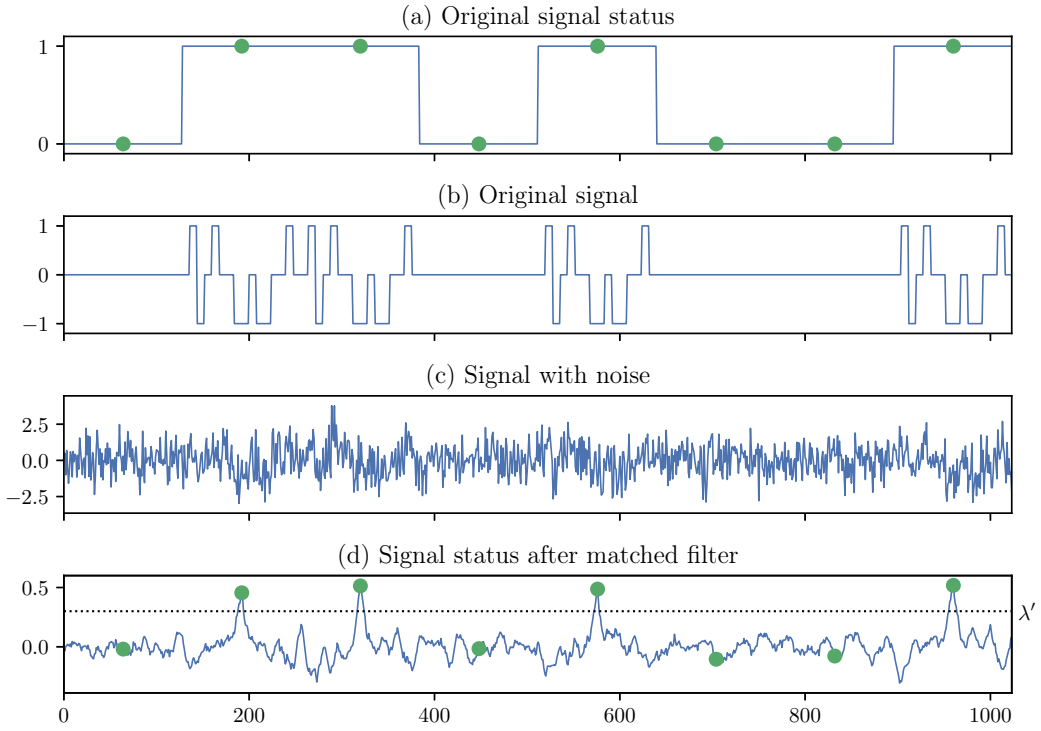

Figure B.2.: (b) contains the known signal from fig. B.1 according to the signal status in (a). Introducing noise in (c) makes it impossible to see whether the signal is present or not with the naked eye. After matched filtering and thresholding, the signal status is nonetheless fully recovered (d) (derived from [112]).

according to fig. B.2 (a) results in fig. B.2 (b). 0 indicates that the signal is not present, while 1 indicates the signal is present. After adding WGN it is not possible to make out whether the signal is present or not in fig. B.2 (c). The SNR is about $-6\,\mathrm{dB}$ in this example. By choosing an arbitrary threshold $\lambda' = 0.3$ and applying the matched filter, one can clearly identify whether the signal is present or not (fig. B.2 (d)).

Limitations

While the matched filter achieves superb performance even in very low SNR, it is not applicable to the problem of spectrum sensing onboard SALSAT. According to MR-01 the detector needs to be blind; the signals trying to detect are not known, thus ruling out the matched filter. Even if all signals were to be known, it would not be practically feasible to implement a matched detector for every signal trying to detect. Therefore, it is necessary to generalize the detector from known deterministic signals to stochastic signal models (see ED in section 3.1.1).

B.1.2. Covariance Detection

When the signals covariance matrix $\mathbf{C}_s$ is known, one can use the so-called *covariance detector* to detect its presence in CWGN. The detector can handle noise uncertainties and improve low SNR performance.

Detector

The following detector is going to be derived in accordance with [60]. It will be assumed that $\mathbf{s}$ is a zero-mean, Gaussian random signal with known covariance matrix $\mathbf{C}_s$; that is $\mathbf{s} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_s)$. In contrast to the model used by the eigenvalue detector (see section 3.1.2), this detector will deliberately use known features from the covariance matrix. The noise is assumed to be uncorrelated $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_w^2\right)$. This results in the following signal model (see section 2.2)

$$\mathcal{H}_0 : \mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \sigma_w) \tag{B.7}$$

$$\mathcal{H}_1 : \mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_s + \sigma_w). \tag{B.8}$$

As has been done before, the LRT will be used to construct the test (see section 2.3) [60]

$$\begin{aligned}
\Lambda(\mathbf{x}) &= \frac{p\left(\mathbf{x} \mid \mathcal{H}_1\right)}{p\left(\mathbf{x} \mid \mathcal{H}_0\right)} \\
&= \frac{\frac{1}{\pi^N \det\left(\mathbf{C_s}+\sigma_w^2\mathbf{I}\right)} \exp\left(-\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\mathrm{H}}\left(\mathbf{C_s}+\sigma_w^2\mathbf{I}\right)^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right)}{\frac{1}{\pi^N \det\left(\sigma_w^2\mathbf{I}\right)} \exp\left(-\left(\mathbf{x}-\boldsymbol{\mu}\right)^{\mathrm{H}}\left(\sigma_w^2\mathbf{I}\right)^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right)} \\
&\overset{\boldsymbol{\mu}=\mathbf{0}}{=} \frac{\frac{1}{\pi^N \det\left(\mathbf{C_s}+\sigma_w^2\mathbf{I}\right)} \exp\left(-\mathbf{x}^{\mathrm{H}}\left(\mathbf{C_s}+\sigma_w^2\mathbf{I}\right)^{-1}\mathbf{x}\right)}{\frac{1}{\pi^N \sigma_w^{2N}} \exp\left(-\frac{1}{\sigma_w^2}\mathbf{x}^{\mathrm{H}}\mathbf{x}\right)}.
\end{aligned} \tag{B.9}$$

Using the logarithm, one can simplify

$$\ln\left(\Lambda(\mathbf{x})\right) = -\ln\left(\frac{\sigma_w^{2N}}{\det\left(\mathbf{C_s}+\sigma_w^2\mathbf{I}\right)}\right)\mathbf{x}^{\mathrm{H}}\left(\left(\mathbf{C}_s+\sigma_w^2\mathbf{I}\right)^{-1} - \frac{1}{\sigma_w^2}\mathbf{I}\right)\mathbf{x}. \tag{B.10}$$

Dividing by the logarithm term on the left and multiplying by σ_w^2 one only retains the data-dependent terms [60]

$$\mathrm{T}(\mathbf{x}) = \mathbf{x}^{\mathrm{H}}\left(\mathbf{I} - \sigma_w^2\left(\mathbf{C}_s+\sigma_w^2\mathbf{I}\right)^{-1}\right)\mathbf{x}.$$

