Semu Mäkinen

Mobile Spectrum Sensing for Cognitive Radio

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 14.11.2014

Thesis supervisor:

Prof. Jussi Ryynänen

Thesis advisor:

D.Sc. Marko Kosunen

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Semu Mäkinen

Title: Mobile Spectrum Sensing for Cognitive Radio

Date: 14.11.2014 Language: English Number of pages: 8+81

Department of Micro- and Nano-Sciences

Professorship: Electronic Circuit Design Code: S-87

Supervisor: Prof. Jussi Ryynänen

Advisor: D.Sc. Marko Kosunen

Radio spectrum is becoming a scarce resource as increasing number of wireless devices attempt to access it. As a solution to this issue, spectrum sensing based cognitive radios have been proposed. However, significant part of them are mobile, consumer-grade devices that have strict price and form-factor limitations.

This works aims to address the issues related to mobile spectrum sensing by characterizing the non-idealities with a spectrum sensor prototype. Two efficient sensing algorithms and a coarse-fine controller, which aims to minimize the energy consumption and run time of an individual sensor, are implemented on an FPGA. Functionality of the implementations is verified by laboratory and field measurements. Finally, a spatial interpolation method, Kriging, is applied to the non-ideal measurement data to create a uniform radio environment map.

Keywords: Cognitive radio, Spectrum sensing, Mobile device, Non-ideality, Energy efficiency, Spatial interpolation

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Semu Mäkinen

Työn nimi: Spektrin Havainnointi Kognitiivisissa Matkaviestinlaitteissa

Päivämäärä: 14.11.2014 Kieli: Englanti Sivumäärä: 8+81

Mikro- ja nanotekniikan laitos

Professuuri: Piirisuunnittelu Koodi: S-87

Valvoja: Prof. Jussi Ryynänen

Ohjaaja: TkT Marko Kosunen

Langattomien laitteiden yleistyminen kasvattaa radiospektrin käyttöastetta ylärajaa kohti. Ratkaisuksi ongelmaan on kehitetty spektrin havainnointiin perustuvat kognitiiviset radiot. Näistä valtaosa on kuitenkin kuluttajatason matkaviestinlaitteita, joilla on tiukat rajoitteet muun muassa hinnan ja fyysisen rakenteen suhteen.

Tässä työssä perehdytään spektrin havainnoinnin haasteisiin tutkimalla havainnoinnin epäideaalisuuksia spektrisensoriprototyypillä. Työssä on toteutettu FPGA:lle kaksi energiatehokasta havainnointialgoritmia sekä karkea-herkkä -ohjain, joka pyrkii minimoimaan yksittäisen spektrisensorin energiakulutusta sekä havainnointiaikaa. Toteutettujen algoritmien toiminta ja suorituskyky verifioidaan laboratorio- sekä kenttämittauksilla. Lopuksi esitellään avaruudellinen interpolaatiomenetelmä, Kriging, jota sovelletaan epäideaaliseen kenttämittausdataan kattavan radiopeitekartan luomiseksi.

Avainsanat: Kognitiivinen radio, Spektrin havainnointi, Matkaviestinlaite, Epäideaalisuus, Energiatehokkuus, Avaruudellinen interpolaatio

Preface

This work has been done in Aalto University's department of Micro- and Nano-Sciences in Electronic Circuit Design unit as a part of ENCOR2-project (Enabling Cognitive Radios). I would like to express my gratitude towards the whole department for giving me a chance to work there, and towards TEKES for funding the project.

I would like to thank my advisor Marko Kosunen and supervisor Jussi Ryynänen for giving helpful advises and guiding me during the work. Finally, I would like to thank my family, friends and collegues for supporting me and occasionally helping me get my mind off the thesis.

Otaniemi, 12.12.2014

Semu J. M. Mäkinen

Contents

Symbols and Abbreviations

α	Angle
γ_i	Weight for measured values at location i
$\hat{\gamma}(\cdot)$	Semivariance
ϵ	Estimation error
η	Threshold
τ	Lag parameter
$\varphi_x[n]$	Argument of x[n]
χ_2^2	Chi-square distribution
$\mathcal{H}_0$	Null hypothesis
$\mathcal{H}_1$	Alternative hypothesis
$Pr(\cdot)$	Probability
R_x^α	Autocorrelation estimate
$s(t)$	Noise
$S(\cdot)$	Spatial sign function
$\mathcal{T}$	Test statistics
$w(t)$	Primary signal
$x(t)$	Received signal
$Z(\cdot)$	Measured value at a location
ADC	Analog-to-Digital Converter
ADDT	Angular Domain Detector
AGC	Automatic Gain Control
AIC	Akaike Information Criterion
AWGN	Additive White Gaussian Noise
BB	Baseband
CAF	Cyclic Autocorrelation Function
CDF	Cumulative Distribution Function
CFAR	Constant False Alarm Rate
CF	Coarse-Fine
CFD	Cyclostationary Feature Detector
CORDIC	COordinate Rotation DIgital Computer
CP	Cyclic Prefix
CR	Cognitive Radio
CS	Cooperative Sensing
CYC-FFT	FFT-Based Cyclostationary feature detector
DFT	Discrete Fourier Transform
DVB-T	Digital Video Broadcasting - Terrestial
EDT	Energy Detector
FAR	False Alarm Rate
FC	Fusion Center
FCC	Federal Communications Committee
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FNR	False Negative Rate

FPR	False Positive Rate
FPGA	Field-Programmable Gate Array
FSA	Fixed Spectrum Access
GFT	Goodness of Fit Test
HW	Hardware
IDW	Inverse Distance Weighted interpolation
IEEE	Institute of Electrical and Electronics Engineers
IIP	Input Intercept Point
IM	Intermodulation
ITC	Information Theory Criterion
LNA	Low-Noise Amplifier
LO	Local Oscillator
LRT	Likelihood Ratio Test
LS	Least Squares
LUT	Look-Up Table
NP	Neyman-Pearson
OFDM	Orthogonal Frequency Division Multiplexing
PCB	Printed Circuit Board
PGA	Programmable Gain Amplifier
PSD	Power Spectral Density
PU	Primary User
QoS	Quality of Service
REML	Residual Maximum Likelihood
RF	Radio Frequency
Rx	Receiver, Reception
ROC	Receiver Operating Characteristics
RSSI	Received Signal Strength Indicator
SDR	Software Defined Radio
SNR	Signal-to-Noise Ratio
SSCC	Spatial Sign Cyclic Correlation
SSF	Spatial Sign Function
SU	Secondary User
TNR	True Negative Rate
TPR	True Positive Rate
TS	Test Statistics
Tx	Transmitter, Transmission
USB	Universal Serial Bus
VCO	Voltage Controlled Oscillator
VHDL	VHSIC Hardware Description Language
VHSIC	Very High-Speed Integrated Circuit
WLAN	Wireless Local Area Network
WLS	Weighted Least Squares

1 Introduction

The trend seems to be that every gadget and application is eventually becoming wireless. Having no wires may enhance the physical look of the product, ease up the installation process but especially improve the mobility of the device. On the other hand, being wireless also means that all communication needs to be done through the radio spectrum. When all the wireless systems are allocated a specific frequency band on which they can operate, the spectrum will eventually become a scarce resource. However, the spectrum may often remain severely under-utilized since the users of the spectrum do not normally transmit at all times and especially not in all locations simultaneously.

The proposed solution to the issue is cognitive radio. The profound idea of a cognitive network is to align the communication on a licensed, locally underutilized frequency band without interfering the primary user. The key enabler for the technology is spectrum sensing, which is currently one of the most widely researched topic in the field. However, significant part of the research is focused on the theoretical operation of the algorithms and implementations on ideal, or at least lab-grade conditions. A simple fact is often omitted: real-world, consumer-grade hardware, as well as the radio environment in any practical scenario, may not be even close to ideal.

The aim of this thesis is to test and verify spectrum sensing techniques in realistic conditions. Sensing methods will be introduced and their implementations on mobile hardware will be presented. Various related aspects, including non-idealities and energy efficiency considerations, will be discussed from mobile equipment point of view, in order to give an insight on what to expect from the implementations and the measurements. The implementation of two energy efficient detection algorithms, along with methods for even further improving their energy efficiency and compensating the issues caused by non-ideal hardware will be presented. Their performance will be evaluated by laboratory and field measurements, using non-ideal hardware that is specifically designed for mobile usage.

The thesis is structured as follows: Chapter 2 introduces the general operating principle of a cognitive radio and gives a theoretical background for spectrum sensing methods and algorithms. The sensing methods will be discussed in a wider sense and those that are relevant for this work are described in detail. In addition, two non-local spectrum sensing methods, cooperative sensing and spatial interpolation, will be introduced.

Chapter 3 focuses on the more practical side of the spectrum sensing. Non-idealities and design challenges that are often present in consumer-grade hardware will be discussed. In addition to the hardware, the environment and transmitter dependent conditions and their effects on spectrum sensing will be analyzed. Furthermore, this chapter discusses the energy efficiency and introduces a two-stage sensing scheme that can ultimately increase the efficiency of a single sensor node considering the overall sensing time and energy consumption.

All the implementations of the work are described in Chapter 4. Two sensing algorithms, energy detector and angular domain feature detector, are re-implemented

based on earlier work. A detector controller is fully implemented to handle the described coarse-fine scheme. Furthermore, the front-end controls are improved to compensate for certain hardware non-idealities.

The functionality of the implementations is verified in Chapter 5 by laboratory and field measurements. Additionally, the non-idealities are further analyzed and characterized from measurement results. Finally, a use-case of the aforementioned spatial interpolation method is presented by the means of MATLAB simulation, using real field measurement data.

The spectrum sensing algorithm implementations presented in this thesis are based on earlier work. The base for energy detector, as well as the prototype platform, is originally designed by Nokia and further developed in Aalto University. The original feature detector implementation is presented in another master's thesis [1] and theoretical base work for the spatial interpolation can be found in [2].

Publications on the same topics and hardware describe the prototype platform [3], a feature detector implementation [4], mobile measurements with non-idealities [5] and cooperative sensing trials [6].

2 Cognitive Radio

Wireless devices that communicate with one another using electromagnetic radio spectrum suffer from interference in a way or another. One consequence of this phenomenon is that such devices cannot usually operate on the same frequency band. The traditional approach to alleviate this issue has been to assign specific frequency bands of the radio spectrum to specific licensed users, such as TV broadcasters. The licensed users are often referred as Primary Users (PU) of the network, and the Secondary Users (SU), such as certain mobile devices, are not allowed to transmit on the band allocated to them. This policy is called Fixed Spectrum Access (FSA) [7].

Fast evolution of wireless communication systems has led to a situation in which the radio spectrum is almost fully allocated to PU's. This is called the spectrum scarcity problem; the radio spectrum is becoming a limited resource and thus cannot support the increasing number of wireless devices infinitely [7, 8]. However, recent studies and measurements in various countries show that significant part of the radio spectrum is inefficiently used, utilization being in the range of 5% to 50% [9–11]. Therefore it could be concluded that instead of the spectral scarcity, the inefficient utilization of the radio spectrum should be of the main concern in today's radio communication policy.

The emerging radio technologies are intended to allow the utilization of locally unoccupied frequency bands by the secondary users without interfering to the primary users. Therefore, the communication capacity of secondary users is strongly dependent on reliable and efficient detection of primary users and spectral opportunities by the means of *spectrum sensing* or geolocation-based databases. Software-Defined Radios (SDR) that align their communication based on the detection of primary users are commonly referred as **Cognitive Radios** (CR) [12, 13].

A definition of cognitive radio that is widely referred to in today's literature is presented by Haykin in [9]: *Cognitive Radio is defined as an intelligent wireless communication system that is aware of its surrounding environment and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g. transmit power, carrier frequency, and modulation strategy) in real time with two primary objectives in mind: One: highly reliable communication whenever and wherever needed, Second: efficient utilization of the radio spectrum.*

2.1 Spectrum Awareness

Efficient detection of spectral communication resources is the key enabler for systems intended to utilize opportunistic communication. For such systems, the main objective is to protect the primary users, which requires *spectrum awareness* from behalf of the secondary users. One possible approach is to align the secondary transmission in close coordination with the primary system in such a way that the interference of the secondary user is either orthogonal to the primary signals [14] or counteracted to the direction of them [15]. However, this method embodies several performance

limitations [16] and is thus a subject to becoming obsolete. Another possibility is to use the primary band opportunistically based on **spectrum sensing**. This approach is often preferable due to its flexibility and fairness in the sense that the responsibility of the spectrum management is entirely on the secondary system [17]. Various aspects of the opportunistic approach will be discussed in detail throughout this thesis. Figure 1 depicts a generalized operation principle of an opportunistic cognitive radio system.

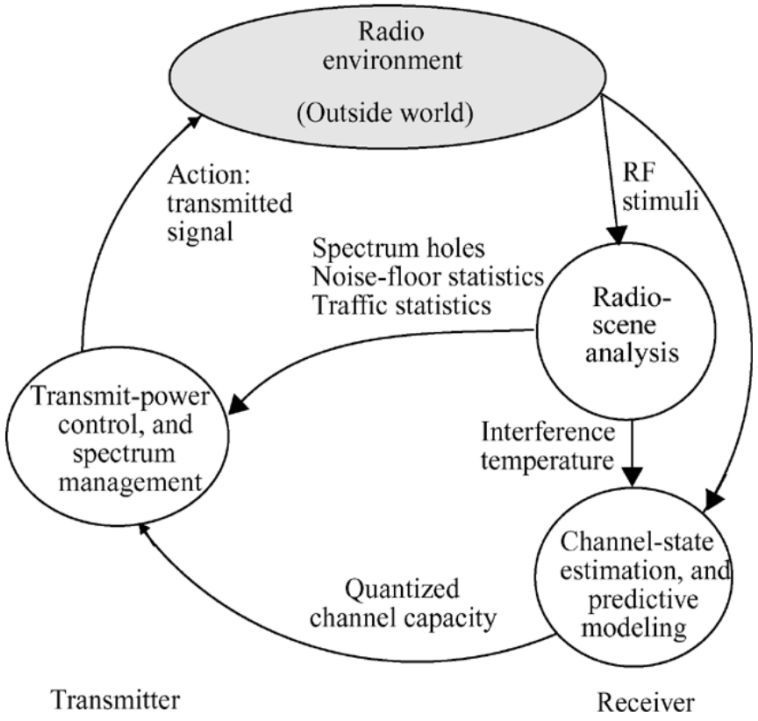

Figure 1: Basic cognitive cycle as presented in [9]

Since access to occupancy databases or coordination systems cannot be always guaranteed, and data is needed to establish and maintain such databases, spectrum sensing is evidently the most crucial component for gaining spectrum awareness and ultimately forming the cognitive network [18]. The profound objective of spectrum sensing is to find local spectrum opportunities, or spectrum holes, that can be utilized as communication bands for the secondary system [19]. [20] defines the spectrum opportunity as *a band of frequencies that are not being used by the primary user of that band at a particular time in a particular geographic area*. For example, in Terrestrial Digital Video Broadcasting (DVB-T) system, spectrum holes are the TV channels that are neither utilized by a transmitter of a certain area, nor blocked by an interferer. An overview and classification of methods for gaining spectrum awareness is given in the next section.

2.2 Spectrum Sensing Methods

Spectrum sensing is a very broad field of study and can be categorized by several criteria. Based on the sensing target, spectrum sensing methods can be classified into *direct and indirect spectrum sensing*. The primary system basically consists of transmitters (Tx) and receivers (Rx), the latter being the end users that need to be protected from the cognitive users. The goal of direct spectrum sensing is to detect primary receivers that are in the coverage of the cognitive Tx. Another approach is to detect the presence or absence of primary Tx signals, which is called indirect spectrum sensing [7].

On the other hand, terms *direct and indirect method* can be used for classifying spectrum sensing based on the operating principle of the sensing algorithm. Direct method means estimating the power spectrum of the received signal in the frequency domain, whereas indirect method applies autocorrelation function to the signal in time domain and estimates powers spectrum from the Discrete Fourier Transform (DFT) of the autocorrelation value [21].

Another way to categorize spectrum sensing algorithms is to classify them into *blind, semi-blind and non-blind schemes*, based on the information they require on the primary system a priori [22]. These schemes, along with sensing algorithms that fall into each category are further introduced in 2.4. Furthermore, the methods can also be divided into parametric and non-parametric methods. *Parametric method* is a model-based approach, where the received signal is modeled by a certain process, such as Autoregressive (AR), Moving Average (MA) or Autoregressive Moving Average (ARMA) [23], and the parameters of the signal are estimated from the model. *Non-parametric method* aims at finding acceptable estimate without prior knowledge about the underlying stochastic approach [21].

In addition to previous categorizations, the term spectrum sensing may cover location-based sensing, beaconing, external sensing, multi-dimensional sensing, cooperation of several sensor nodes, or using detection algorithms in parallel. The focus of this thesis is on indirect methods as well as parallel and cooperative sensing schemes.

2.3 Hypothesis Testing

A key task in the detection of primary signals is to decide whether the sensed frequency band is idle or not. From signal detection point of view, this can be formulated as a binary hypothesis testing problem. In the scenario, the null hypothesis, denoted by $\mathcal{H}_0$, indicates that the received signal is only noise. On the other hand, the alternative hypothesis denoted by $\mathcal{H}_1$ indicates that the received signal contains primary user (PU) signal in addition to the noise.

The binary hypothesis test can be expressed on an Additive White Gaussian Noise (AWGN) channel by [8, 24]

$$\begin{aligned} \mathcal{H}_0 : &\quad x(t) = w(t) \\ \mathcal{H}_1 : &\quad x(t) = s(t) + w(t). \end{aligned} \tag{1}$$

Here t denotes the discrete time index and $x(t), s(t)$ and $w(t)$ correspond to the received signal, PU signal and AWGN, respectively. For most signal detection algorithms, a scalar test statistics $\mathcal{T}$ is computed from the observation vector $x(t)$. Furthermore, a threshold η is used to divide the observation space into two regions: one that fulfills $\mathcal{H}_0$ and another that fulfills $\mathcal{H}_1$. In such cases, the detection is based on comparing the test statistics to the threshold. If the test statistics is greater than the threshold, $\mathcal{H}_1$ is declared true and the signal is detected.

From the definition of the basic detection criteria, the following terms that are relevant to the performance of the detector [7] can be defined:

- **False alarm:** False alarm, which is sometimes referred as false-positive, corresponds to a case when the detector declares the frequency band occupied while it is actually vacant. Considering the binary hypothesis test, this means choosing $\mathcal{H}_1$ when the correct hypothesis was $\mathcal{H}_0$. The probability of false alarm, or **False Alarm Rate** (FAR), can be expressed by

$$P_{FA} = Pr\{\mathcal{T} > \eta \quad |\mathcal{H}_0\}. \tag{2}$$

  If the probability of false alarm is too high, the spectrum opportunities may be overlooked resulting in inefficient spectrum utilization. Therefore, in majority of the single-user detection algorithms, the threshold is chosen so that a fixed false alarm rate is obtained and the detection probability is then maximized under that constraint. This approach is known as Neyman-Pearson (NP) detection criteria [25].

- **Detection:** The case in which the detector declares the channel correctly occupied, i.e. detects the presence of the PU is called a detection, or a true-positive. Therefore, **detection probability** can be defined as the probability that $\mathcal{H}_1$ is correctly chosen:

$$P_D = Pr\{\mathcal{T} > \eta \quad |\mathcal{H}_1\}. \tag{3}$$

  However, the distribution of samples under $\mathcal{H}_1$ and hence the corresponding threshold η often differs from those of the samples under $\mathcal{H}_0$. Since the distribution of the primary system is not always known or cannot be reasonably determined, it may be difficult to compute the threshold based on that. Thus, it is usually assumed that discarding $\mathcal{H}_0$ is a sufficient prerequisite for a positive signal detection. Furthermore, since the distribution of noise can be assumed to be AWGN, it is often used for determining the threshold for the hypothesis test.

  Maximizing the detection probability withing the limits set by the chosen false alarm rate is crucial for efficient spectrum sensing. The relation between false alarm and detection probabilities can be expressed by a Receiver Operating Characteristic (ROC) curve, which is a commonly used for characterizing the overall performance of a signal detector or a detection algorithm. Another

commonly used performance metric is the detection probability as a function of Signal-to-Noise Ratio (SNR), which characterizes the sensitivity of a detector or an algorithm.

- **Miss:** As opposed to a false alarm, a miss (or false-negative) means declaring the frequency band vacant when the PU signal is actually present. Logically, this means choosing $\mathcal{H}_0$ when $\mathcal{H}_1$ is the correct hypothesis. **False negative rate** can be derived simply from the detection probability:

$$P_{FN} = 1 - P_D. \tag{4}$$

Test Statistics and Detection Criteria

As mentioned in the previous section, a scalar test statistics is calculated for the hypothesis test. Assuming that individual observations are independent of each other, the Likelihood Ratio Test (LRT) is considered to give optimal test statistics under several detection criteria. LRT statistics is given by

$$\mathcal{T}_l = \frac{L(\mathcal{H}_1|x)}{L(\mathcal{H}_0|x)}, \tag{5}$$

where $L(\cdot)$ represents the likelihood function and x the observation. Several signal detection algorithms that are based on hypothesis testing under a certain detection criteria can be derived from the LRT. Common examples of such criteria are Neyman-Pearson, Bayesian, minimax, locally optimum, and sequential detection. The choice of criterion depends on which of the performance parameters should preferably be optimized. A more comprehensive study on the criteria can be found in [25, 26].

This work focuses on **Neyman-Pearson** (NP) detection criterion. It is based on a lemma, which states that when performing a binary hypothesis test, the LRT which rejects $\mathcal{H}_0$ when $\mathcal{T}_l > \eta$ is the most powerful test of size α for a threshold η if η is chosen so that $P_{FA} = \alpha$ (from 2) [27]. Hence, the NP test maximizes the probability of detection for a chosen constant probability of false alarm. Prior information about the noise statistics, especially the distribution is required for NP test so that a threshold corresponding to given P_{FA} can be calculated. Chapter 2.4.3 describes an implementation of a detector that is based on NP formulation.

Sensitivity Requirements

In order to have a common understanding of what should be considered a miss-detection, a sensitivity requirement has been proposed by FCC to IEEE 802.22 standard [28]. Under current understood requirements, secondary cognitive users are required to sense signals as low as -116 dBm averaged over 6 MHz bandwidth, with false negative and false positive rates being 0.1 [29, 30].

Let's consider a real-world example: Channel bandwidth in Finnish DVB-T system is 8 MHz [31], and averaging the sensitivity requirement over that bandwidth

gives a sensitivity requirement of -114.8 dBm. Thermal noise power is -174 dBm/Hz, which corresponds to -104.9 dBm over an 8 MHz band. Consequently, the sensitivity requirement for Signal-to-Noise Ratio (SNR) settles at -9.9 dB for an ideal system. However, the SNR requirement is affected by the Noise Figure (NF) of the receiver, which measures the degradation to the SNR caused by radio frequency signal chain, and is never zero with real physical hardware. Hence, in order to meet the IEEE standard, a cognitive radio user operating in a Finnish DVB-T band needs to meet the given P_D and P_{FA} constraints with signals as weak as -114.8 dBm and thus be able to satisfy corresponding ROC with SNR being below -9.9 dB, depending on the quality of the signal chain.

Having one strict sensitivity requirement entails two major issues: The detection of such weak signal may often be unnecessary and may lead to missed spectrum opportunities. On the other hand, in in the presence of deep channel fading even the strict requirement might not be enough for reliable detection, which leads to a hidden node problem (see 3.2). For this reason, various other methods than single user spectrum sensing are crucial for efficient spectrum utilization. The formulation of the optimal single-user sensitivity requirement under various conditions is outside of the scope of this thesis and thus the standard set by IEEE 802.22 standard is assumed to be sufficient.

2.4 Single-User Spectrum Sensing Algorithms

Single user spectrum sensing algorithms can be classified into three main categories: blind sensing, semi-blind sensing and non-blind sensing. Classification is based on the amount of prior information they require on the primary system [22]. Most commonly recognized detection algorithms are presented in this section briefly. More comprehensive algorithm analysis among with performance comparisons can be found in [18, 21, 32]

Blind sensing schemes requires no information about the primary system, nor the noise distribution. Three distinguishable detection methods fall into this category: *wavelet-based, covariance-based and Information Theory Criteria (ITC) -based detection.* Wavelet-based detection uses wavelet transform to identify non-overlapping spectrum sub-bands efficiently over a wide frequency range. The sub-bands can then be categorized into black, grey and white spaces (spectrum holes) based on Power Spectral Density (PSD) [33]. ICT-based detection calculates the similarity between distributions of the received signal and AWGN to locate the vacant sub-bands. One approach based on Akaike Information Criterion (AIC) is presented in [34]. Finally, as expressed in [22], covariance (or eigenvalue) -based detection *"exploits the differences between the covariance matrices of the correlated signal and the independent noise to deliver highly reliable spectrum sensing"* [35]. None of the blind sensing schemes can differentiate the primary signal from interference and thus are only suitable for coarse spectrum scanning, i.e. for two-stage sensing schemes (see 3.4).

Semi-blind sensing schemes do not require prior information on the primary system either. However, as opposed to blind sensing, these algorithms obtain noise

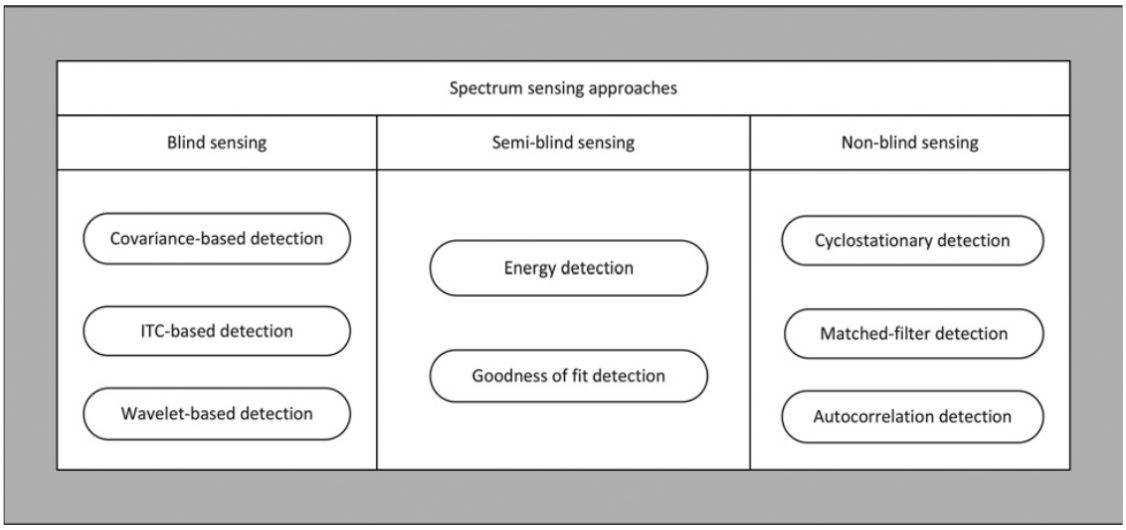

Figure 2: Classification of most common standalone spectrum sensing algorithms [22].

information from learning or training. The most widely used method that falls into this category is *energy detection*, which is based on the energy summation of the received signal [36]. Energy detection is introduced in more detail in its own section (see 2.4.1). *Goodness of Fit Test* (GFT) calculates the discrepancy between the distribution of the observed samples and the distribution of the samples expected under noise conditions [37]. The performance of the semi-blind sensing methods is strongly limited by the noise and/or system uncertainty [17]. However, since the distribution of the samples is empirically estimated, GFT detection is robust to non-Gaussian noise.

Non-blind sensing methods can be only deployed given the knowledge of the primary system. In return, they offer significantly more robust sensing performance compared to blind or semi-blind schemes. Non-blind detectors can be divided into *feature detectors and matched-filter detectors*. Matched-filter detectors are in the extreme end of the algorithm classification and require explicit assumptions on the known pilot waveform or the preamble [38], while feature detectors only rely on certain structural or statistical properties of the primary signal. A general introduction to feature detectors is given in 2.4.2, followed by detailed description of Spatial Sign Cyclic Correlation (SSCC) detector both in time and angular domain (ADSSCC) and the latter is also implemented in this work (see 4.3.2).

2.4.1 Energy Detection

Energy detection, also called radiometry or periodogram, is one of the simplest and most generic forms of spectrum sensing [12]. Theory of the energy detection has been studied in a classic paper by Urkowitz [36] in 1967 and it has since become the most commonly used spectrum sensing method due to its low computation and

implementation complexities [18]. As the name implies, energy detection is based on the measured power of the received signal. Hence, test statistics of an energy detector (EDT) correspond to the estimated energy of the received signal and the binary decision of the presence of the primary signal is obtained by comparing it to a threshold [18].

The test statistics for the EDT can be computed from normalized square sum of individual input samples:

$$\hat{\mathcal{T}}_E = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2, \tag{6}$$

where N is the number of samples for one test and $x(n)$ the complex input signal, which consists of the primary signal s(n) and noise w(n):

$$x(n) = s(n) + w(n). \tag{7}$$

The threshold is based on the noise floor and several methods for computing the threshold exist [36, 39]. This work focuses on the usage of the energy detector as a part of a two-stage sensing scheme, so the threshold formulation from a standalone detection point of view is omitted.

The simplicity of the energy estimation itself proposes several strengths. Since the implementation can be done with low amount of logic elements, energy detectors require little chip area and consume little power compared to several other sensing methods [22, 32]. This makes EDT ideal for applications where high detection accuracy is not required, such as a coarse detector of a two-stage sensing scheme (see 3.4). Another asset of the EDT is that it can operate with very short sensing time. It has been proposed in [40] that feasible information on the presence of the primary signal can be obtained with as little as 2 samples when the Signal-to-Noise Ratio (SNR) is greater than 10 dB. Obviously, this amount of samples can only give a very coarse estimate of the signal. While this may not be not suitable for standalone energy detection or signal strength estimation, it fits very well in a two-stage sensing scheme (see 3.4).

However, the generic nature of energy detection embodies some severe limitations. Due to the fact that only the observed signal strength is used for detection, EDT is not capable of differentiating between different types of signals. Thus, energy detection counts as a blind sensing scheme, which entails two issues: Firstly, EDT cannot be dedicated for detecting signals of a specific system. Instead, it only decides whether any system is operating on the specific frequency. Secondly, EDT cannot differentiate signal from noise, which makes it perform weakly under low signal-to-noise ratios and significantly degrades its performance under noise uncertainty [18]. Section 3.2 discusses the effect of various channel conditions to energy detection and an implementation of energy detection is described in section 4.3.1.

2.4.2 Feature Detection

Feature detectors utilize periodic properties of the sensed signal for detection. More specifically, a cyclic feature detector computes estimates of the cyclic features of the received signal and use them to distinguish the communication signal from noise or interferers [41]. Practically all types of communication signals have periodic properties, which makes the application field of the feature detectors broad [42].

A cyclostationary process can be described as a stationary random process for which statistical properties change periodically as a function of time. Cyclostationary features of a signal may be related to its carrier frequency, symbol structure such as cyclic prefix, chip, code or hop rates, as well as their harmonics, sums and differences [43]. Cyclostationary Feature Detectors (CFD) exploit these features for signal detection by measuring the spectral correlation of the signal, and since noise is wide-sense stationary with no spectral correlation, CFDs are able to differentiate primary signal from noise. Furthermore, the CFDs are classified as non-blind sensing schemes (see 2.4), since the common assumption made in all cyclostationary feature detectors is that the cycles present in the signal statistics of interest are known a priori [41].

A common method for detecting the cyclic features of the primary signal is autocorrelation. While autocorrelation detectors essentially detect cyclic features of the received signal, cyclostationary and autocorrelation detectors are usually separated in their own categories. What distinguishes CFDs from autocorrelation detectors is that the CFDs may be based on the detection of any kind of cyclic features of the PU signal, although it may happen with the aid of autocorrelation. Vice-versa, autocorrelation detectors are based on the output of autocorrelation function, but technically the output depends on cyclic features of the signal and at least the length of the symbol and its prefix need to be known.

Autocorrelation in OFDM Systems

Several modern broadband wireless systems, such as Wireless Local Area Networks (WLAN), DVB and LTE, are based on Orthogonal Frequency Division Multiplexing (OFDM) modulation [44]. Thus, it can be assumed that OFDM will be the key technology when the first cognitive radio systems are deployed. This raises a demand for detection algorithms that are capable of sensing OFDM-based signals with high reliability. The emphasis of this thesis will be on the detection of DVB-T systems, which utilize Cyclic Prefix (CP) based OFDM modulation.

The physical layer characteristic of DVB-T signal is specified in ETSI document [45]. The general idea in CP based OFDM systems is that a fixed length sequence of the data is copied from the end of the symbol and appended to the beginning of the symbol. The copied part is called the *Cyclic Prefix*. Thus, one OFDM-symbol consists of N_{FFT} amount of samples and a replica of the last N_{CP} data samples [46]. The structure of the symbol as well as its autocorrelation mechanism is illustrated in Figure 3.