Using a slightly simplified version of the *Woodbury matrix identity*, which states that $(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\left(\mathbf{B}^{-1}+\mathbf{A}^{-1}\right)^{-1}\mathbf{A}^{-1}$, one can rearrange the equation into a more elegant form

$$\begin{aligned}
\text{with} \quad \left(\mathbf{C}_s + \sigma_w^2\mathbf{I}\right)^{-1} &= \frac{1}{\sigma_w^2}\mathbf{I} - \frac{1}{\sigma_w^4}\left(\frac{1}{\sigma_w^2\mathbf{I}} + \mathbf{C}_s^{-1}\right)^{-1} \\
\mathrm{T}(\mathbf{x}) &= \mathbf{x}^{\mathrm{H}}\left(\mathbf{I} - \sigma_w^2\left(\frac{1}{\sigma_w^2}\mathbf{I} - \frac{1}{\sigma_w^4}\left(\frac{1}{\sigma_w^2\mathbf{I}} + \mathbf{C}_s^{-1}\right)^{-1}\right)\right)\mathbf{x} \\
&= \mathbf{x}^{\mathrm{H}}\left(\frac{1}{\sigma_w^2}\left(\frac{1}{\sigma_w^2}\mathbf{I} + \mathbf{C}_s^{-1}\right)^{-1}\right)\mathbf{x} \\
&= \mathbf{x}^{\mathrm{H}}\left(\frac{1}{\sigma_w^2}\left(\frac{1}{\sigma_w^2}\left(\mathbf{C}_s + \sigma_w^2\mathbf{I}\right)\mathbf{C}_s^{-1}\right)^{-1}\right)\mathbf{x} \qquad \text{(B.11)}
\end{aligned}$$

which yields the final detector [60]

$$\mathrm{T}(\mathbf{x}) = \mathbf{x}^{\mathrm{H}}\mathbf{C}_s\left(\mathbf{C}_s + \sigma^2\mathbf{I}\right)^{-1}\mathbf{x} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda. \qquad \text{(B.12)}$$

This detector can be thought of taking the incoming signal $\mathbf{x}$ and correlating it with itself, while weighing the relevant parts of the known covariance matrix $\mathbf{C}_s$ respectively.

Limitations

According to MR-01, the detector needs to be blind. The covariance detctor however assumes knowledge of the covariance matrix $\mathbf{C}_s$. It is thus not applicable to SALSAT.

B.1.3. Cyclostationary Detection

If the signal $\mathbf{s}$ is not entirely known, but some of its features are, one may take advantage of this knowledge to form a test statistic closely matched to the signal. A feature capable of distinguishing a modulated signal from noise is obtained by observing that, generally, modulated signals are not stationary, meaning that they have statistical properties that vary cyclically with time [26]. This property is called *cyclostationarity* and allows these types of signals to be modeled as cyclostationary random processes. This is in contrast to noise $\mathbf{w}$, which is assumed to be a stationary random processes. Most man-made signals show periodic patterns related to symbol rate, channel code, cyclic prefix, etc., that introduce cyclostationarity. Thus, cyclostationary signal detection consists of analyzing the cyclic autocorrelation function of the received signal, which is periodic for data signals but non-periodic for noise [31].

Detector

A discrete-time zero-mean ($\mathrm{E}[\mathbf{x}] = 0$) stochastic signal $\mathbf{x} = [x_1, \ldots, x_N]^{\mathrm{T}} \in \mathbb{C}^N$ is said to be wide-sense cyclostationary with period T_0 if its autocorrelation function

$$\mathrm{R}_x(n,\tau) = \mathrm{E}[x_n^* x_{n-\tau}] \qquad \text{(B.13)}$$

is cyclic in n with period T_0

$$\mathrm{R}_x(n,\tau) = \mathrm{R}_x(n+T_0,\tau). \tag{B.14}$$

The autocorrelation function can consequently be expanded as a Fourier series (see appendix A.3.1) [26]

$$\mathrm{R}_x(n,\tau) = \sum_{\alpha} \mathrm{R}_x^{\alpha}(\tau) e^{i2\pi\alpha n} \tag{B.15}$$

with Fourier coefficients $\mathrm{R}_x^{\alpha}(\tau)$, also known as the *cyclic autocorrelation function*, defined by

$$\mathrm{R}_x^{\alpha}(\tau) = \sum_{n=0}^{T_0-1} \mathrm{R}_x(n,\tau) e^{-i2\pi\alpha n} \tag{B.16}$$

and where $\alpha = k/T_0, k \in \mathbb{Z}$ are called *cyclic frequencies*. The Fourier coefficients, interpreted as a function of τ, form the *cyclic autocorrelation function* (CAF) of x_n. The signal is thus said to be cyclostationary, if there exists an $\alpha \neq 0$ such that $\mathrm{R}_x^{\alpha}(\tau) > 0$ [31]. The *cyclic spectral density* (CSD) S_x^{α} of x_n is defined as the discrete Fourier transform of $\mathrm{R}_x^{\alpha}(\tau)$

$$\mathrm{S}_x^{\alpha}(f) = \sum_{\tau} \mathrm{R}_x^{\alpha}(\tau) e^{-i2\pi f\tau}. \tag{B.17}$$

Knowing T_0, or some other cyclic characteristics of signal x_n, one can use the CAF to detect the signal by evaluating $\mathrm{R}_x^{\alpha}(\tau)$ at T_0. Equivalently, the CSD has peaks at cyclic frequencies that are multiples of $1/T_0$ [26, 31].

Limitations

Cyclostationary signal detection is generally dependent on the assumption that the period T_0 of x_n is known a priori [26]. If this assumption is not valid, as is the case with SALSAT (MR-01), then cyclostationary detection requires the search for cyclic frequencies. This has been implemented for a subspace of all possible cyclic frequencies in [52]; more specifically, the authors chose $\alpha \in [-f_s, f_s]$ and $f \in [-f_s/2, f_s/2]$ for $\mathrm{S}_x^{\alpha}(f)$. The resulting search leads to increased complexity and longer processing times. The authors reported $2 \sim 3$ orders of magnitude increase compared to the eigenvalue detector, and $4 \sim 5$ orders of magnitude increase compared to the ED; with similar or slightly worse detection performance in unknown noise environments as the eigenvalue detector. This makes the cyclostationary detector not viable as a blind spectrum sensing method for SALSAT (CT-06).

B.2. Wideband

B.2.1. Compressed Sensing

The primary motivation behind *compressed sensing* (CS) is to significantly reduce the sampling rate f_s below the Nyquist rate $f_s > 2B$; and thereby either increase the available bandwidth B one can observe or lower the demands put upon the RF system [30, 31, 79, 80]. The reduction of

the sampling rate below the Nyquist rate is possible because most signals are sparse, meaning that they can be fully described by very few values when represented in a suitable basis [86]. Because the wideband spectrum is generally underutilized, as only a few of the overseen subbands are concurrently active (this is the main motivation for introducing cognitive radio in the first place), the processed wideband signal $\mathbf{x}$ is inherently sparse in the frequency domain [79]. The minimum required sampling rate is thus proportional to the joint bandwidth of the active subbands B_A, instead of the entire monitored spectrum bandwidth B. Therefore, CS appears to be a good candidate for wideband spectrum sensing. One however, has to note that CS itself does not incorporate signal detection. Only in conjunction with other signal detection methods is CS able to function as a spectrum sensing method. Because of its complexity, only a brief overview of CS will be given here. For a more rigorous treatment of the topic, consult [86, 148].

Detector

Let $\mathbf{y} \in \mathbb{C}^M$ denote the sub-Nyquist samples of the data $\mathbf{x} \in \mathbb{C}^N$, which encompasses the full wideband signal and its active transmissions, captured at or above the Nyquist rate. They are related via

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} \tag{B.18}$$

where $\mathbf{\Phi} \in \mathbb{C}^{M \times N}$ is called the measurement matrix [79, 86]. $\mathbf{\Phi}$ is a random matrix with some entries set to non-zero values. While the choice of $\mathbf{\Phi}$ is of critical importance, it will not be discussed any further here.

The task at hand is to reconstruct the original signal $\mathbf{x}$ from $\mathbf{y}$ using very few samples; that is $M \ll N$. The compressible signal $\mathbf{x}$ may be written as a sparse vector $\mathbf{s} \in \mathbb{C}^N$ (containing mostly zeros) in terms of a transform basis matrix

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s} \tag{B.19}$$

with $\mathbf{\Psi} \in \mathbb{C}^{N \times N}$ [31, 79, 86]. A possible basis matrix $\mathbf{\Psi}$ could be the DFT matrix, as presented in appendix A.3.3. The vector $\mathbf{x}$ is called K-sparse in $\mathbf{\Psi}$, if there are exactly K non-zero elements in $\mathbf{s}$, with $K < M$ [86]. Therefore, compressed sensing involves finding the sparsest vector $\mathbf{s}$ that is consistent with the measurements $\mathbf{y}$

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s}. \tag{B.20}$$

The sparsest solution to this underdetermined system is the $\hat{\mathbf{s}}$ that satisfies the following optimization problem

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{s.t.} \quad \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{s} - \mathbf{y}\|_2 < \varepsilon \tag{B.21}$$

where $\|\cdot\|_0$ denotes the L_0 pseudo-norm (given by the number of nonzero entries in $\mathbf{s}$) and $\|\cdot\|_2$ denotes the L_2 norm [30, 79, 86]. The effect of WGN can be modeled by ε, which should be chosen proportional to the noise magnitude.

Limitations

With CS being a relatively recent topic, the amount of ongoing research and development is tremendous [30, 31, 79, 80, 86, 148]. Thus, no clear implementation or method has yet emerged in the realm of spectrum sensing, complicating practical implementations. There are no reliable perfor-

mance metrics available, which could aid in designing a reliable spectrum sensing system. CS, in general, lacks the stability and robustness of Nyquist-based methods required by SALSAT (CT-03), especially in low SNR environments [86]. Additionally, CS requires specialized RF sampling hardware to be effectively deployed, which SALSAT does not have. CS also relies on solving a computationally complex optimization problem, making a real-time implementation on SALSAT not feasible (CT-06, CT-05). Finally, CS-based spectrum sensing on SALSAT is not a necessity, as the spectrum bands of interest are not "super" wide (see table 1.1). The LMS7002M RF transceiver and the SALSAT computation system can handle the required bandwidth. Nonetheless, it could very well be possible that future missions, with higher bandwidth requirements and more constrained RF resources, could benefit from CS-based spectrum sensing.

C. 3D Spectrograms

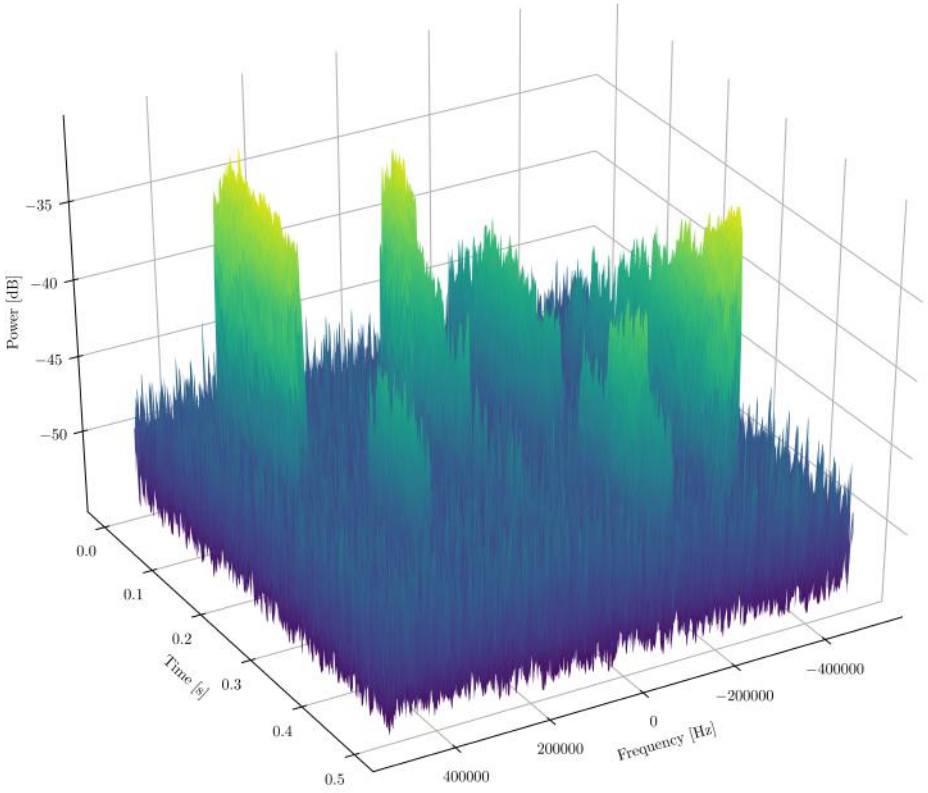

Figure C.1.: 3D spectrogram visualization of the in fig. 4.2 presented wideband signal (made with SpecSens by employing non-causal filtering).