The presence of cyclostationary features in signal can be identified with Conjugate Cyclic Autocorrelation Function (CAF/CCAF)

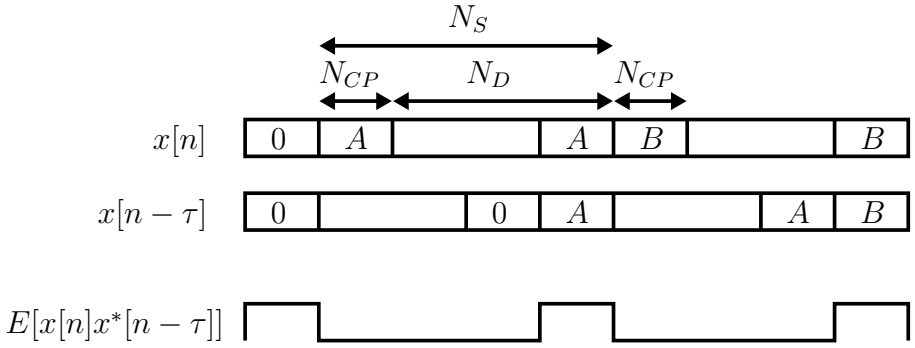

Figure 3: Correlation of an OFDM symbol [46].

$$\hat{R}_x(\alpha,\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-\tau]e^{-j2\pi\alpha n/N} = R_x(\alpha,\tau) + \epsilon, \tag{8}$$

where ϵ is the estimation error and $R_x(\alpha,\tau)$ is nonzero for some cyclic frequency $\alpha \neq 0$ in case the signal $x[n]$ is cyclostationary [47]. Now, the hypothesis problem (1) can be formulated as

$$\begin{aligned} \mathcal{H}_0 :&\quad \hat{R}_x^\alpha = \epsilon \\ \mathcal{H}_1 :&\quad \hat{R}_x^\alpha = R_x^\alpha + \epsilon. \end{aligned} \tag{9}$$

Clearly, the OFDM signal correlates with itself with delay the length of $\pm N_D$ samples, so we can choose the lag parameter $\tau = \pm N_D$ for the CAF in 8. In this case, the cyclic frequency is the length of the symbol: $\alpha = (N_D + N_{CP})^{-1}$ [47]. Considering Neyman-Pearson detection criteria, a constant false alarm rate test under null hypothesis is also needed, and such test is derived in [41] where the test statistics is shown to be χ_2^2-distributed when $x[n]$ is AWGN. A threshold η can be calculated from the inverse of χ^2 Cumulative Distribution Function (CDF)

$$\eta = F_{\chi_2^2}^{-1}(1 - P_{FA}) \tag{10}$$

The test statistics that is compared to the threshold to obtain the binary decision can be derived in several ways. One common approach is to compute the test statistics from a covariance matrix, which can be computed from the autocorrelation of the signal in either time domain or after conversion to frequency domain with Fast Fourier Transform (FFT) [41]. However, such computation often entails high complexity and thus might not be suitable for applications with strict physical constraints. A more trivial approach that is based on Spatial Sign Cyclic Correlation is introduced in the next section and an implementation of it is described in section 4.3.2.

2.4.3 Spatial Sign Cyclic Correlation Detection

Many variants of the cyclic feature detector algorithms are computationally complex and thus also power-consuming. Especially the conversion to frequency domain

with Fast Fourier Transform (FFT) or the calculation of test statistics from covariance matrix are computationally expensive operations [46]. The *Spatial Sign Cyclic Correlation* (SSCC) detector has been first presented by Lundén in [48]. It can be implemented entirely in either time or angular domain and thus does not require FFT. It is also shown in the paper, that with SSCC, the covariance matrix takes a trivial form and does not need to be estimated from the received data when calculating the test statistics. These features make SSCC detector one of the most energy-efficient variants among cyclic feature detection algorithms.

In addition to energy consumption, the SSCC has also been shown to be more robust against impulsive noise [24,48]. In a real-world environment, the noise is rarely only AWGN, but instead has an impulsive nature [49, 50]. Spatial Sign Function (SSF) that is the core component of the SSCC algorithm discards the amplitude of the received signal prior to the detection, which mitigates the issues caused by impulsive noise. Furthermore, tests implemented with SSCC are Constant False Alarm Rate (CFAR), and thus satisfy the Neyman-Pearson detection criteria. As explained in 2.3, Neyman-Pearson detectors maximize the detection probability given a fixed FAR.

Energy efficiency can be improved even further by estimating the cyclic correlation in angular domain and by calculating the test statistics with a uniform distribution test directly in the angular domain. These methods are described in the following two subsections. However, the methods embody some minor drawbacks, most significant being the degradation of the sensitivity relative to the detection time. Figure 4 depicts the sensitivity of each implementation compared to a FFT-based CFD.

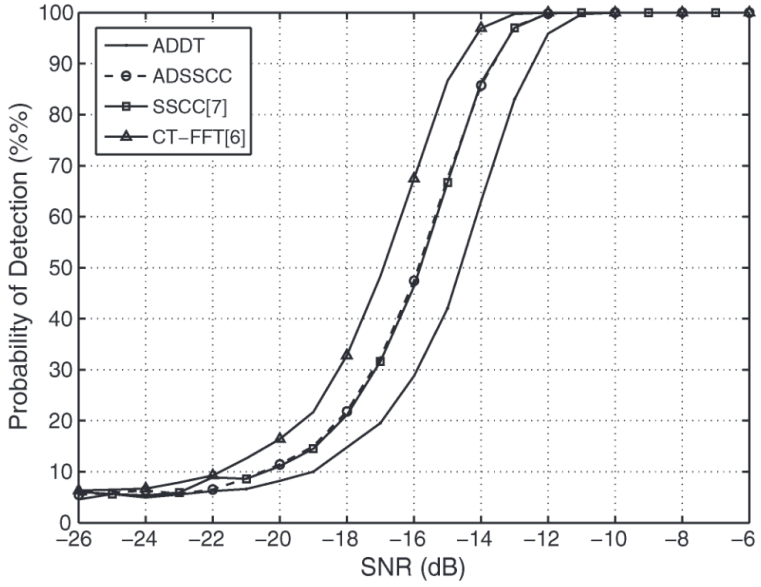

Figure 4: Sensitivity of the SSCC algorithm variants compared to a FFT-based CFD [47].

SSCC Estimator

It is known that cyclic frequency components are also present in the autocorrelation of the Spatial Sign Function (SSF) of the complex input data [47]. The SSF for input data $x[n]$ is defined as [51]

$$S(x[n]) = \begin{cases} \frac{x[n]}{|x[n]|}, & \text{if } x[n] \neq 0 \\ 0, & \text{if } x[n] = 0, \end{cases} \tag{11}$$

where $|\cdot|$ denotes the modulus of complex-valued argument. Practically, the SSF can be understood as normalization of the absolute values of non-zero complex samples [47]. The Spatial Sign Cyclic Correlation (SSCC) estimator can then be defined as

$$\hat{R}_S(\alpha, \tau) = \frac{1}{N} \sum_{n=0}^{N-1} S(x[n]) S(x^*[n-\tau]) e^{-j2\pi\alpha n/N}, \forall \tau \neq 0, \tag{12}$$

where $x[n]$ is the discrete time signal, τ is a discrete lag parameter, N is the number of observations and α is the cyclic frequency [48]. It is worth noticing that the equation is very similar to conjugate CAF (8), with $x[n]$ replaced by $S(x[n])$. Compared to the time domain CAF, this is computationally far simpler, as it only requires summation of angles instead of calculating the absolute values followed by a division [47]. In-depth analysis of the spatial sign based detection can be found in [24] and formulas for the computation of the test statistics in [48].

$$E[S(x[n])S(x^*[n-\tau])] \tag{13}$$

Angular Domain SSCC Estimator

By converting the SSF to angular domain prior to estimation, the computational complexity of the cyclic correlation can be reduced. Although the implementation does require transformations to the angular domain and back, the whole algorithm requires less computational effort than straightforward calculation of the SSF in 13. The transformations can be implemented very efficiently with COrdinate Rotating Digital Computer (CORDIC) algorithm [52]. Two complex multiplications are hereby replaced by additions and the size of the memory needed to implement the delay is reduced, since the total number of samples is lower after the transformation [47].

The spatial sign function can be expressed in angular domain by

$$S(x[n]) = \begin{cases} \frac{x[n]}{|x[n]|} = r_x[n] e^{i\varphi_x[n]}, & \text{if } x[n] \neq 0 \\ 0, & \text{if } x[n] = 0 \end{cases} \tag{14}$$

where

$$\varphi_x[n] = arg(x[n]) \tag{15}$$
$$r_x[n] = 1 \quad \forall\ x[n] \neq 0. \tag{16}$$

The amount of zero samples can be assumed to be negligible and hence the radius $r[n]$ can be discarded. Now, the spatial sign cyclic correlation can be rewritten as

$$\begin{aligned} \hat{R}_S(\alpha, \tau) &= \frac{1}{N} \sum_{n=0}^{N-1} e^{j\varphi_x[n]} e^{j\varphi_x[n-\tau]} e^{-j2\pi\alpha n/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} e^{j\Phi_x^\alpha[n]} \end{aligned} \tag{17}$$

where

$$\Phi_x^\alpha[n] = \varphi_x[n] - \varphi_x[n-\tau] - \varphi_\alpha[n] \tag{18}$$
$$\varphi_\alpha[n] = 2\pi\alpha n/N. \tag{19}$$

Uniform Distribution Test for $\mathbf{\Phi_x^\alpha[n]}$

In order to calculate the summation in 18, conversion to Cartesian coordinates is needed. An alternative method for performing the hypothesis test is to observe the distribution of the arguments $\varphi_x[n]$, instead of calculating the test statistics with 17. This way the final coordinate conversion and hence the second CORDIC can be omitted which simplifies the computation even further.

As described in 2.3, the signal $x[n]$ ideally contains only uniformly distributed noise under null hypothesis. Therefore, also its arguments $\varphi_x[n]$ can be assumed to be uniformly distributed in $[-\pi, \pi)$ and the presence of the PU signal can be detected by simply observing if the distribution is non-uniform. However, if Φ_x^α is not clearly non-uniform, the sum in 18 would tend to zero and a clear hypothesis could not be made [47].

The uniform distribution test can be conducted by dividing the unit circle into M equally sized sections and calculating the amount of observations that fall into each sector. For uniformly distributed data, the observations follow binomial distribution [47]

$$O \sim B(N, 1/M). \tag{20}$$

If the amount of observations deviate from the mean value ($N_S/4$) by excess or defect more than a precalculated threshold, which is based on the distribution, the null hypothesis can be rejected and the signal eventually detected. The equation for calculating the threshold for binary decision is presented in section 4.3.2 along with the description of the implementation.

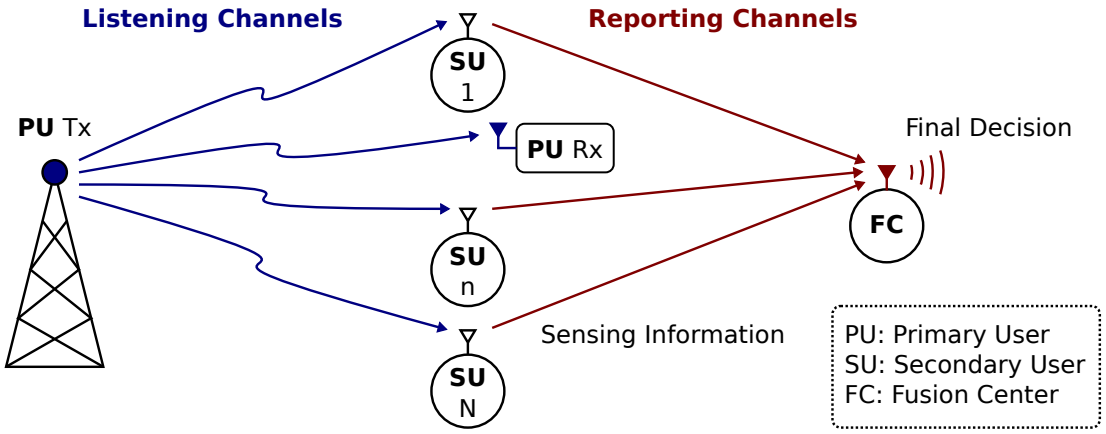

Figure 5: Architecture of cooperative sensing scheme [56].

2.5 Cooperative Sensing

A typical cognitive radio network consists of multiple geographically distributed secondary users which all have to detect whether the primary user is active or not. The decisions made by a single detector may become highly unreliable in the presence of propagation effects such as shadowing and fading. Thus, it is often beneficial to perform spectrum sensing by cooperating with other users in the geographical neighborhood [53]. The approach is called **Cooperative Sensing** (CS), which originates from the field of distributed detection that has been of great interest in the field of sensor networks and radar [54, 55].

There are basically two main objectives that CS aims to fulfill. Firstly, it can improve the reliability of primary user detection, especially in the presence of channel fading and shadowing (see 3.2). Secondly, it relaxes the sensitivity requirements for a single sensor node and thus can ultimately reduce the need for local sensing [53]. While cooperative sensing by itself may incur cooperation overhead due to communication with other sensors, methods such as data censoring allows CS to reduce the overall energy consumption and delay of individual sensors. The censoring based CS is discussed separately later on.

Cooperative sensing can be classified by its operating principle based on the data processing model or the way the information is exchanged among secondary users. In *centralized* CS scheme, one node acts as a **Fusion Center** (FC), which combines the results from neighboring detectors according to the chosen fusion rule. The fused result is then transmitted to other nodes for evaluation. Conversely, in *decentralized* scheme, each individual sensor acts as a fusion centers and all the measurement data is shared and fused by the sensors in a geographical neighborhood. Another way to classify CS is based on how the information is exchanged among different secondary users. The most widely used topologies are *parallel, serial* and *tree* [56].

Test results are combined in the fusion center using a specific **fusion rule**. The rules can be divided into *hard* and *soft combining*. Hard combining means the combining the boolean decision with boolean rules such as AND, OR or MAJ.

Soft combining, on the other hand, combines the test statistics, for example by summation, and performs the hypothesis test on the result. While hard combining induces some performance loss due to quantization of the results, it requires less communication bandwidth since the individual decisions can be expressed with just one bit. Thus, it may often be beneficial for the overall outcome not to aim for lossless combination [57].

Further analysis on cooperative sensing and results from field trials, carried out in collaboration with this project with the same hardware (see chapter 4), can be found in [6, 56].

Censoring-based Cooperative Sensing

Other methods for energy saving exist besides the methods that aim at reducing the required run time or energy consumption of an individual sensor node. Two methods worth mentioning are censoring [58] and sleeping [59]. Both of them are essentially related to cooperative sensing or database-centered spectrum access approach. These methods are based on a simple fact: running all the sensors with full capacity constantly is very inefficient.

In cooperative sensing schemes that are based on data fusing in Fusion Centers (FC), a significant part of samples produced by individual sensor nodes contain very little feasible information. Therefore, it may be beneficial to filter the data so that only informative samples are sent to the FC. The decision statistics are deemed informative if they are sufficiently high or low, that is, favor one of the hypotheses clearly. Hence, if a sample indicates that the primary signal might or might not be present, the sample is discarded. This process is called **censoring**. Furthermore, if the FC already contains enough information about the presence of primary users, there is no need to constantly sense the spectrum with each sensor node. Instead, they should be run with certain interval and only when truly needed. This method is called duty-cycling or **sleeping**. Both of the aforementioned methods are strongly related to energy efficiency of spectrum sensing, which is discussed in section 3.3.

2.6 Spatial Interpolation

Spatial interpolation is a database-centric method for addressing the coverage issues of geospatial spectrum utilization databases. The general idea of spatial interpolation is to generate occupancy data from existing measurements over areas where the data cannot be feasibly provided. Hence, it is not directly spectrum sensing, but a method for post-processing the data that the sensing provides. Especially large scale wireless sensor networks are known to suffer from coverage holes, that may result from hardware failure, data corruption, extensive costs of redeployment or the hostility of deployment areas [60].

The interpolation methods can be divided to deterministic and statistical methods. The deterministic methods focus on the specific model starting from the interpolation functions, whereas the statistical methods focus on modeling the phenomenon itself, starting from the measurement data. Examples of the deterministic

methods are Thiessen, Voronoi and natural neighbor interpolation, as well as trend surfaces and splines. A typical example of statistical methods is Kriging.

There are several benefits from using Kriging compared to the deterministic methods. First, Kriging is capable of extrapolating the data outside the measurement hull. Second, it calculates the estimation error in addition to the actual interpolation data, which significantly increases its reliability and feasibility in spectrum sensing purposes, as well as improves the reliability with noisy data. Since the estimation is based on all the data points in the range instead of just the surrounding ones, the method also mitigates the hidden node problem and other channel conditions (3.2) better. However, the main downside of Kriging is the significant complexity of the calculation. For this reason, the data produced with the deterministic methods are ideally used as a threshold for deciding if a statistical calculation or local sensing is needed, whereas Kriging should be used whenever the calculation time and resources are not constrained.