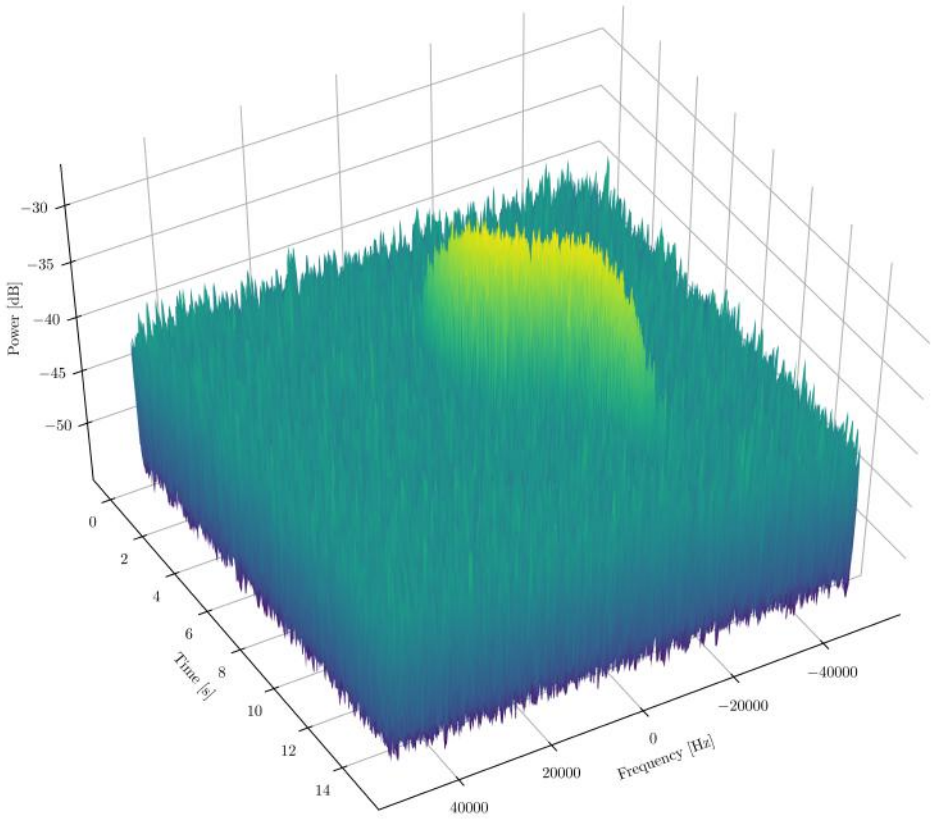

Figure C.2.: 3D spectrogram visualization of the in fig. 4.3 presented Doppler shift signal (made with SpecSens by employing non-causal filtering).

D. Monte Carlo Simulation Algorithm

Algorithm 1 SpecSens Monte Carlo simulation

Input: $Gens, Itrs, Seed, \sigma_w^2, \sigma_s^2, desired\ P_{FA}, \ldots$
Output: P_{FA}, P_D

$\lambda \leftarrow threshold(P_{FA}, \sigma_w^2)$
$Seeds \leftarrow seedSequence(Seed)$
$Results \leftarrow Null$
for $i \leftarrow 1$ to $Gens$ **do**
    $P \leftarrow randomEnvironment(Seeds[i])$
    $W \leftarrow noiseSource(P, \sigma_w^2)$
    $S \leftarrow signalSource(P, \sigma_s^2)$
    for $j \leftarrow 1$ to $Itrs$ **do**
        $w \leftarrow W.get()$
        $s \leftarrow S.get()$
        if $P.random() == True$ **then**
            $x = s + w$
        else
            $x = w$
        $T = detector(x)$
        if $T > \lambda$ **then**
            $Record\ True$
        else
            $Record\ False$
    $Results[i] \leftarrow store\ detection\ results$
$Print\ overall\ Results$
$P_{FA}, P_D \leftarrow overall\ Results$
return P_{FA}, P_D

Figure D.1.: Simplified Monte Carlo simulation pseudocode of the algorithm used by SpecSens.

E. Software Repository Structure

High-level structure of SpecSens package directories:

1. **eigenvalue_detect/** - Eigenvalue detection algorithms based on signal covariance matrices.
2. **energy_detect/** - Energy detector and functions to calculate performance statistics (prob. of false alarm, prob. of detection, etc.).
3. **noise_estimation/** - Noise estimation methods and related utilities.
4. **plot/** - Plotting functions to visualize signals and detection results.
5. **signal/** - Signal and noise generators used mainly for simulations, as well as some utilities that simplify working with signals.
6. **simulation/** - Simulations to evaluate performance statistics.
7. **wideband_detect/** - Wideband detection algorithms (wideband energy detector, wavelet edge detector, etc.).

Selection of Jupyter Notebooks demonstrating and visualizing signals and detection methods:

1. **01_signal_and_noise.ipynb** - Basic overview of complex signal and noise generation, as well as time domain visualization.
2. **02_simple_energy_detector.ipynb** - Simple narrow band time domain energy detector.
3. **03_energy_detector_statistics.ipynb** - Analytical performance statistics overview and comparison (prob. of false alarm, prob. of detection, etc.). Comparison between Chi-square and CLT statistics.
4. **04_energy_detector_simulation.ipynb** - Simulation using simple energy detector and comparison between analytical and numerical statistics.
5. **05_short_time_fourier_transform.ipynb** - Short-time Fourier transform and frequency domain visualization.
6. **06_wideband_signal.ipynb** - Wideband signal generation using signal matrix.
7. **06a_doppler_signal.ipynb** - Doppler signal generation realistically reproducing Doppler shifts.
8. **07_wideband_detect.ipynb** - Short-time Fourier transform based wide band energy detection.
9. **07a_wideband_detect_simulation.ipynb** - Simulation of wide band energy detection.
10. **07b_wideband_detect_doppler.ipynb** - Wideband detection of Doppler signal.
11. **08_edge_detect.ipynb** - Spectrum edge detection using wavelet transforms.
12. **09_variable_band_detect.ipynb** - Variable band wideband energy detection using edge detection.
13. **09a_variable_band_detect_doppler.ipynb** - Variable band wideband energy detection of Doppler signal.

14. **10_noise_estimation_simulation.ipynb** - Energy detection using noise estimation.
15. **11_eigenvalue_detector.ipynb** - Eigenvalue detector based on covariance matrix.
16. **12_eigenvalue_detector_simulation.ipynb** - Simulation of eigenvalue detector.
17. **13_eigenvalue_detector_filter.ipynb** - Eigenvalue detector with FIR bandpass filter.
18. **14_eigenvalue_detector_whitening.ipynb** - Eigenvalue detector with FIR bandpass filter using noise whitening.
19. **15_eigenvalue_simulation_whitening.ipynb** - Simulation of eigenvalue detector with bandpass filter and noise whitening.
20. **16_eigenvalue_noise_estimation.ipynb** - Noise power estimation using covariance matrix eigenvalues.
21. **17_eigenvalue_noise_estimation_simulation.ipynb** - Simulation of wideband energy detection using eigenvalue noise power estimation.
22. **18_noise_estimation_comparison.ipynb** - Comparison simulation of noise estimation techniques.

Bibliography

[1] B.G. Evans and Institution of Electrical Engineers. *Satellite Communication Systems*. EngineeringPro collection. Institution of Engineering and Technology, 1999. ISBN: 9780852968994. URL: https://books.google.de/books?id=T2eD0-pP%5C_0UC.

[2] Anders Kose Nervold et al. "A pathway to small satellite market growth". In: *Advances in Aerospace Science and Technology* 1.1 (2016), pp. 14–20.

[3] Jonathan C McDowell. "The Low Earth Orbit Satellite Population and Impacts of the SpaceX Starlink Constellation". In: *The Astrophysical Journal Letters* 892.2 (2020), p. L36.

[4] *Curbing space debris in the era of mega-constellations*. URL: https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/Curbing_space_debris_in_the_era_of_mega-constellations (visited on 01/13/2021).

[5] Jeff Foust. "SpaceX's space-Internet woes: Despite technical glitches, the company plans to launch the first of nearly 12,000 satellites in 2019". In: *IEEE Spectrum* 56.1 (2018), pp. 50–51.

[6] *Amazon planning 3,236-satellite constellation for internet connectivity*. URL: https://spacenews.com/amazon-planning-3236-satellite-constellation-for-internet-connectivity/ (visited on 01/13/2021).

[7] Martin Buscher. *Investigations on the current and future use of radio frequency allocations for small satellite operations*. Vol. 7. Universitätsverlag der TU Berlin, 2019.

[8] *Committed to connecting the world*. URL: https://www.itu.int/en/Pages/default.aspx (visited on 01/13/2021).

[9] Jens Großhans et al. "SALSAT-An innovative nanosatellite for spectrum analysis based on SDR technology". In: *no. October* (2018), pp. 1–5.

[10] Jens Großhans et al. "SALSA-A novel Spectrum Analyzer board for LEO Satellite Allocations based on SDR technology". In: *2018 AIAA SPACE and Astronautics Forum and Exposition*. 2018, p. 5283.

[11] Jens Großhans et al. "An Approach for Classification of RF Spectrum Data On-Board the Nanosatellite SALSAT". In: *70th International Astronautical Congress (IAC), Washington D.C., United States* (2019).

[12] Jens Großhans et al. "SALSAT: A nanosatellite to analyze the global VHF, UHF and S band spectrum utilization - ready for launch!" In: *71th International Astronautical Congress (IAC) – The CyberSpace Edition* (Oct. 2020).

[13] Jens Großhans et al. "SALSAT: Ready for Launch - Overview of the final flight configuration, mission concept and first flight results". In: *71th International Astronautical Congress (IAC) – The CyberSpace Edition* (Oct. 2020).

[14] Phytec. *phycore-imx-7*. URL: https://www.phytec.eu/product-eu/system-on-modules/phycore-imx-7/ (visited on 12/19/2020).

[15] Lime Microsystems. *LimeSDR*. URL: https://d.com/products/boards/limesdr/ (visited on 12/18/2020).

[16] Intel Corporation. *cyclone-iv*. URL: https://www.intel.de/content/www/de/de/products/programmable/fpga/cyclone-iv.html (visited on 12/19/2020).

[17] Joseph Mitola and Gerald Q Maguire. "Cognitive radio: making software radios more personal". In: *IEEE personal communications* 6.4 (1999), pp. 13–18.

[18] Jacques Palicot. *Radio engineering: From software radio to cognitive radio*. John Wiley & Sons, 2013.

[19] Beibei Wang and KJ Ray Liu. "Advances in cognitive radio networks: A survey". In: *IEEE Journal of selected topics in signal processing* 5.1 (2010), pp. 5–23.

[20] Nguyen Cong Luong et al. "Applications of deep reinforcement learning in communications and networking: A survey". In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3133–3174.

[21] Charles Clancy et al. "Applications of machine learning to cognitive radio networks". In: *IEEE Wireless Communications* 14.4 (2007), pp. 47–52.

[22] Ying-Chang Liang et al. "Cognitive radio networking and communications: An overview". In: *IEEE transactions on vehicular technology* 60.7 (2011), pp. 3386–3407.

[23] Ian F Akyildiz et al. "A survey on spectrum management in cognitive radio networks". In: *IEEE Communications magazine* 46.4 (2008), pp. 40–48.

[24] Sachitha Kusaladharma and Chintha Tellambura. "An overview of cognitive radio networks". In: *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999), pp. 1–17.

[25] Ying-Chang Liang. *Dynamic Spectrum Management: From Cognitive Radio to Blockchain and Artificial Intelligence*. Springer Nature, 2020.

[26] Ezio Biglieri et al. *Principles of Cognitive Radio*. Cambridge University Press, 2012. DOI: 10.1017/CBO9781139236850.

[27] Alexander M Wyglinski, Maziar Nekovee, and Thomas Hou. *Cognitive radio communications and networks: principles and practice*. Academic Press, 2009.

[28] Zhi Quan et al. "Collaborative wideband sensing for cognitive radios". In: *IEEE Signal Processing Magazine* 25.6 (2008), pp. 60–73.

[29] Tevfik Yucek and Huseyin Arslan. "A survey of spectrum sensing algorithms for cognitive radio applications". In: *IEEE communications surveys & tutorials* 11.1 (2009), pp. 116–130.

[30] Youness Arjoune and Naima Kaabouch. "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions". In: *Sensors* 19.1 (2019), p. 126.

[31] Erik Axell et al. "Spectrum sensing for cognitive radio: State-of-the-art and recent advances". In: *IEEE signal processing magazine* 29.3 (2012), pp. 101–116.

[32] Simon Haykin, David J Thomson, and Jeffrey H Reed. "Spectrum sensing for cognitive radio". In: *Proceedings of the IEEE* 97.5 (2009), pp. 849–877.

[33] Yonghong Zeng et al. "A review on spectrum sensing for cognitive radio: challenges and solutions". In: *EURASIP Journal on Advances in signal Processing* 2010 (2010), pp. 1–15.

[34] Guodong Zhao. *Advanced sensing techniques for cognitive radio*. Springer, 2017.

[35] Amir Ghasemi and Elvino S Sousa. "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs". In: *IEEE Communications magazine* 46.4 (2008), pp. 32–39.

[36] Konstantinos Pelechrinis et al. "Cognitive radio networks: realistic or not?" In: *ACM SIGCOMM Computer Communication Review* 43.2 (2013), pp. 44–51.

[37] Marko Höyhtyä et al. "Application of cognitive radio techniques to satellite communication". In: *2012 IEEE International Symposium on Dynamic Spectrum Access Networks*. IEEE. 2012, pp. 540–551.