An application of Kriging on field measurement data is evaluated in this work, so we focus on the method more specifically in the following subsection.

Kriging

Kriging is essentially a two-stage interpolation procedure that generates an estimated surface from scattered data points with z-values. From the interpolation point of view, the method is very similar to Inverse Distance Weighed (IDW) interpolation, except that the weights for the surrounding points are calculated with a more sophisticated method. The weighting is based on a statistical calculation of the spatial nature of the data points, generally to satisfy the basic principle of geography: things that are closer are more alike than those that are farther apart.

The Kriging procedure begins with modeling of the functions that estimate the statistical dependence of the measured values, or in other words, their *spatial autocorrelation*. Spatial modeling, also known as *variography*, usually involves a graph of empirical semivariogram, which illustrates the semivariances for all pairs of data of a measurement location. For location pairs $\{s_i, s_j\}$ the semivariogram can be calculated as

$$\hat{\gamma}(s_i, s_j) = 0.5 * \{Z(s_i) - Z(s_j)\}^2, \tag{21}$$

where $Z(s)$ denotes the measured value at location s, and $\hat{\gamma}(s_i, s_j)$ is the semivariance for a location pair $\{s_i, s_j\}$.

However, since the data is very rarely uniformly distributed, there are often so many location pairs that the variogram would become difficult to analyze and be sensitive to outliers. Hence, a *binned variogram*, that divides the location pairs to a limited amount of distance classes, is often used [61]. It can be considered as a moving average of the semivariance cloud.

After obtaining the semivariance plot, a *variogram* model needs to be fitted to the data. There are only general guidelines for finding a sufficient model, such as Least Squares (LS), Weighted Least Squares (WLS) or Residual Maximum Likelihood (REML). The correctness of the aforementioned methods depends on the amount of

data points, nature of the affecting phenomena and properties of the data, so it may often be preferable to select the theoretical model by eye and use a mathematical method, such as LS, for finding the optimal parameters for the model. Figure 6 depicts the empirical semivariaces and a fitted model. Commonly used models for the variograms are linear, exponential, circular, spherical and Gaussian. The model presented in the figure follows an exponential function.

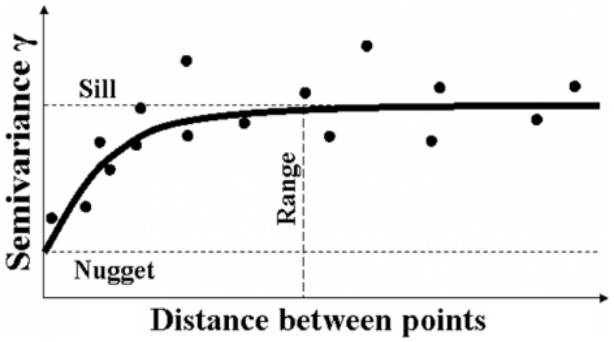

Figure 6: Semivariances and a variogram model [62].

The figure also introduces the terms *sill, nugget* and *range*. *Range* is the distance at which the y-value of the variogram reaches 95% of the value it converges to, or in practice, where the model first flattens. *Sill* is the semivariance value at which the theoretical variogram model attains the range. *Nugget* is a non-zero value at infinitely small separation distance, which is basically consequence of measurement errors of the data. In addition to the terms presented in the figure, a *partial sill* is the sill minus nugget.

If it is possible to scientifically rationalize that an external condition causes an overriding trend to the scattered data points, the condition can be used as a *covariate* of the process. Practically, the phenomenon is modeled with a deterministic function, a polynomial, which is subtracted from the data values prior to variogram fitting. The polynomial is then added back to the data before the actual estimation, in order to make the final results meaningful. A covariate can be either directional, such as attenuation of a signal, or direction-independent. A method that models such a condition, is called *Universal Kriging*. On the other hand, *Ordinary Kriging* assumes that the mean is constant over the measured area and is therefore more generic [63].

From spectrum sensing point of view, the covariates can be for example the distance to the transmitter and height of the measurement point, since these two parameters can be assumed to affect the strength of the observed signal. Conversely, the variogram model defines how the distance to the neighboring data points affects their correlation [60].

After the modeling of spatial autocorrelation is completed, the estimation phase can be carried out. The general formula for the interpolator can be defined as

$$\hat{Z}(s_0) = \sum_{i=0}^{N} \lambda_i Z(s_i), \tag{22}$$

where λ_i is an unknown weight for the measured value at location s_i, obtained from the aforementioned variogram model, s_0 is the prediction location, and N is the number of measured values. Finally, an estimation surface can be plotted from the computed Z-values.

Extensive theory of variography and Kriging can be found in [61, 64]. Section 5.3 presents a practical application of Kriging on field measurement data obtained by spectrum sensing.

3 Spectrum Sensing with Mobile Hardware

One of the ultimate goals of cognitive radio is *to provide highly reliable communication for all users of the network, wherever and whenever needed* [9]. The cognitive network is basically formed by two types of Secondary Users (SU): fixed devices, such as Wi-Fi access points, and mobile devices that either join the existing networks established by the access points or form a network of their own. For fixed devices, a database centered approach for gaining reliable spectrum awareness is often preferred, since their database access can be easily organized. On the other hand, with mobile devices, the exact geographical location is not always available and access to the database cannot be guaranteed. This in turn makes spectrum sensing for mobile compliant hardware a crucial field of study [65].

This chapter explains the relevant background theory for non-idealities and other aspects that need to be taken into account in mobile sensing. The aim is to give an understanding on what to take into account in the implementations and what to expect from the measurements, especially in field conditions. This is especially important, since all the presented implementations are carried out on a hardware platform that is designed for mobile usage.

Many of the CR demonstrators presented in literature [66–68] use lab grade instruments with price and form factor. While these are suitable for theoretical research and validation of spectrum sensing methods, they are not suitable for mobile and consumer usage. This work focuses on the implementations of spectrum sensing techniques on real hardware that can be embedded to a mobile device. Various issues related to spectrum sensing with such hardware is discussed throughout this chapter with emphasis on physical limitations and energy efficiency considerations.

3.1 Physical Constraints

Spectrum sensing hardware that is suitable for mobile usage is constrained by several factors such as price, delay, energy efficiency and form factor. Especially the energy consumption of an algorithm or a sensing scheme plays an important role, since it can be assumed that mobile devices are battery operated. On the other hand, price and footprint are limited by the design parameters of the device, and sensing time (delay) chosen to meet the performance requirements set by the standards.

In order to guarantee operation without causing interference to hidden primary systems or deeply faded channels, the SU needs to be able to sense the presence of very weak primary signals. As discussed in 2.3, the current requirement set by FCC states that TV transmissions as low as -116 dBm (SNR -22 dB) need to be detected [28, 30]. Hence, the practical goal for spectrum sensors is to meet a given ROC curve at given SNR. Classical detection theory suggests that degradation in the ROC due to reduced SNR can be countered by increasing the sensing time [25, 26]. This perspective implies that cognitive radio system can always be engineered at the cost of low enough Quality of Service (QoS). In reality, the performance of spectrum sensor may be greatly degraded by practical non-idealities such as channel impairments, hardware non-idealities and errors in underlying assumptions.

Hardware-originated issues depend on the used architecture of the receiver: superheterodyne or direct conversion. The difference between these architectures is that superheterodyne receivers first converts the RF signal to Intermediate Frequency (IF) for processing before converting it to baseband (BB), whereas direct conversion receivers convert the signal straight from RF to BB. This thesis focuses on issues related to direct conversion receiver, as the prototype platform used for implementations is based on this architecture. In addition, direct conversion receivers are considered a more cost effective solution for multi-standard/multi-band radio designs due to their high selectivity, possibility of using microprocessors and no issues of image frequency (the spectrum mirrored to negative frequencies) [69]. Hence, it is also most commonly used architecture in cognitive radio sensor nodes [70, 71].

In addition to non-idealities of the RF front end, other hardware choices and design patterns such as antenna and its matching circuitry and mechanics may significantly affect the performance of the detector. There are also several affecting factors that are not dependent of the hardware. Most of the relevant non-idealities and resulting issues present in a cognitive radio system are discussed in the following subsections.

3.1.1 Hardware Non-idealities

Generally speaking, none of the parameters in a real-world system is known to infinite precision and physical hardware is never ideal. A few notable examples of non-idealities are:

- **Filters:** The frequency response of filters is never ideal. The pass-band is not exact, the filtering slope is not infinitely steep and the attenuation in the stop-band is finite. These issues may cause noise or LO leakage and lead to errors in noise and signal power assumptions.

- **Offsets:** The physical differences in the In-phase (I) and Quadrature (Q) signal branches of the receiver lead to IQ-imbalance. These signals might also feature DC offset that can either originate from second-order nonlinearity (see 3.1.3) or offset in the threshold of ADC. DC offset may decrease the SNR, but can be compensated with various methods [71, 72].

- **Noise:** Various parts of RF front-end chain generate some type of noise. For example, the limited precision of A/D converters generate quantization noise, imperfect waveform of Local Oscillator (LO) generates phase noise. Each noise source increases the noise figure of a detector which degrades the sensitivity. Furthermore, phase noise may significantly degrade the performance of pilot detectors.

- **Interference:** Various parts in the hardware can interfere with each other or catch interference from the environment. The interference may be caused by clock feed-through, DSP blocks, intentional interferer or jammer (narrowband interference). Interference degrades sensitivity and may cause false alarms in almost any detector types.

3.1.2 Non-linearity

One of the most problematic non-idealities that possess consumer-grade hardware is non-linearity. In short, when the output of a system, such as RF signal chain, is not directly proportional to the input, the system is nonlinear. Non-linearity may originate basically from any part of an analog signal chain, including the amplifiers, mixers, filters and ADC.

Especially third- and second-order non-linearity is common in physically constrained hardware. For example, when two or more signal components are present in a systems, third order non-linearity causes **Intermodulation (IM) products** on frequencies determined by a third-order polynomial of the signal frequencies. The amplitude of the resulting products is proportional to the original signals' amplitude by a factor of 3 [73]. For example, if the amplitude of the primary signals is increased by 10 dB, the IM products are amplified by 30 dB. Furthermore, the more primary signals fall in the frequency range of a receiver, the more intermodulation products will be present. This is called the **wideband signal effect** and it proposes strict linearity requirements for wideband systems, which are becoming increasingly common in today's radio technology [70, 74].

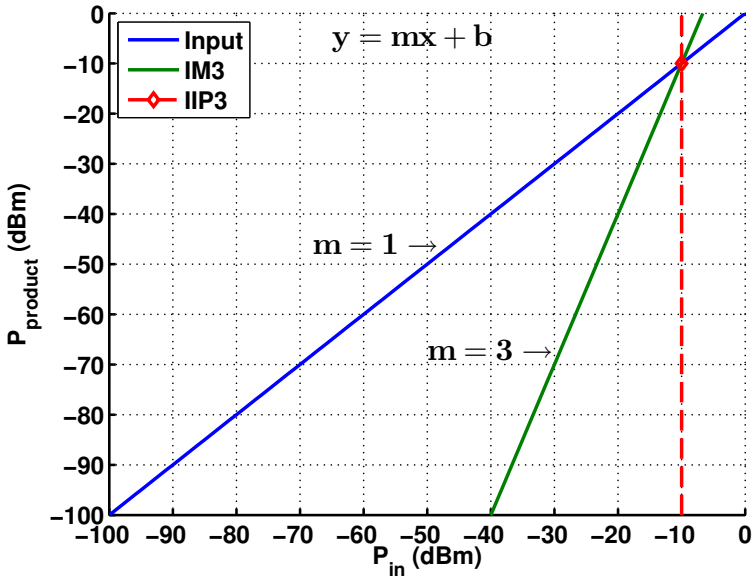

Figure 7: Third-order Input Intercept Point (IIP3) for spectrum sensor prototype.

Figure 7 demonstrates the relation of the amplitudes of primary signal and third-order IM product. The point where primary signal's amplitude equals that of the IM product is called the **Input Intercept Point (IIP)**. In the figure, the marked point represents the intercept point for third-order intermodulation, hence the abbreviation IIP3. IIP is commonly used to characterize the linearity of an RF front-end or other analog component.

When the subcarriers of a frequency-divided signal, such as OFDM signal, modulate with each other, they produce **self-modulation** products. This, in turn, causes

the components of the signal to spread on the adjacent bands, causing interference with higher bandwidth compared to the primary signal. The phenomenon is called **spectral regrowth**. Considering feature detectors, self-modulation of a primary signal is one of the most problematic results of non-linearity. Since the interference resulting from the spectral regrowth embodies same cyclic features than the primary signal, the primary signal might be detected on the adjacent frequency bands, which is rarely intentional [71].

On the other hand, intermodulation products of two or more primary signals are not as easily detected, since the prefixes of the signals are very rarely synchronized. However, if the amplitude of the resulting intermodulation products is above the noise floor, the performance energy detectors may be significantly affected. The problem may be partly countered by the means of noise modeling (see 3.4). Furthermore, although the intermodulation products do not necessarily cause false alarms in feature detectors, their sensitivity is degraded due to reduced SNR. Measurements on nonlinearity are presented and further discussed in section 5.1.2.

3.1.3 Antenna Limitations

Mobile hand-held devices have strict constraints regarding the form factor of the parts that they consist of. One of the major implementation challenges in such devices is often the antenna. Normally, the antenna needs to be placed in close proximity to the electronics, which entails two possible issues: the antenna may cause interference to other circuitry, and vice-versa, the interference produced by the circuitry may be coupled to the antenna. Especially the latter case may be very problematic, since strong spurious emissions may distort the measurement results severely by causing false alarms or degrading the SNR. The phenomenon along with its effects is measured and analyzed in section 5.1.2.

Another issue is the strict size and shape restriction of the antenna. In order to fit a mobile device, the antenna cannot be designed only to obtain the best performance but it has to be also practical to physically embed in a mobile system. Hence, compromises have to be made, since smaller antennas tend to have lower sensitivity. The aforementioned restrictions also cause performance degradation due to orientation of the antenna. Small mobile antennas and especially printed circuit board (PCB) antennas are almost consistently non-isotropic which means that the sensitivity, or the antenna gain, is dependent on the orientation of the antenna with respect to the location of the transmitter [75]. If the mobile device is held in such position that the gain drops very low, the PU signal might fade so much that the signal cannot be detected.

3.2 Channel Conditions

Propagation effects, such as shadowing and multipath fading [76], degrade the performance of a single detector. *Shadowing*, or shadow fading means channel fading due to obstacles that affect the wave propagation. *Multipath fading* is basically destructive interference which occurs when the radio signal reaches the receiver via

two or more paths. These conditions may result to hidden node problem, where a secondary transceiver is outside the listening range of a primary transmitter but close enough to the primary receiver to create interference.

Channel fading models can be divided into Rayleigh, Rician and Nakagami fading, depending on the distribution of the attenuation [22]. Furthermore, if the user or the reflectors in the environment are moving, the velocity causes Doppler shift in the frequency of the signal transmitted along different signal paths. The resulting channel fading caused by the phase shifts is called Doppler spread. Due to constant changes in the environment, channel fading is neither flat nor constant over time. Figure 8 depicts Receiver Operating Characteristics (ROC) of an energy detector with various channel conditions.

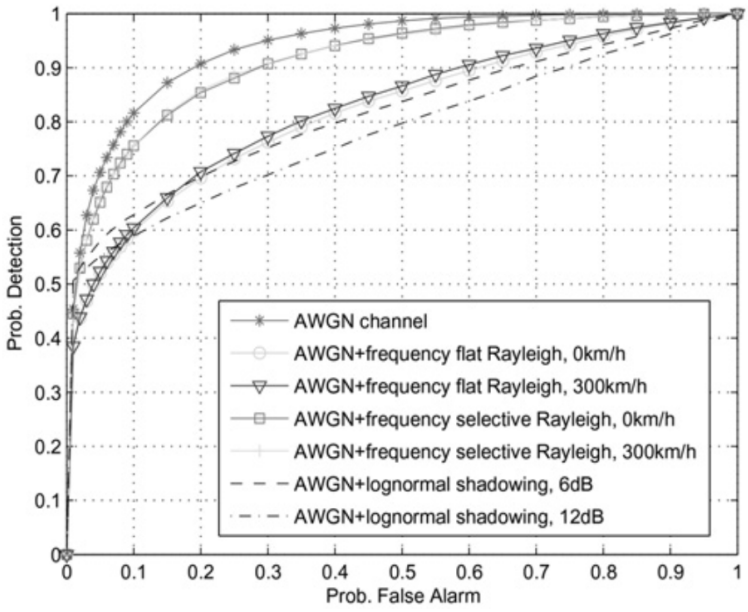

Figure 8: Reveiver Operating Characteristic (ROC) curves with various channel conditions for energy detector [22].