[38] Ezio Biglieri. "An overview of cognitive radio for satellite communications". In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. IEEE. 2012, pp. 1–3.

[39] European commission. *COgnitive RAdio for SATellite Communications*. 2017. URL: `https://cordis.europa.eu/project/id/316779` (visited on 12/16/2020).

[40] Konstantinos Liolis et al. "Cognitive radio scenarios for satellite communications: The CoRaSat approach". In: *2013 Future Network & Mobile Summit*. IEEE. 2013, pp. 1–10.

[41] Symeon Chatzinotas et al. "Cognitive approaches to enhance spectrum availability for satellite systems". In: *International Journal of Satellite Communications and Networking* 35.5 (2017), pp. 407–442.

[42] Alessandro Vanelli-Coralli et al. "Cognitive radio scenarios for satellite communications: the corasat project". In: *Cooperative and Cognitive Satellite Systems*. Elsevier, 2015, pp. 303–336.

[43] WWTG Peterson, T Birdsall, and We Fox. "The theory of signal detectability". In: *Transactions of the IRE professional group on information theory* 4.4 (1954), pp. 171–212.

[44] H.L. Van Trees. *Detection, Estimation, and Modulation Theory: Detection, estimation, and linear modulation theory*. Detection, Estimation, and Modulation Theory. Wiley, 2001. ISBN: 9780471095170. URL: `https://books.google.de/books?id=cQwfAQAAIAAJ`.

[45] "IEEE Standard - Information Technology-Telecommunications and information exchange between systems-Wireless Regional Area Networks-Specific requirements-Part 22: Cognitive Wireless RAN MAC and PHY specifications: Policies and Procedures for Operation in the Bands that Allow Spectrum Sharing where the Communications Devices May Opportunistically Operate in the Spectrum of Primary Service". In: *IEEE Std 802.22-2019 (Revision of IEEE Std 802.22-2011)* (2020), pp. 1–1465. DOI: `10.1109/IEEESTD.2020.9086951`.

[46] Carlos Cordeiro et al. "IEEE 802.22: the first worldwide wireless standard based on cognitive radios". In: *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* Ieee. 2005, pp. 328–337.

[47] Carl R Stevenson et al. "IEEE 802.22: The first cognitive radio wireless regional area network standard". In: *IEEE communications magazine* 47.1 (2009), pp. 130–138.

[48] Stephen J Shellhammer et al. "Spectrum sensing in IEEE 802.22". In: *IAPR Wksp. Cognitive Info. Processing* (2008), pp. 9–10.

[49] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Television White Spaces (TVWS) Operation". In: *IEEE Std 802.11af-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, IEEE Std 802.11ad-2012, and IEEE Std 802.11ac-2013)* (2014), pp. 1–198. DOI: `10.1109/IEEESTD.2014.6744566`.

[50] Adriana B Flores et al. "IEEE 802.11 af: A standard for TV white space spectrum sharing". In: *IEEE Communications Magazine* 51.10 (2013), pp. 92–100.

[51] T Dhope and Dina Simunic. "Spectrum sensing algorithm for cognitive radio networks for dynamic spectrum access for IEEE 802.11 af standard". In: *IJRRWSN* 2.1 (2012).

[52] Tomaž Šolc, Mihael Mohorčič, and Carolina Fortuna. "A methodology for experimental evaluation of signal detection methods in spectrum sensing". In: *PloS one* 13.6 (2018), e0199550.

[53] Vincent Le Nir and B Scheers. "Cogwave: Open-source software framework for cognitive radio waveform design". In: *IST-123 Symposium on Cognitive Radio and Future Networks, The Hague, The Nederlands*. 2014.

[54] ECSS Secretariat. *Space Engineering Verification*. Tech. rep. Report ECSS-E-10-02A. ECSS - European Cooperation for Space Standardization, 1998.

[55] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[56] John G.. Proakis and Masoud Salehi. *Digital communications*. McGraw-Hill., 2008.

[57] Cheng Yu Andy Shang. "Linear transceivers for MIMO relays". In: (2014).

[58] Kevin McClaning and Tom Vito. *Radio receiver design*. Noble Publishing, 2000.

[59] Kun Il Park and Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018.

[60] S.M. Kay. *Fundamentals of Statistical Signal Processing: Detection theory*. Fundamentals of Statistical Si. Prentice-Hall PTR, 1998. ISBN: 9780133457117. URL: https://books.google.de/books?id=vA9LAQAAIAAJ.

[61] Yonghong Zeng and Y-C Liang. "Eigenvalue-based spectrum sensing algorithms for cognitive radio". In: *IEEE transactions on communications* 57.6 (2009), pp. 1784–1793.

[62] Lu Wei, Olav Tirkkonen, and Ying-Chang Liang. "Multi-source signal detection with arbitrary noise covariance". In: *IEEE Transactions on Signal Processing* 62.22 (2014), pp. 5907–5918.

[63] Ayse Kortun et al. "On the performance of eigenvalue-based cooperative spectrum sensing for cognitive radio". In: *IEEE Journal of Selected Topics in Signal Processing* 5.1 (2010), pp. 49–55.

[64] R. Hoffmann and M. Wolff. *Intelligente Signalverarbeitung 1: Signalanalyse*. Online access with purchase: Springer. Springer Berlin Heidelberg, 2014. ISBN: 9783662453230. URL: https://books.google.de/books?id=QHC1BQAAQBAJ.

[65] H.V. Poor. *An Introduction to Signal Detection and Estimation*. Springer Texts in Electrical Engineering. Springer New York, 2013. ISBN: 9781475723410. URL: https://books.google.de/books?id=1afqBwAAQBAJ.

[66] R. Hoffmann and M. Wolff. *Intelligente Signalverarbeitung 2: Signalerkennung*. Springer Berlin Heidelberg, 2015. ISBN: 9783662467268. URL: https://books.google.de/books?id=ogTSBwAAQBAJ.

[67] Wikipedia. *Receiver operating characteristic — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Receiver%20operating%20characteristic&oldid=1008201512. [Online; accessed 6-March-2021]. 2021.

[68] H. Urkowitz. "Energy detection of unknown deterministic signals". In: *Proceedings of the IEEE* 55.4 (1967), pp. 523–531. DOI: 10.1109/PROC.1967.5573.

[69] L. Rugini, P. Banelli, and G. Leus. "Small Sample Size Performance of the Energy Detector". In: *IEEE Communications Letters* 17.9 (2013), pp. 1814–1817. DOI: 10.1109/LCOMM.2013.080813.131399.

[70] Saman Atapattu, Chintha Tellambura, and Hai Jiang. *Energy detection for spectrum sensing in cognitive radio*. Vol. 6. Springer, 2014. ISBN: 978-1-4939-0494-5.

[71] E. Axell, G. Leus, and E. G. Larsson. "Overview of spectrum sensing for cognitive radio". In: *2010 2nd International Workshop on Cognitive Information Processing*. June 2010, pp. 322–327. DOI: 10.1109/CIP.2010.5604136.

[72] Martijn Franciscus Arts and Universitätsprofessorin Dr-Ing Anja Klein. "Eigenvalue-Based Spectrum Sensing for Cognitive Radio". In: (2017).

[73] Yonghong Zeng and Ying-Chang Liang. "Spectrum-sensing algorithms for cognitive radio based on statistical covariances". In: *IEEE transactions on Vehicular Technology* 58.4 (2008), pp. 1804–1815.

[74] Yonghong Zeng and Ying-Chang Liang. "Maximum-minimum eigenvalue detection for cognitive radio". In: *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2007, pp. 1–5.

[75] Pu Wang et al. "Multiantenna-assisted spectrum sensing for cognitive radio". In: *IEEE transactions on vehicular technology* 59.4 (2009), pp. 1791–1800.

[76] Pascal Bianchi et al. "Performance of statistical tests for single-source detection using random matrix theory". In: *IEEE Transactions on Information theory* 57.4 (2011), pp. 2400–2419.

[77] Fuhui Zhou and Norman C Beaulieu. "An improved and more accurate expression for a PDF related to eigenvalue-based spectrum sensing". In: *IEEE Systems Journal* 13.2 (2018), pp. 1320–1323.

[78] Chang Liu et al. "Maximum eigenvalue-based goodness-of-fit detection for spectrum sensing in cognitive radio". In: *IEEE Transactions on Vehicular Technology* 68.8 (2019), pp. 7747–7760.

[79] Bashar I Ahmad. "A Survey of Wideband Spectrum Sensing Algorithms for Cognitive Radio Networks and Sub-Nyquist Approaches". In: *arXiv preprint arXiv:2001.02574* (2020).

[80] Hongjian Sun et al. "Wideband spectrum sensing for cognitive radio networks: a survey". In: *IEEE Wireless Communications* 20.2 (2013), pp. 74–81.

[81] Ghaith Hattab and Mohamed Ibnkahla. "Multiband spectrum access: Great promises for future cognitive radio networks". In: *Proceedings of the IEEE* 102.3 (2014), pp. 282–306.

[82] Zhi Quan et al. "Optimal multiband joint detection for spectrum sensing in cognitive radio networks". In: *IEEE transactions on signal processing* 57.3 (2008), pp. 1128–1140.

[83] Zhi Quan et al. "Wideband spectrum sensing in cognitive radio networks". In: *2008 IEEE international conference on communications*. IEEE. 2008, pp. 901–906.

[84] Tsung-Wei Chiang, Jung-Mao Lin, and Hsi-Pin Ma. "Optimal detector for multitaper spectrum estimator in cognitive radios". In: *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE. 2009, pp. 1–6.

[85] Petre Stoica, Randolph L Moses, et al. "Spectral analysis of signals". In: (2005).

[86] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690.

[87] Behtash Babadi and Emery N Brown. "A review of multitaper spectral analysis". In: *IEEE Transactions on Biomedical Engineering* 61.5 (2014), pp. 1555–1564.

[88] Behrouz Farhang-Boroujeny. "Filter bank spectrum sensing for cognitive radios". In: *IEEE Transactions on signal processing* 56.5 (2008), pp. 1801–1811.

[89] Qi Tu Zhang. "Theoretical performance and thresholds of the multitaper method for spectrum sensing". In: *IEEE Transactions on vehicular technology* 60.5 (2011), pp. 2128–2138.

[90] Fei Hu and Sunil Kumar. *Multimedia over cognitive radio networks: algorithms, protocols, and experiments*. CRC Press, 2014.

[91] Bo Duan et al. "Floating-point mixed-radix FFT core generation for FPGA and comparison with GPU and CPU". In: *2011 International Conference on Field-Programmable Technology*. IEEE. 2011, pp. 1–6.

[92] Alexander Maaß. "Integration einer FFT in FPGA-Designs und Optimierung für den Einsatz in der Raumfahrt". MA thesis. Technische Universität Berlin, 2018.

[93] Rahul Tandra and Anant Sahai. "SNR walls for signal detection". In: *IEEE Journal of selected topics in Signal Processing* 2.1 (2008), pp. 4–17.

[94] Andrea Mariani, Andrea Giorgetti, and Marco Chiani. "SNR wall for energy detection with noise power estimation". In: *2011 IEEE international conference on communications (ICC)*. IEEE. 2011, pp. 1–6.

[95] DD Ariananda, MK Lakshmanan, and H Nikookar. "A survey on spectrum sensing techniques for cognitive radio". In: *2009 Second International Workshop on Cognitive Radio and Advanced Spectrum Management*. IEEE. 2009, pp. 74–79.

[96] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[97] Stephane Mallat and Wen Liang Hwang. "Singularity detection and processing with wavelets". In: *IEEE transactions on information theory* 38.2 (1992), pp. 617–643.

[98] Zhi Tian and Georgios B Giannakis. "A wavelet approach to wideband spectrum sensing for cognitive radios". In: *2006 1st international conference on cognitive radio oriented wireless networks and communications*. IEEE. 2006, pp. 1–5.

[99] Yonghong Zeng, Ying-Chang Liang, and Meng Wah Chia. "Edge based wideband sensing for cognitive radio: Algorithm and performance evaluation". In: *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE. 2011, pp. 538–544.

[100] Marc Alexa. *Wissenschaftliches Rechnen*. Fachgebiet Computer Graphics, Technische Universität Berlin, 2020.

[101] Andrea Mariani, Andrea Giorgetti, and Marco Chiani. "Effects of noise power estimation on energy detection for cognitive radio applications". In: *IEEE Transactions on Communications* 59.12 (2011), pp. 3410–3420.

[102] Andrea Mariani. "Spectrum sensing algorithms for cognitive radio applications". In: (2013).

[103] Mohamed Hamid, Niclas Bjorsell, and Slimane Ben Slimane. "Sample covariance matrix eigenvalues based blind SNR estimation". In: *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. IEEE. 2014, pp. 718–722.

[104] Raj Rao Nadakuditi and Alan Edelman. "Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples". In: *IEEE Transactions on Signal Processing* 56.7 (2008), pp. 2625–2638.

[105] Mati Wax and Thomas Kailath. "Detection of signals by information theoretic criteria". In: *IEEE Transactions on acoustics, speech, and signal processing* 33.2 (1985), pp. 387–392.

[106] Jorma Rissanen. "Modeling by shortest data description". In: *Automatica* 14.5 (1978), pp. 465–471.

[107] LA Pastur and VA Martchenko. "The distribution of eigenvalues in certain sets of random matrices". In: *Math. USSR-Sbornik* 1 (1967), pp. 457–483.