In addition to fading of the primary signal, there always exists uncertainty in the background noise. For example, noise power may vary due to temperature and out-of-band interference, which leads to uncertainty in noise variance. Thus, noise is never perfectly stationary, white nor Gaussian-distributed. This results to SNR wall [17], which is practically an environment-dependent limiter for achievable performance rates of a detector. Furthermore, noise is never perfectly white, nor it is perfectly Gaussian, which leads to uncertainty in noise distribution in addition to its variance. Since many algorithms are based on the assumption that the noise distribution is known, the variance may cause significant performance degradation. Section 5.1.2 presents results from measurement with real hardware that aim to validate this issue.

3.3 Energy Efficiency

In addition to meeting several performance requirements set by standards, such as IEEE 802.22 [28], another very important design parameter in mobile spectrum sensors is the energy consumption, since it can be assumed that almost all the mobile devices are battery operated. For individual spectrum sensor, the energy consumption is dominated by two design parameters: the sensing algorithm(s) and their run time. Normally, these parameters need to be chosen to meet those performance standards within the limitations set by the hardware. However, energy efficiency in respect to sensing accuracy can be greatly improved with various techniques that may be based on the collaboration of sensor nodes or different detection algorithms within a single node.

Obviously, the hardware sets the limits for achievable energy efficiency. The quality of the components contribute to the noise figure and linearity of the device and hence affect the required sensitivity (see 3.1.1, 3.1.2, 5.1.2), the efficiency of the power supply and regulators affect the overhead energy consumption, the chip architecture affects the achievable efficiency of the detection algorithms, and so on. However, within the limitations set by the hardware, the choice of the detection algorithm is a major contributor to the overall energy consumption. It is shown in [46] that the difference of the consumption between algorithms can be even more than ten-fold. In addition, the sensitivity of the algorithm at hand defines how long it has to operate in order to meet the sensitivity standards.

Cooperative sensing (CS) was introduced in chapter 2.5 and it was stated that the ultimate goal of CS is to increase reliability of individual sensors in the presence of channel fading and shadowing, and to relief the sensitivity requirements. However, CS can be also considered a technique for improving the energy efficiency in a sense that it allows the design of individual sensor nodes with lower performance requirements and thus with lower energy consumption. Two additional CS-related techniques, censoring and sleeping, that aim to further reductions in energy consumption were introduced in 2.5.

For a single sensor node, a commonly used technique for improving both energy consumption and speed of the spectrum sensing, is called two-stage (or coarse-fine) sensing scheme. The following subsection introduces such a scheme in detail and an implementation of a coarse-fine controller is presented in section 4.4.

Finally, it should be kept in mind that the overall energy efficiency in a cognitive radio network is not limited to only spectrum sensing techniques. The chosen protocols on different layers (PHY, MAC, routing, etc.), the applications, as well as each parts cooperation with spectrum sensing may greatly affect the overall efficiency of the cognitive radio. For example, the reliability of the spectrum sensor and the channel access model define how far the transmission power can be optimized [77] and the routing protocol and PHY layers define how large packets and how often the radio needs to transmit, in order to fulfill the requirements set by the application.

3.4 Coarse-Fine Sensing Scheme

Coarse-fine detection scheme refers to a two-stage, single-user spectrum sensing architecture, where a measurement metric of a coarse detector is used for controlling the behavior of a fine detector. The profound idea of the coarse-fine scheme is to find the first unoccupied channel with minimal effort by gathering feasible information of the spectrum with lowest possible computational complexity and making the final decision based on a fine detector when needed. This allows the fine detector to focus entirely on bands with low SNR and ignore the bands that can be reliably detected or classified as low priority with a coarse detector, ultimately increasing the efficiency of the fine detector [40] without deteriorating its accuracy. Hence, the coarse-fine approach is often preferred when the implementation area is not limited.

Two commonly considered sensing techniques for cognitive radios are energy detection (section 2.4.1) and cyclostationary detection (section 2.4.2). Due to low complexity, low energy consumption and relatively fast detection time, energy detection is suitable for a coarse spectrum sensing. Furthermore, since cyclostationary detectors do not suffer from noise uncertainty nearly as much as the energy detectors do, they are ideal as fine detectors. Theory of operation and further performance considerations of the detection algorithms can be found in chapter 2.

Based on the current literature, two-stage sensing schemes can be classified into two main categories that differ significantly by their operation principle. The first approach is based on ordering the channels with coarse detector prior to using the fine detector. In this case, the coarse detector is not used for actual decisions on the occupancy of the channels. Instead, it orders the channels for fine detector so that channels that are most likely vacant will be scanned first. For this purpose, a wavelet-based detector with the aid of edge detection is ideal as the coarse detector. It provides a fast overview of the energy distribution over a wide frequency band at once with very low energy and detection time [33].

One implementation and performance simulations of this approach is presented in [40]. Flow chart of this scheme is presented in Figure 9.

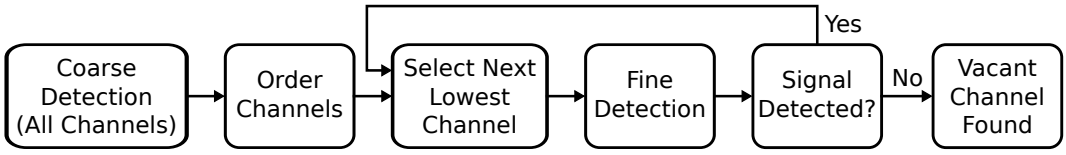

Figure 9: Ordering coarse-fine scheme, as presented in [40]

Another approach is to scan channels separately with an energy detector and using its decision metric for deciding whether a fine detection is needed. This is usually done by comparing the energy of a frequency band to a certain threshold, as done in [78]. Figure 10 depicts the basic architecture of this method. The method is also chosen for the implementation presented in this work (see chapter 4).

The achievable benefit of the CF-scheme is usually dependent on the occupancy level in the radio environment, since the energy- and time consuming fine detector

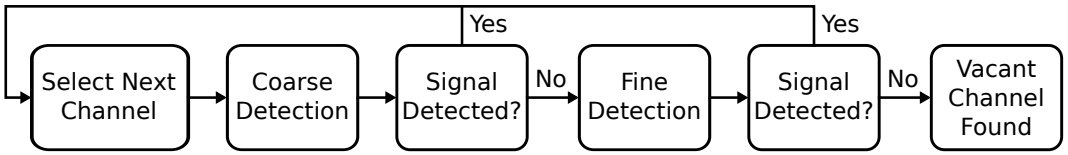

Figure 10: Channel-wise coarse-fine architecture, as presented in [78]

is run each time a channel is declared free with the energy detector. Considering the extreme examples, if the occupancy is 0%, the fine detector needs to be run on each channel or until a free channel is found, which means that the overall performance is equal to that of a standalone fine detector. Conversely, if the occupancy is 100% with high enough SNR, each channel can be declared occupied with the coarse detector and the fine detections can be completely omitted. This phenomenon makes the scheme even more beneficial in the presence of multiple systems: The potential strong interferers from other systems that could be missed by the parametric fine detectors can be distinguished with the more generic coarse detectors.

Noise Modeling

Based on the desirable behavior of the scheme, the threshold of the energy detector has to correspond to a desired sensitivity and false alarm rate. These are relative to the noise floor and noise uncertainty, which is always present in the real-world radio environment. Noise uncertainty is dominated by the unknown interference that is coming from an aggregation of many sources [17]. Hence, the threshold of the energy detector has to be adapted to the environment and channel conditions by continuously adjusting it. For that purpose, an estimate or a model of the noise is needed.

Various advanced noise modeling techniques exist and further analysis can be found in [17]. This work presents a rather simple adjustment scheme, where the threshold is calculated based on the Boolean decisions of the two sensors (see 4.4). This approach is motivated by the idea that significant part of the PU signals, as well as signals from other systems can often be identified with very low sensitivity. By configuring the energy detector to behave very opportunistically, practically no channel capacity is lost compared to using only the fine detector, especially since the goal is to reduce overall run time and energy consumption of the detector.

Implementation of a coarse-fine controller, as well as a description of the aforementioned noise modeling technique are presented in section 4.4. Section 5.2.3 presents verification of functionality and measurements on the achievable benefits from the aforementioned implementations.

4 Implementations

This chapter focuses on implementations of the algorithms described in chapters 2 and 3. We start with a brief description of the prototype platform that is used for the implementations, followed by discussion of its hardware limitations that may contribute to the performance of the algorithms. Next, the implemented sensing algorithms, the new automatic gain controller of the RF front-end, and finally the implementation of coarse-fine controller are presented in their respective subsections.

4.1 Prototype Platform

All implementations presented in this chapter were done with a Spectrum Sensor Prototype platform, depicted in Figure 11.

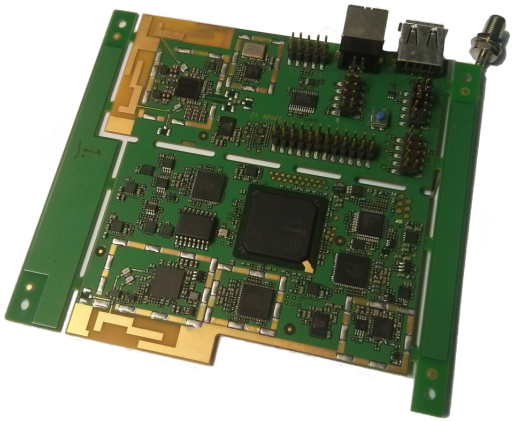

Figure 11: Spectrum sensor prototype platform

The platform is originally designed by Nokia with the ability to be embedded in N900 mobile phone, and an additional wideband section was designed and added to the platform by Aalto University. The prototype platform consists of three RF front-ends (DVB-T, WLAN and wideband), 10-bit A/D-Converters (ADC) for each of the front-ends, an Field Programmable Gate Array (FPGA) and a USB communication bridge for interfacing with a PC or a mobile phone. The FPGA is intended for the implementation of the detection algorithms and controlling schemes. Since the emphasis of this work is on the detection of DVB-T signals, WLAN and wideband front ends and their related algorithms are omitted.

Figure 12 depicts the organization of the functional blocks within the FPGA. The Control and Communication units are Nios II embedded 32-bit cores that are capable of running control software written as a C or C++ application. Currently, the Communication unit serves as an interface for the USB communication while the Control unit interfaces with other hardware, including the RF front end and other components implemented in the FPGA. Digital baseband handles the digital processing and filtering of the signal received from ADCs and consists mostly of

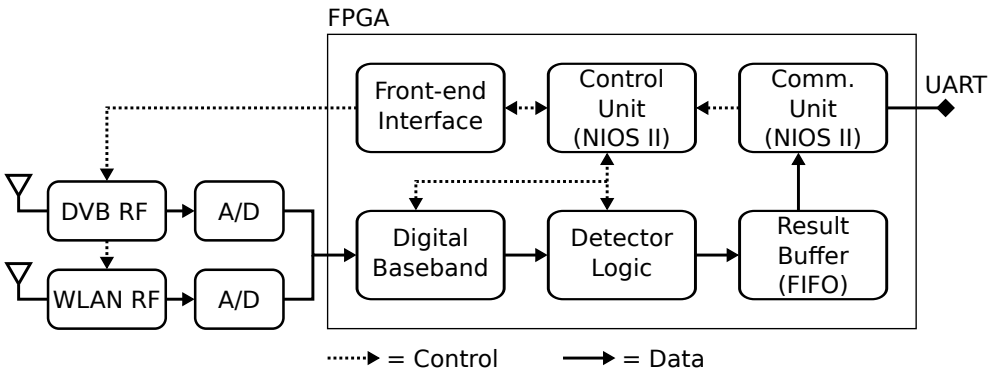

Figure 12: Function blocks on the detector board

Finite Impulse Response filters (FIR) and decimators. The detector logic block includes the Energy Detector (section 4.3.1), ADSSCCDT (section 4.3.2) and the coarse-fine controller (section 4.4). The detection results are fed to a First In First Out (FIFO) buffer, from where they can be transferred to the interfacing device by the Communication unit. Further details on the implementation of the hardware can be found in [3].

Hardware Limitations

A general overview of real world constraints was given in chapter 3 and most of these limitations apply to the spectrum sensor prototype as well. Real hardware is never optimal and compromises have to be made between complexity, price and performance. As the SSP is designed mainly for mobile usage and intends to be close to real end application hardware, especially the RF signal path embodies several non-idealities.

The most significant issue in the spectrum sensor prototype is non-linearity. Especially the 3rd order non-linearity (explained in 3.1.2) is proven to cause false alarms via self modulation of the signal. A validation of this issue is presented in section 5.1.2. Other hardware-constrained issues are the slightly nonlinear frequency response of the front-end chip, as well as its inaccurate amplifier gains and slow operation of the automatic gain controller. Both of the aforementioned issues lead to inaccurate input power estimation which eventually degrades the sensitivity of the energy detector.

The digital signal path is only 10 bits, so the amplitude of the baseband signal has to be in certain range to get predictable behavior from the detection algorithms. Since the build-in gain controller is slow and inaccurate, a new approach was needed (see 4.2).

It is also worth mentioning that since the algorithm implementations are done on an FPGA, the energy consumption of the platform is higher compared to end products. However, the analysis capabilities of the design software allow for accurate comparison of the power consumption of different sensing algorithms and implementations. In addition to increasing energy consumption, the FPGA causes

spurious signal emissions that increase the false alarm rate of cyclostationary and energy detectors, especially if the antenna is placed poorly. More on this issue can be read in section 5.1.2.

4.2 RF Front-end Control

In order to achieve the theoretical false alarm rate of 0.05 on unoccupied channels, it is assumed that the received signal is almost ideal Additive White Gaussian Noise (AWGN). Due to limited accuracy caused by 10-bit signal paths, this assumption cannot be met with low signal power, since the majority of the input samples will tend to zero or a small integer value. When this is the case, the discrete small-valued samples start correlating, causing the test statistics to increase and eventually get over threshold with almost 100% rate (see 5.1.2). In addition, the signal will drop below the linear range of the RSSI estimator, which may distort the energy estimate severely.

The RF front end chip used in the DVB-T section of the spectrum sensor prototype is primarily designed for mobile TV reception instead of spectrum sensing. Thus, it is not optimized for rapid stabilization after adjustment of the frequency band. This feature can be seen as a long settling of the Automatic Gain Controller (AGC) and Voltage Controlled Oscillators (VCO). Hence, during the settling period, part of the detections may occur at the time the gains temporarily differ from the desired value, which leads to aforementioned distortion in the test statistics and energy estimate.

The front-end utilizes two VCO's, one of which is for the lower half of the frequency band and one for the upper. Hence, when measuring a wide range of DVB-T channels, as in the majority of the measurements presented in this work, the VCO swap inevitably happens several times during the measurements. The effect of VCO settling can be seen in the results as severe oscillation in the RSSI readings as well as occasionally non-deterministic detection probability.

The VCO issue can be fixed by adding a long enough delay after the VCO swap or simply limiting the measurement band to channels 43 and above or 42 and below. The delay has to be at least the length of the worst-case settling time in order to completely mitigate the issue. Since the settling time of the prototype's front-end varies randomly between 1 ms and over 300 ms, a delay of 400 ms was chosen. Such delay can be considered a significant overhead, especially when using short detection times, but cannot be compensated otherwise without modifying the hardware.

New Implementation of Automatic Gain Control

One way to cope with the fluctuations during the settling time of AGC would be to wait after each frequency change and locking the gains prior to the detections, similarly as with the VCO. However, it appeared that the time until gains had completely settled was in the magnitude of hundreds of milliseconds, which obviously made this solution unfeasible for energy- and time-constrained spectrum sensing purposes. Instead, the automatic gain controller was completely re-implemented

by disabling the automatic gain adjustment loop in the RF front-end and giving the software control over the front-end gains. This approach also makes the energy estimation simpler, since the control software is constantly aware of the current gains without reading them from the front-end's registers. The operation of the implemented energy detector is described in its own section (4.3.1).

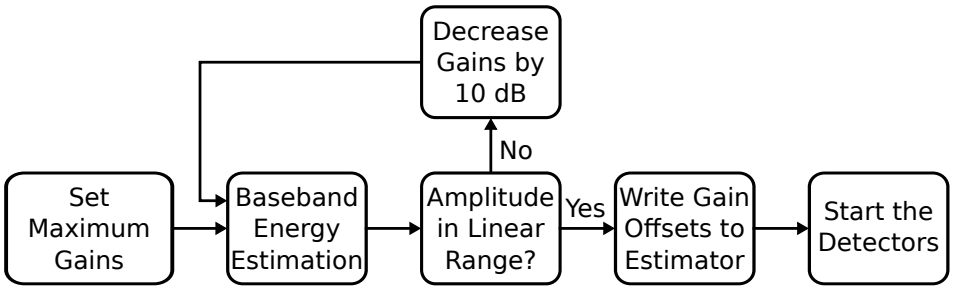

Figure 13: Spectrum sensor prototype platform

Figure 13 depicts the flowchart of the new gain control. It adjusts the signal power at the ADC's input between certain limits by iterating through discrete gain steps. First, after the frequency is changed, the gains are set to maximum. The RSSI is read from the estimator and if falls above a certain threshold, the gains are decreased by 10 dB. This iteration continues until the estimate is below the threshold, which indicates that the signal power is within the linear range of the RSSI estimator and strong enough to cause deterministic detection probability in feature detectors.

The analog signal includes three amplifier stages: two for radio frequencies and one for baseband. First stage is the Low Noise Amplifier (LNA), which can be only switched on or off. The second stage is a Programmable Gain Amplifier (PGA) with roughly 400 digitally controlled gain steps. Both of the aforementioned stages are controlled by the software. The gain of the baseband amplifier was chosen to be set at maximum at all times in order to avoid additional overhead in the adjustment. Although this configuration slightly degrades the noise figure of the entire signal chain, it improves the linearity by allowing lower RF gains. Since the non-linearity is the dominant issue in the spectrum sensor prototype, and certain amount of noise is assumed by the feature detector (see section 5.1.2), maximum baseband gains proved to be an ideal setting. For receiver applications such as mobile TV, it would be more beneficial to optimize the gain of each gain stage to obtain as pure signal as possible.

Figure 14 depicts the amplifier gains for chosen steps. The magnitude of each step was chosen to be 10 dB, since the feasible input range of the RSSI estimator is slightly over 10 dB. Typically, on a vacant channel, the first gain step is sufficient and the adjustment scheme does not cause any overhead. Furthermore, taken into account the settling time of the original AGC loop, the new implementation also proved to outperform it in terms of overall delay and accuracy. The C-code for the adjustment software is presented in appendix A.

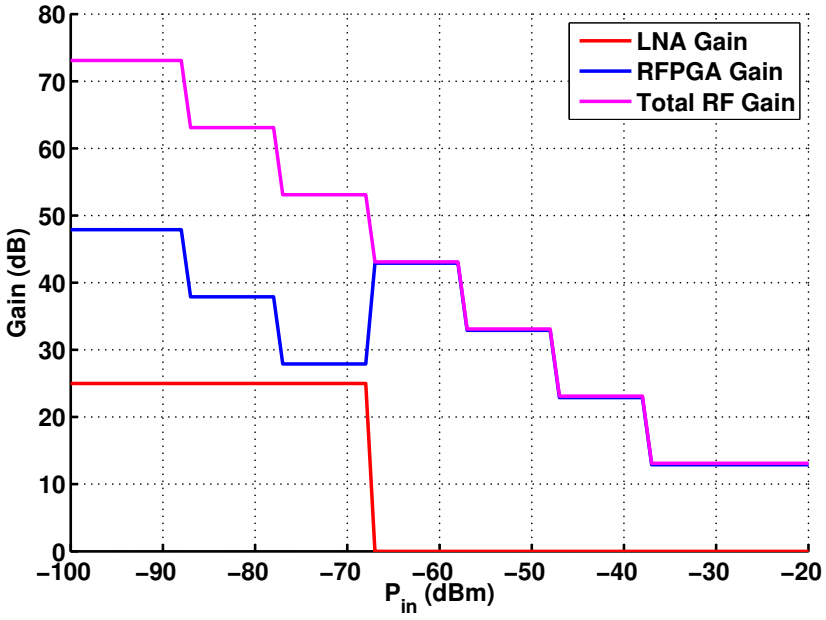

Figure 14: Gain settings as a function of input power

4.3 Implemented Sensing Algorithms

In this section, the implemented detection algorithms are described in detail. Part of the implementation is based on earlier work, and the goal was to optimize their operation considering energy consumption and simplicity and make them suitable for coarse-fine algorithm.

First, the implementation of the Energy Detector (EDT) is described. The EDT is based on previously implemented RSSI estimator and control software that computes the front-end gains to the results. In addition, a comparison to a threshold was added to enable the EDT to output binary decisions. Second described algorithm implementation is the Spatial Sign Cyclic Correlation Detector that is implemented with Angular Domain calculation (ADSSCCDT) and uniform distribution test for deriving the test statistics. These algorithms were chosen due to their energy efficiency. For the binary decision, a normalization mechanism and comparison with threshold for the test statistics were also implemented. The base of the ADSSCCDT implementation is described in [1].

4.3.1 Energy Detector

The implementation of the energy detector is based on formerly implemented Received Signal Strength Indicator (RSSI) estimation block. The goal was to improve the estimator so that it takes the front end gains into account as well as to implement the binary comparison test for the final estimation result. It was also required to retain the ability to read RSSI estimates without saving the detection results. Figure 15 depicts the complete implementation of the component.

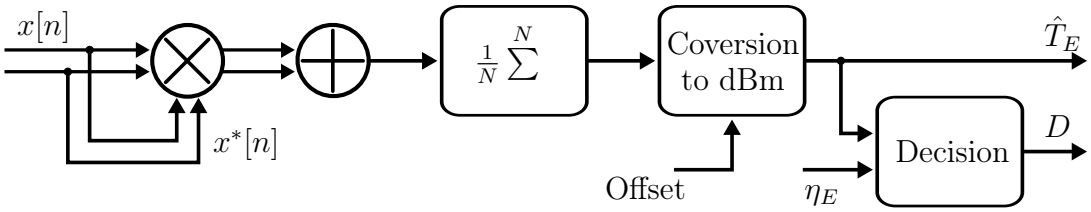

Figure 15: Implementation of the energy detector

First, the energy estimate of the baseband signal is computed as a square sum of complex input samples (see 2.4.1). The estimate is converted to dBm with a Look-Up Table (LUT). The next block in the signal chain applies the front-end gains to the output. Since this feature was not included in the original RSSI estimator did not take the gains into account, the RSSI readings were correct only with very weak signals, and its dynamic range was thus very constrained.

As explained in 4.2, the control software is constantly aware of the current gain settings, which greatly simplifies the implementation of the correction. The RSSI estimator block features an input for the offset, so the value based on the front-end gains can be directly written into it to get correct output from the estimator block itself. In order to derive the input power from the raw baseband estimate, the following equation has to be met:

$$Pin = RSSI - G_{RF} - G_{BB}. \tag{23}$$

For example, if total front-end gain were 100 dB and the estimator would output 0 dBm without the correction, the corresponding input power would be -100 dBm.

Finally, the binary test is carried out from the corrected input power estimate by simply comparing the estimate to externally supplied threshold (see 4.4).

The energy consumption of the implementation was estimated to be 0.89 mW. Considering that the detection time of the energy detector is typically very low compared to that of a feature detector, the total energy needed for one detection is practically negligible. Next section discusses more on the energy consumption of the implemented feature detector.

4.3.2 Angular Domain Feature Detector

Theory of operation of the Angular Domain Spatial Sign Cyclic Correlation detector (ADSSCCDT) presented here is originally based on [47]. The base of the implementation is presented in [1]. The goal of this implementation was to simplify the decision process and get a more consistent test statistics by normalizing it before comparison to a threshold. A look-up table (LUT) was implemented so that the threshold for the binary hypothesis test can be mapped from the detection time. Furthermore, the implementation is expanded to support arbitrarily large detection time within the stability limitations of the USB communication.

Figure 16 depicts the ADSSCC calculation block. It computes the spatial sign with the equation presented in 2.4.3. The equation is repeated here for clarity:

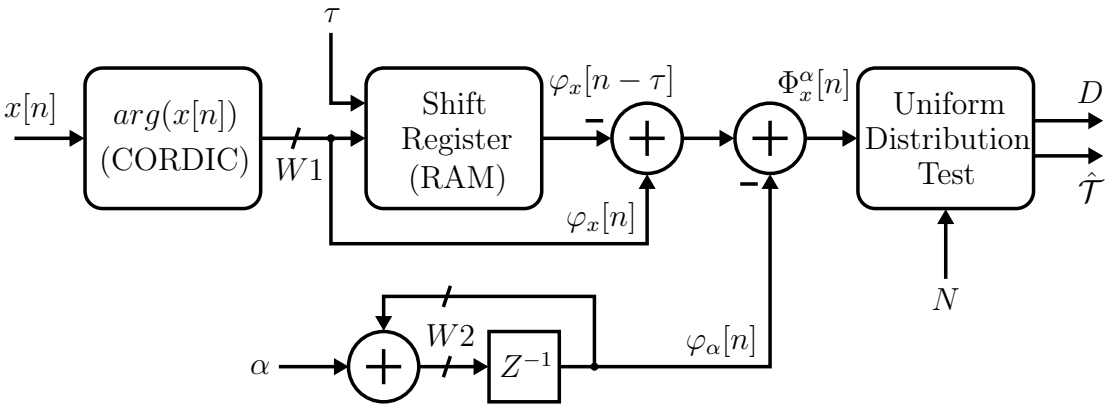

Figure 16: Spatial sign calculator of the ADSSCCDT.

$$\Phi_x^\alpha[n] = \varphi_x[n] - \varphi_x[n-\tau] - \varphi_\alpha[n] \tag{24}$$

The operation goes as follows: First, the phase vector $\Phi_x^\alpha[n]$ is calculated from the cyclic correlation of the spatial sign of the input signal samples. This happens by subtracting the the argument of $x[n]$, that is delayed by lag τ, from the non-delayed argument of $x[n]$, and subtracting the argument of the cyclic frequency α from the result. The argument is computed using CORDIC algorithm [52]. The calculation hardware is depicted in Figure 16 [47]. Parameters α and τ are specific constants for the primary system.

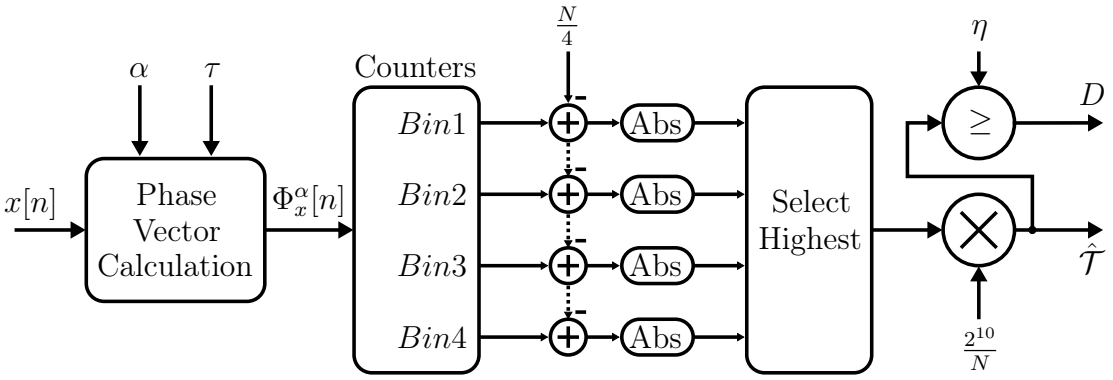

Figure 17: Implementation of the uniform distribution test for ADSSCCDT.

The calculation process of the test statistics as well as the binary hypothesis test is depicted in Figure 17. It is based on the Uniform Distribution Test (UDT), which was briefly introduced in section 2.4.3. Practically, the Test Statistics (TS) is obtained by testing the distribution of the phase $\Phi_x^\alpha[n]$. The phase can be expressed as a vector in the set $[0...2\pi)$, which then can be divided into fixed number of sectors (bins), defined by the chosen amount of bits resembling the phase vector. The amount of bins (M) is chosen to be 4. Hence, the counters may take into account merely the two most significant bits of the phase vector.

Once the predefined amount n_T of phase vectors is calculated, the mean $\frac{n_T}{4}$ is subtracted from each bin and the absolute value of the result is calculated. The final test statistics is obtained by normalizing the count of the bin that deviates most from the mean. The multiplication and division operands are powers of 2, which makes the operations computationally very efficient, since they can be carried out by bit shifting.

The normalization is carried out to be able to represent the measurement metric with reasonable amount of bits, even with the longest detection times. While this simplifies the hardware implementation, it also makes the test statistics more consistent with different detection times, which helps comparisons and visualization. Furthermore, it improves the speed and reliability of the data transfers between the detector and the interfacing device (PC), especially with long detection sequences.

Finally, the binary decision is obtained by comparing the computed test statistics to a threshold, which can is obtained from a LUT. If the TS is below the threshold, the phase is uniformly distributed in the set $[0...2\pi)$ and no signal is detected. Conversely, if the TS is above the threshold, the null hypothesis can be discarded which means that the primary signal is present. The threshold can be derived from the sample amount (detection time) by

$$\eta = \left[P_B^{-1}\left(\frac{1 + (1 - P_{FA})^{1/M}}{2}, N, \frac{1}{M} \right) - \frac{Ns}{4} \right] * \frac{2^{10}}{N_S} \tag{25}$$

It is worth mentioning that due to Shift Register (Figure 16), the spatial sign calculation requires N_τ samples until it can output valid phase vectors. In practice, this adds $T * N_\tau$ to the detection regardless of the total detection time. Furthermore, the data sequences need to be longer than one full symbol (data + prefix) in order to cause correlation. The shortest detection time that fulfills this requirement is therefore 1.8 ms, or 16384 samples.

The energy consumption of the angular domain SSCC detector implementation was estimated to be about 7 mW. While this is significantly higher than that of the energy detector, it is very low compared to several other feature detection algorithms. For example, it was shown in [46] that the energy consumption of an FFT-based cyclostationary feature detector implemented on an FPGA was 61.29 mW while an SSCC detector with complete test statistic calculation operated with 20.9 mW.

4.4 Coarse-fine Controller

In order to control the previously described detection algorithms efficiently, a coarse-fine controller was implemented. This section describes the implementation of a the controller, which is based on the theory presented in section 3.4. A non-ordering, channel-wise scheme was chosen, where the detectors are run in series on each selected channel (see Fig. 10). For this purpose, a separate detector control component was implemented in VHDL. The control block is responsible for starting and stopping both of the detectors as well as timing the writes to result buffer.

During the detections, the energy detector (EDT) is first run on each channel. Depending on its decision, the feature detector is run if needed, that is, when the energy detector declares the channel vacant. In addition, the threshold of the energy detector is constantly adjusted based on the output of the feature detector, which can be considered a variant of noise modeling (see 3.4). Adjustment is done to adapt the energy detector to the uncertain noise floor (5.1.2) and keep the sensitivity at a feasible level. Furthermore, to prevent the threshold getting stuck to a too low value, a random check is carried out with the fine detector frequently, even when the energy detector declares the channel occupied.

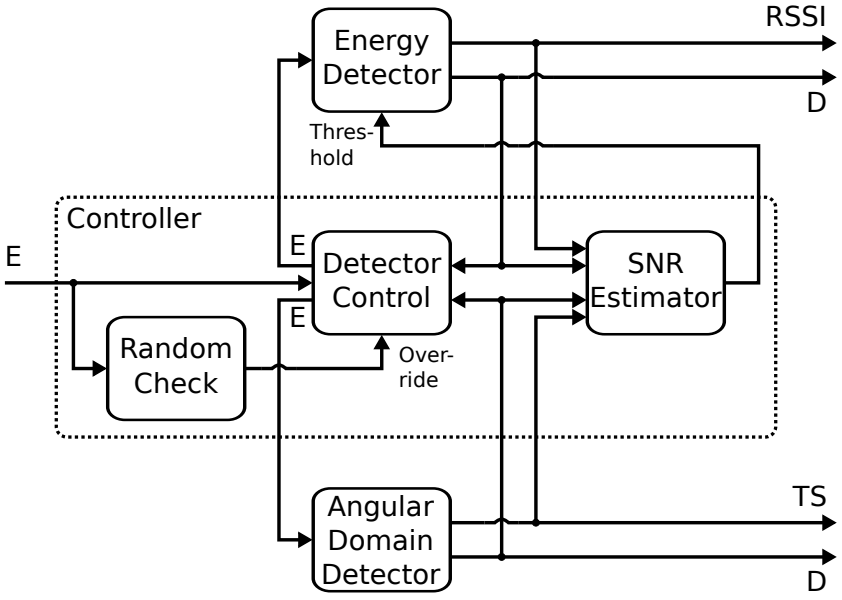

Figure 18: Implemented coarse-fine controller

The function block diagram of the implementation is depicted in Figure 18. The SNR estimator and random check blocks are described in more detail in the following subsections.

4.4.1 SNR Estimator

As mentioned in 2.4.1 and 4.3.1, the energy detector compares received signal power to a threshold in order to make the binary decision on the occupancy of the chan-

nel. In most cases, especially in mobile usage, the radio environment is constantly changing, which causes the noise floor and signal-to-noise ratio (SNR) of the signals to vary over time. For this reason, a fixed threshold would be unfeasible considering the detection probability and keeping the false alarm rate within reasonable limits.