[108] Kenneth W Wachter et al. "The strong limits of random matrix spectra for sample matrices of independent elements". In: *The Annals of Probability* 6.1 (1978), pp. 1–18.

[109] Andrea Mariani, Andrea Giorgetti, and Marco Chiani. "Energy detector design for cognitive radio applications". In: *2010 International Waveform Diversity and Design Conference*. IEEE. 2010, pp. 000053–000057.

[110] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[111] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[112] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[113] Gregory R. Lee et al. "PyWavelets: A Python package for wavelet analysis". In: *Journal of Open Source Software* 4.36 (2019), p. 1237. DOI: 10.21105/joss.01237. URL: https://doi.org/10.21105/joss.01237.

[114] Yao-Yuan Mao. *Providing the Tracy-Widom distribution functions for beta = 1, 2, or 4 in Python*. 2020. URL: https://github.com/yymao/TracyWidom/ (visited on 12/22/2020).

[115] Fabian Peddinghaus. *SpecSens: Spectrum sensing for the SALSAT*. 2020. URL: https://git.tu-berlin.de/rftcom/students/spectrum-sensing.

[116] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[117] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[118] Lutz Prechelt. "Are scripting languages any good? A validation of Perl, Python, Rexx, and Tcl against C, C++, and Java." In: *Adv. Comput.* 57 (2003), pp. 205–270.

[119] Chris Clanton, Mark Kenkel, and Yang Tang. "Wireless microphone signal simulation method". In: *IEEE 802.22-07/0124r0* (2007).

[120] Daniel Romero and Roberto Lopez-Valcarce. "Spectrum sensing for wireless microphone signals using multiple antennas". In: *IEEE Transactions on Vehicular Technology* 63.9 (2014), pp. 4395–4407.

[121] Arnaud Doucet and Xiaodong Wang. "Monte Carlo methods for signal processing: a review in the statistical signal processing context". In: *IEEE Signal Processing Magazine* 22.6 (2005), pp. 152–170.

[122] Peter Welch. "The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms". In: *IEEE Transactions on audio and electroacoustics* 15.2 (1967), pp. 70–73.

[123] Youness Arjoune et al. "Spectrum sensing: Enhanced energy detection technique based on noise measurement". In: *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*. IEEE. 2018, pp. 828–834.

[124] Norman P Jouppi et al. "In-datacenter performance analysis of a tensor processing unit". In: *Proceedings of the 44th annual international symposium on computer architecture*. 2017, pp. 1–12.

[125] A Castillo-Atoche, D Torres-Roman, and Yuriy Shkvarko. "Towards real time implementation of reconstructive signal processing algorithms using systolic arrays coprocessors". In: *Journal of Systems Architecture* 56.8 (2010), pp. 327–339.

[126] Kari Kokkinen et al. "FPGA implementation of autocorrelation-based feature detector for cognitive radio". In: *2009 NORCHIP*. IEEE. 2009, pp. 1–4.

[127] Kari Kokkinen et al. “On the implementation of autocorrelation-based feature detector”. In: *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*. IEEE. 2010, pp. 1–4.

[128] Jan Buchholz et al. “FPGA implementation of a 32x32 autocorrelator array for analysis of fast image series”. In: *Optics express* 20.16 (2012), pp. 17767–17782.

[129] Mahendra Vucha and Arvind Rajawat. “Design and FPGA implementation of systolic array architecture for matrix multiplication”. In: *International Journal of Computer Applications* 26.3 (2011), pp. 18–22.

[130] Aziz Ahmedsaid, Abbes Amira, and Ahmed Bouridane. “Improved SVD systolic array and implementation on FPGA”. In: *Proceedings. 2003 IEEE International Conference on Field-Programmable Technology (FPT)(IEEE Cat. No. 03EX798)*. IEEE. 2003, pp. 35–42.

[131] Xiaojun Wang and Miriam Leeser. “A truly two-dimensional systolic array FPGA implementation of QR decomposition”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 9.1 (2009), pp. 1–17.

[132] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993. ISBN: 0133457117.

[133] A Bejan. “Largest eigenvalues and sample covariance matrices. tracy-widom and painleve ii: computational aspects and realization in s-plus with applications”. In: *Preprint: http://www. vitrum. md/andrew/MScWrwck/TWinSplus. pdf* (2005).

[134] Gaëtan Borot and Céline Nadal. “Right tail asymptotic expansion of tracy–widom beta laws”. In: *Random Matrices: Theory and Applications* 1.03 (2012), p. 1250006.

[135] Wikipedia. *Window function — Wikipedia, The Free Encyclopedia*. `http://en.wikipedia.org/w/index.php?title=Window%20function&oldid=1010259605`. [Online; accessed 2-March-2021]. 2021.

[136] Clemens Gühmann. *Messdatenverarbeitung*. Fachgebiet Elektronische Mess- und Diagnosetechnik, Technische Universität Berlin, 2011.

[137] Tobit Flatscher. *Schematic Fourier Analysis*. 2019. URL: `https://tex.stackexchange.com/a/513431` (visited on 12/20/2020).

[138] James W Cooley and John W Tukey. “An algorithm for the machine calculation of complex Fourier series”. In: *Mathematics of computation* 19.90 (1965), pp. 297–301.

[139] Paul N Swarztrauber. “Vectorizing the ffts”. In: *Parallel computations*. Elsevier, 1982, pp. 51–83.

[140] Matteo Frigo and Steven G Johnson. “The design and implementation of FFTW3”. In: *Proceedings of the IEEE* 93.2 (2005), pp. 216–231.

[141] Henry Gomersall. *pyFFTW*. 2016. DOI: `10.5281/zenodo.59508`. URL: `https://doi.org/10.5281/zenodo.59508`.

[142] Gerhard Heinzel, Albrecht Rüdiger, and Roland Schilling. “Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new at-top windows”. In: (2002).

[143] Siemens. *Window Types: Hanning, Flattop, Uniform, Tukey, and Exponential*. 2020. URL: `https://community.sw.siemens.com/s/article/window-types-hanning-flattop-uniform-tukey-and-exponential` (visited on 11/15/2020).

[144] Wikipedia. *Spectrogram — Wikipedia, The Free Encyclopedia*. `http://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=1001441730`. [Online; accessed 22-February-2021]. 2021.

[145] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.

[146] Martin Vetterli and Cormac Herley. "Wavelets and filter banks: Theory and design". In: *IEEE transactions on signal processing* 40.ARTICLE (1992), pp. 2207–2232.

[147] Imran Touqir, Muhammad Saleem, and Adil Masood Siddiqui. "Wavelet based boundary detection". In: *Proceedings of the 5th WSEAS international conference on System science and simulation in engineering*. Citeseer. 2006, pp. 475–480.

[148] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge university press, 2012.