The energy detector alone is not capable of differentiating noise from primary signal, and thus cannot determine the noise floor. Conversely, feature detectors are not aware of the received signal power and their test statistics are mostly dependent of the SNR. Thus, the noise floor estimate can be obtained by combining information from both of the detectors.

The coarse-fine control block presented in this work is responsible for adjusting the threshold for the energy detector, since it can control both of the detectors and access their results. The threshold η_E is determined by estimating the noise floor from the RSSI and cyclostationary test statistics $\hat{T}_C$ and adding a predefined marginal to it. Table 1 depicts the adjustment scheme.

Table 1: Threshold adjustment scheme

EDT	**ADDT**	**Action**
0	0	η_E = RSSI + M
0	1	Decrease η_E if $\hat{T}_C >$ K
1	0	Increase η_E
1	1	No action

If both decisions are negative, the received signal can be assumed to be noise and thus the noise floor equals the estimated energy of the input. Thus, in this case, the threshold value can be directly set to RSSI plus the marginal. If only the feature detector declares the channel occupied but the EDT does not, it is possible that the threshold is too high. However, since the sensitivity of the EDT is acceptably lower than that of the feature detector, the threshold should be lowered only when the SNR is relatively high, that is, when the test statistics of the feature detector is high. However, this feature was not implemented in the scheme since the situation occurred very rarely and thus the minor adjustment was insignificant.

If only the energy detector declares the channel occupied but the feature detector does not agree, the threshold may be too low and needs to be therefore increased by a predefined constant, which in this case is chosen to be 6 dB. Note that this can happen only when the fine detector is forced to run (see next subsection), since positive decision of the EDT would normally prevent the fine detection. Finally, if both decisions are positive, the threshold is in the right range and no adjustment is required.

Simulations for determining optimal value for the aforementioned threshold marginal were carried out by post-processing measurement data in MATLAB. Real data from field measurements was used for the analysis, and only the detection scheme was synthesized in the simulation. Detections on channels 44, 46 and 53 were considered true positives. Since those channels are strong in the measurement area, energy

detector can always detect them and thus reach 0% false negative rate (FNR). Also, since part of the measurements happened in the coverage of different transmitters, the definition of false detection would be very complicated to derive analytically. Hence, the constant 5% false alarm rate (FPR) was used for the feature detector.

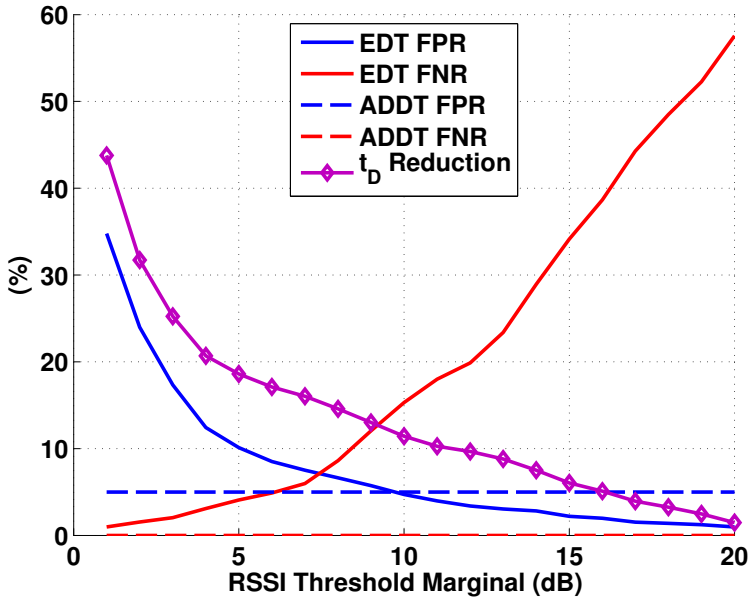

Figure 19: Effect of RSSI marginal above estimated noise floor

Figure 19 depicts the simulation results. The aim was to maximize the detection time reduction while not losing any channel capacity. Practically this is the point where the FPR of the EDT crosses that of the feature detector. According to the simulation, this is roughly 10 dB, which was chosen for the scheme.

4.4.2 Random Check

Another weakness of the energy detector, discussed in 2.4.1, is that it cannot differentiate signals from different systems. For example, if we only want to find vacant DVB-T channel and EDT compares the received signal power to a threshold determined by noise floor, signals from other systems will cause energy detector to indicate that the channel is occupied. Although this is beneficial in some cases, it might lead to missed spectrum opportunities in case the interfering signal is weak and would not interfere with the primary system. This issue can also be addressed with the feature detector.

To prevent the threshold getting 'stuck' below the noise floor, a random check feature is implemented in the coarse-fine control block. In practice, it is a counter that forces the angular detector on every N^{th} detection regardless of the decision of the energy detector. If the threshold is below the noise floor, energy detector will detect the signal but the feature detector may not. As presented in Table 1, this will

force the threshold to increase and eventually make the EDT to correctly indicate that the channel is free.

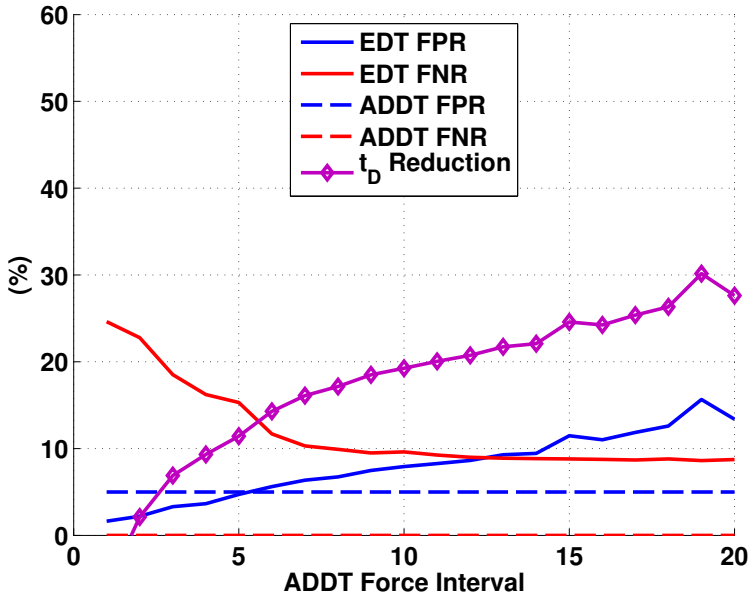

Figure 20: Effect of angular detector's force interval

The optimal value for the threshold was studied with a MATLAB-simulation, similarly to the RSSI marginal. The same criterion was used for finding the optimal value. The interval where the detectors' FPRs cross appears to be 5 and thus the feature detection is forced to run every 5th time the EDT is run.

5 Verification of Functionality

To get an understanding of the actual performance of the implemented functions on the spectrum sensor prototype, measurements with real hardware are needed. The verification of functionality of each implementation described in chapter 4 is presented in this section. The measurement setups and processes are described and the results are compared to theoretical and simulated values. The measurements were carried out with laboratory equipment and by field trials in urban and sub-urban environments.

5.1 Laboratory Measurements

The goal of laboratory measurements was to verify the functionality of the spectrum sensor implementations in a controlled environment prior to field testing. Special attention was paid for the observations whether the operation of the sensor implementations was deterministic and if the obtained results corresponded to theoretically derived assumptions.

The measurements were carried out in a shielded, anechoic measurement room in Aalto University's Electronic Circuit Design unit. A vector signal generator, Anritsu MG3700A, was used for the measurements that required an input stimulus. The test signal was generated in MATLAB software and resembled the Finnish DVB-T system's OFDM-modulated signal that has FFT size of 8192 samples, cyclic prefix of 1/8 (1024 samples) and bandwidth of 8 MHz.

5.1.1 Sensitivity

First measured parameter was the sensitivity of the refined ADSSCC detector algorithm (sections 2.4.3 and 4.3.2). The probability of detection (P_D) was measured as a function of the input power with 9 different detection times. Table 2 presents the relevant parameters that were used in the measurement setup. The detection times and their corresponding sample amounts are also presented in the table.

Figure 21 depicts the results for the measurement. As mentioned in section 4.3.2, the minimum possible detection time for the algorithm is 1.8 ms when sensing a system where the length of one symbol is 1 ms, as there would otherwise be no correlation. However, the maximum sensing time is only limited by the hardware, i.e. how large values the counters can hold and how many detection results the FIFO buffer can store. Within those limitations, the sensing time can be arbitrarily set to any value that is 1.8 ms multiplied by a power of 2.

According to the obtained results and simulations, doubling the detection time effectively increases the sensitivity by 1.5 dB. If the sensitivity requirement regulated by FCC was to be met with only a single detection, the detection time should be at least 458.4 ms. Taken into account the possible interference present in the real radio environment, even longer detection time would be needed. Clearly, such detection time would be unfeasible, especially for systems with wide frequency range, since the overall detection time and subsequently the energy consumption would increase

Table 2: Parameters for sensitivity measurement

$\mathbf{N_{Samples}}$	$\mathbf{t_{Det}}$	**Common Parameters**	
2^{14}	1.8 ms	f_{Center}	698 MHz
2^{15}	3.6 ms	N_{Tests}	250
2^{16}	7.2 ms	P_{FA}	0.05
2^{17}	14.3 ms	RF Gain	73 dB
2^{18}	28.6 ms	BB Gain	42 dB
2^{19}	57.3 ms	Modulation	OFDM
2^{20}	114.6 ms	N_{FFT}	8192
2^{21}	229.2 ms	CP	1/8
2^{22}	458.4 ms	BW	8 MHz

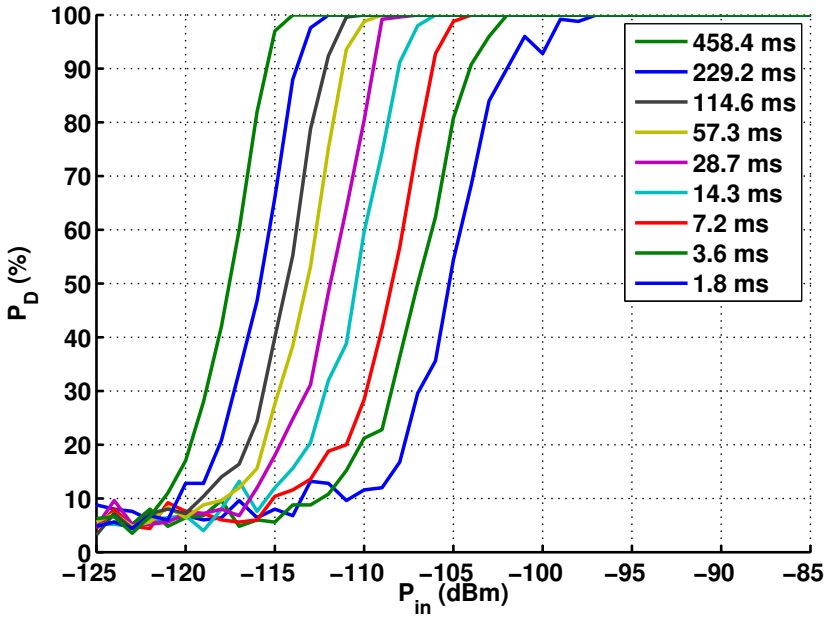

Figure 21: Detection probability as a function of input power for different detection times.

to unacceptable levels. Thus, other methods such as cooperation are needed so that single sensor nodes would suffice with a shorter detection time.

Sensitivity measurement with the same hardware platform using an FFT-based cyclostationary feature detection (CYC-FFT) algorithm is presented in [3]. There, 90% detection probability is reached with the input power of about -117 dBm. Thus, ADSSCC algorithm falls only about 1.5 dB short of the sensitivity with corresponding detection time, while needing only about one tenth of the power needed by CYC-FFT [46]. Even though this means that the detection time needs to be doubled to achieve same sensitivity, it can be concluded that the ADSSCC is significantly more

energy efficient compared to CYC-FFT.

Another observation from Figure 21 is that the detection probability settles very close to 5% below the minimum sensitivity of the algorithm with respect to the detection time. Therefore, considering the minor noise uncertainty and the various non-idealities of the used hardware, it can be concluded that the false alarm rate corresponds satisfyingly to the theoretical 5% that was used for the calculation of the detection thresholds.

5.1.2 Non-idealities

An overview of possible non-idealities present in a real hardware was given in section 3.1.1 and the most relevant of them considering the sensor prototype was specified in section 4.1. The effect of these non-idealities is verified here and methods for mitigating them are discussed.

Non-linearity

Non-linearity of the RF front-end was measured by feeding a DVB-T signal on channel 49 and observing the detections on three adjacent channels (50, 51 and 52). Similar measurement with the same platform but with different sensing algorithm (FFT-based CFD) was published in [5]. The measurement was repeated for the new detector and gain controller implementations.

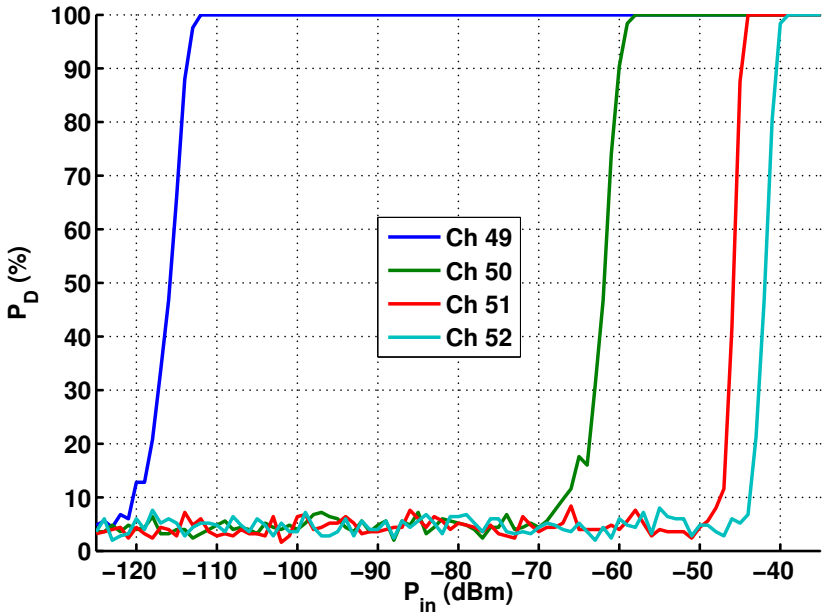

Figure 22: Adjacent channel measurements: sensing time set to 229.2 ms, DVB signal transmitted on channel 49.

As Figure 22 indicates, signal is detected from adjacent channel when the amplitude over -60 dBm and even from 4 channels offset, when the signal amplitude is -40

dBm. This is primarily consequence of 3^{rd} order nonlinearity, and presumably the detections on channels 51 and 52 are also caused by 5th and 7th order nonlinearity. The self-mixing of the multicarrier input signal (spectral regrowth) causes the false detections on the adjacent channels, since the modulation products embody the same cyclic features as the input signal (see 3.1.2).

In addition to causing false alarms, the spectral regrowth exacerbates the sensitivity of the detector on channels where the modulation products fall by increasing the noise floor. Even though the false alarms could be mitigated by increasing the threshold when a strong signal is present on the adjacent channel, that is often not beneficial since it would allow a weak primary signals to go undetected under the increased noise floor.

Noise Uncertainty

It was stated in section 2.4.3 that the angular domain spatial sign function assumes the amount of zero-samples to be negligible in order to be able to discard the radius from the individual samples (equation 16). In addition, the noise is assumed to be AWGN in order to achieve the derived false alarm rate of 0.05.

It was perceived in the field measurements that having too low gains leads to unexpected false alarm rates. Hence, this phenomenon was measured by observing the detection probability as a function of front-end gain without any input signal. The antenna feed connector was ended with a 50 ohm stub to ensure no signal was coupled to the connector. Figure 23 depicts the results from the measurement.

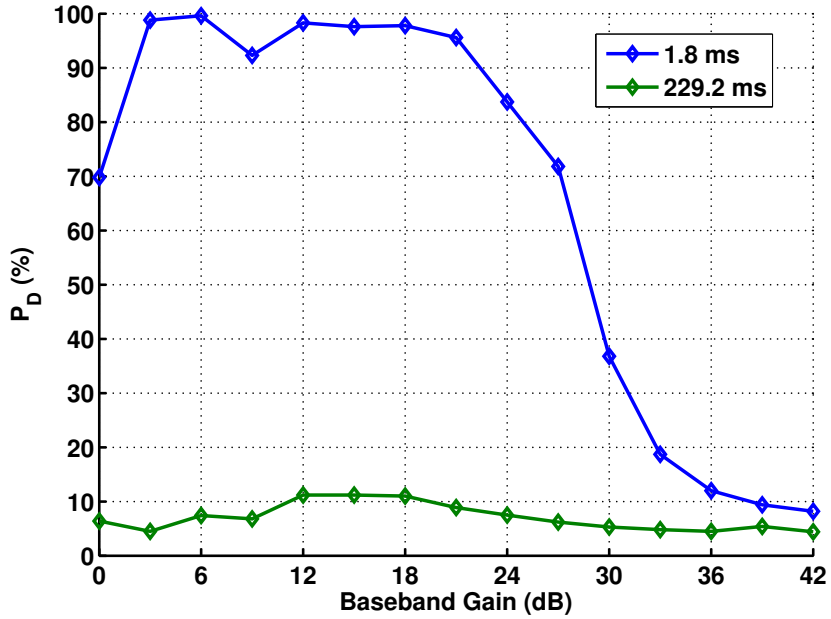

Figure 23: Probability of detection with ADT as a function of input gain with no input signal.

According to the results, the effect is very strong when detecting with the short-

est detection time (1.8 ms). False alarm rate of almost 100% can be observed below the middle point of the gain range. Furthermore, as the gains approach the maximum (42 dB), the detection probability approaches the assumed 5% but does not completely reach it, since the noise is still not ideal AWGN. However, the difference from assumption can be considered negligible compared to the other parameters and given the very short detection time. This also leads to a conclusion that it is crucial to set the amplifier gains at maximum and preferably use longer detection time on the empty channels in order to get deterministic results.

It can also be noted from the graph that using a long detection time, such as 229.2 ms, clearly mitigates the false alarm issue, albeit not completely removing it. The reason for this phenomenon is that the correlation of noise converges to zero as the number of samples increases.

Spurious Emissions

Most part of the unexpected (non-deterministic) behavior was identified to be originated from the coupling of spurious emissions from the electronics. Measurements revealed that when the antenna is in the proximity of the FPGA, a correlating signal is coupled to the detector input. This appears to cause significant increase in the false alarm rate.

The phenomenon was measured and characterized with the aid of an anechoic Faraday's cage. Measurements with three configurations were carried out. First, the antenna was placed inside of the cage and the detector board outside of it. This measurement indicated that no signals were coupled to the input of the front-end through other paths than the antenna. Next, both the detector and the antenna were placed inside of the box in order to find out the spectral locations of the spurious emissions. Finally, it was tested how the real-world measurement results look without the effect of spurious emissions by placing only the detector board inside of the box and antenna outside of it. Figure 24 depicts the measurement results when both the antenna and the detector were inside of the Faraday's cage.

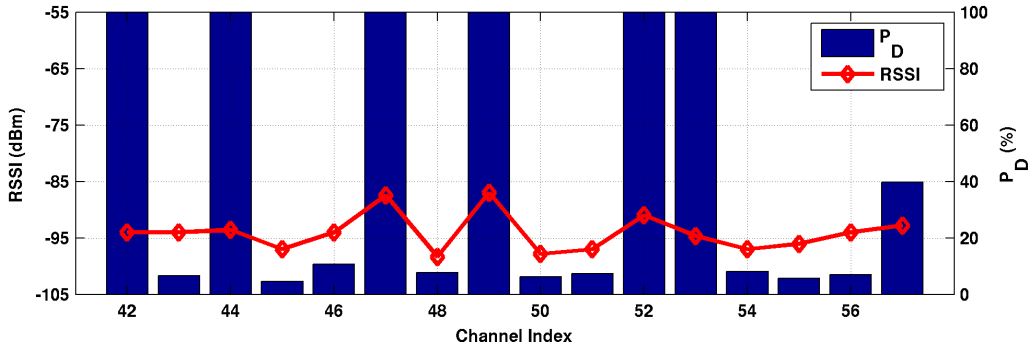

Figure 24: Probability of detection with detector and antenna inside of a Faraday's cage.

It is also worth mentioning that the effect increases as the detection time is decreased. As was explained in the previous subsection, by correlating a longer sequence of samples, the unwanted signals are not showing an increase in the test statistics. As can be seen in Figure 24, the detection probability on the correlating channels is 100% when the detection time is set to 3.6 ms.

The identified issue emphasizes the importance of the form factor and top-level designing of mobile hardware. The shape, placement and orientation of the antenna given the form and material of the enclosure may have great impact on the amount of false alarms and thus on overall detection performance. In addition, the components, their placement and the parameters need to be chosen so that the spurious emissions with similar features than the wanted signals would be minimized.

Stabilization Delays

Another front-end originated issue is related to the stabilization times of various internal blocks, namely the internal Automatic Gain Controller and the Voltage Controlled Oscillator (VCO) were identified to cause fluctuations to the detection results.

The front-end chip utilizes two separate VCO's to cover the whole UHF frequency range and the swap between the VCO's happens roughly in the middle of the range. The stabilization of the oscillators measurably caused fluctuations to the signal and thus the RSSI as well as test statistics were severely distorted during that period. The issue was addressed by increasing the delay before detections to 400 ms after the VCO swap, in order to cover the worst-case settling times.

The internal AGC was also identified to be relatively slow to settle, which caused unacceptably large variance in the RSSI values. The first approach to cope with this issue was to lock the gains after they are sufficiently stable, as this would be an intuitive solution. However, the settling time appeared to be much longer than assumed and have a very large variance. Hence, the gains were occasionally not stabilized prior to locking, which led to an error in the RSSI readings. Furthermore, the gains could settle to a too low value, which caused false alarms on empty channels and ultimately degraded the overall sensitivity. Thus, the software-based AGC had to be implemented (see 4.2).

One plausible explanation for the inconvenient settling time is that the front-end chip is primarily intended for mobile TV reception instead of rapid spectrum sensing. This again puts an emphasis on the importance of the component choices for the sensing hardware, even when it is constrained by the limitations of mobile devices.

5.1.3 Received Signal Strength Indicator

Accuracy of the Received Signal Strength Indicator (RSSI) is a major contributor to the sensitivity and reliability of the energy detector. Hence, the estimation output was measured as a function of input power.

An input signal that resembles DVB-T signal was generated in MATLAB and fed to the input of the detector from the signal generator. Input powers were swept

from -120 dBm to -30 dBm on 2 different frequencies in order to cover the feasible input range. Measurement results are depicted in Figure 25.

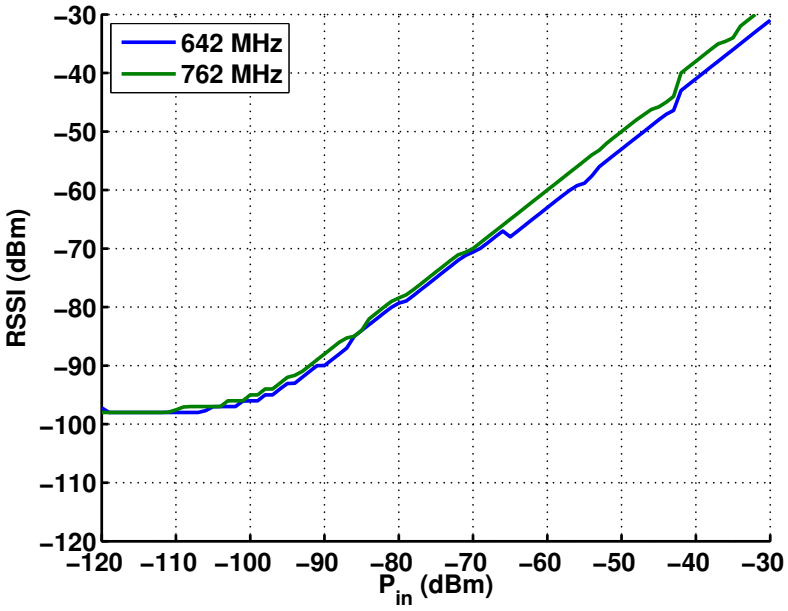

Figure 25: RSSI as a function of input power

The measurements are carried out for frequencies 642 MHz and 762 MHz, since these are the frequency boundaries for majority of the measurements presented in this work. For input power range from -90 dBm to -30 dBm, the accuracy appears to be approximately ± 3 dB. The inaccuracy mainly caused by the nonlinear frequency response and slightly inaccurate amplifier gains of the front-end chip. Below -90 dBm, the RSSI value starts to saturate towards noise floor, which appears to be about -98 dBm with full gains, given the noise figure of the receiver and the inaccuracy of the estimation. Furthermore, it can be assumed that the input power does not exceed -30 dBm during normal usage.

5.2 Field Measurements

Verifying the functionality and performance of the implementations in the field is of great importance, since laboratory measurements do not necessarily reflect real-world conditions. For example, non-deterministic channel conditions, unknown interference, shadowing and multipath fading (3.2) are not affecting in laboratory test signals. Furthermore, laboratory signals are fed directly to the RF input of the sensor board, and thus the effect of antenna (3.1.3) can only be seen in field measurement results.

There are two DVB-T transmitters that clearly affect the the measurement results. The other one is located in Espoo and is transmitting channels 44, 46 and 53 with high power. Since the measurements are carried out in the capital area of Finland, including Espoo, the aforementioned channels can be assumed to be reliably

detectable in each of the measurements. The other transmitter is located in Tallinn, Estonia, which is approximately 80 km away over the sea. It transmits channel 45, which can be detected mainly near the shore and on high ground. Section 5.3 presents an illustrative coverage map for the channel. The transmitter parameters are presented in Table 3.

Table 3: DVB-T transmitter parameters [5]

DVB-T Transmitters	**Espoo**	**Tallinn**
Latitude	60.1778	59.4713
Longitude	24.6403	24.8875
Mast Height	313 m	272 m
Transmission Power	47 dBm	42 dBm
Transmitted Channels	32, 35, 44, 46, 53	45, 59, 64

5.2.1 Detection Time Comparison

The effect of the detection time was also tested in a real radio environment in addition to lab measurements presented in section 5.1.1. This time the maximum detection time was limited to 57.3 ms since it would have been unpractical to obtain sufficient amount of samples with longer times in the field, and the given set of detection times clearly verifies the assumed effect.

The measurement was carried out in campus area in Espoo. Channels 42 to 57 were measured simultaneously and 500 samples per channel was obtained in order to mitigate the effect of unknown interference. It is known that channels 44, 46 and 53 are transmitted from Espoo with high power and thus should be detected reliably. Channel 45, on the other hand, is transmitted from Tallinn and is therefore very weak in Espoo region and detectable only near the coast of the sea and on the high ground.

Figure 26 depicts the detection probabilities on each measured channel with their corresponding detection times. As the detection time is increased, the detection probability on channel 45 is clearly increasing, which implies that the primary signal is actually detected, albeit with certain unreliability. Detection probability on the channels that are transmitted from Espoo is exactly 100% without exceptions, as it was assumed. Furthermore, 5% detection probability is achieved on empty channels, excluding indexes 43, 47 and 49.

The slightly increased detection probabilities on channels 43 and 47 do not behave deterministically with respect to the detection time. Thus, the phenomenon can be assumed to be consequence of self-modulation of the signals on channels 44 and 46, as explained in 3.1.2. Another possible cause for the detections is the spurious emissions from the sensor hardware, since it was shown in 5.1.2 that channels 47 and 49 are among the affected channels. Third possible explanation would be another transmitter located in Lohja, which is approximately 50 kilometers away from the

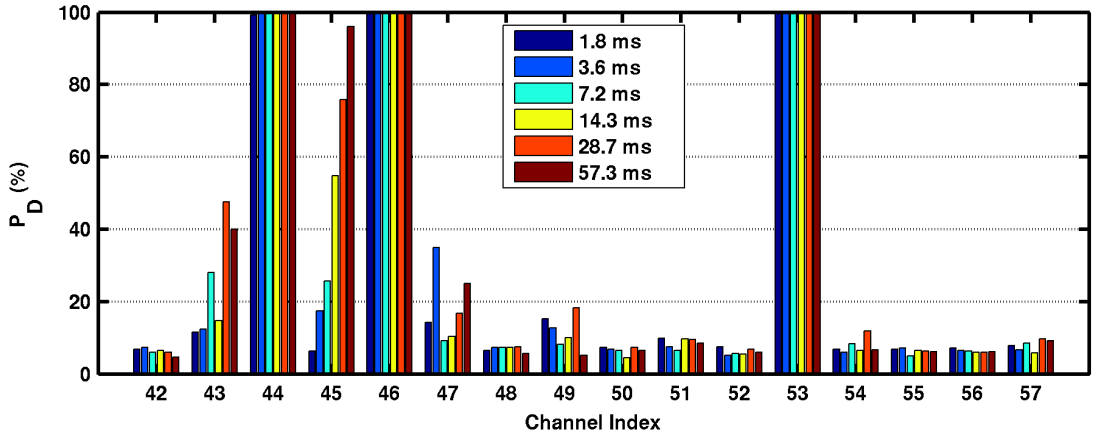

Figure 26: Probability of detection with different detection times -Add 114 and 229 ms?

measurement location. The transmitter broadcasts on channels 47 and 49, but since the transmission power is relatively low, this can be considered the least plausible cause for the occasional detections.

5.2.2 Antennas and Stability

The antennas were replaced in the beginning of the project by new isotropic antennas in order to mitigate hardware originated issues, such as spurious emissions. Thus, a validation measurement on the antenna's performance was conducted. Placing the antenna close to the detector board indicated similar correlating emissions than the measurements in Faraday's cage, and the optimal location for the board was found out to be in the position of the normal of the detector's ground plane, which implies that a grounded surface can mitigate the issue significantly and should be considered in mobile device layout design. Naturally, placing the antenna as far away as possible from the board reduced the amplitude of the emitted signals. This configuration was used with all the field measurements so that the results would not be distorted by hardware-specific issues.

Figure 27 depicts the RSSI values in time domain and RSSI as well as test statistics values in frequency (or channel) domain when measuring with the aforementioned setup. The noise floor appears to be -97 dBm which corresponds to results obtained from the laboratory measurements. The RSSI on the reserved channels (44, 46 and 53) is approximately -65 dBm with about 4 dB variance, which is sufficient and even outperforms the old antenna.

5.2.3 Coarse-Fine Scheme

The goal of the measurements presented in this section was to verify the achievable benefit from coarse-fine controlling scheme, described in section 4.4. Total detection time and energy consumption were compared to a case where no coarse-fine control

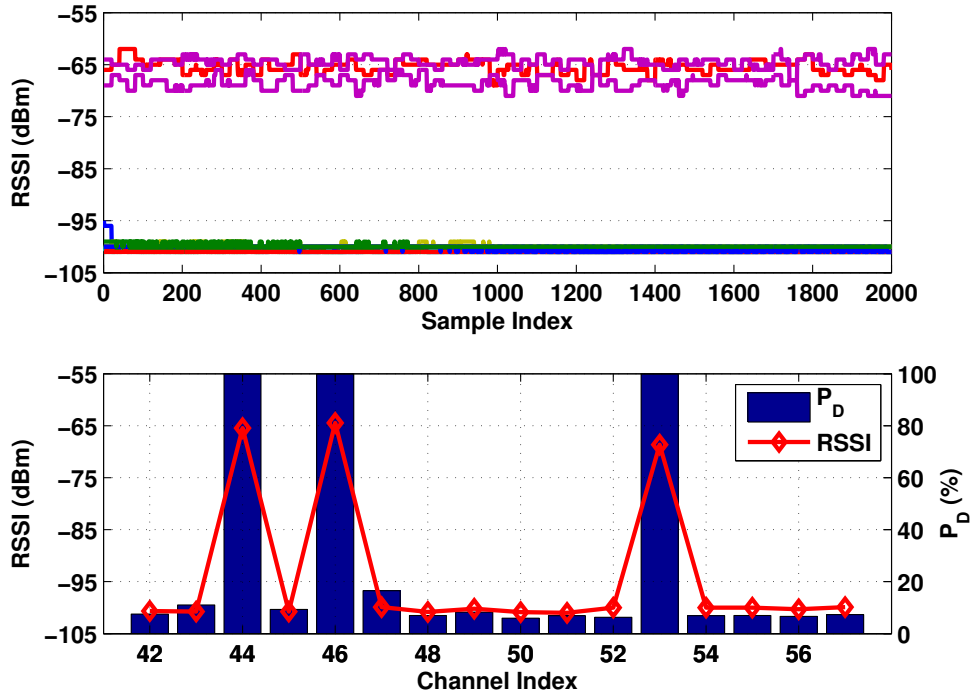

Figure 27: RSSI stability with the new antenna in time and frequency (channel) domain.

exists and the detection relies only on the feature detector.

The measurement setup includes a laptop with USB-communication, which causes a significant latency issue. To get the measurement results correspond to realistic implementation scheme, only the actual detection times are used for the calculation of the results, and the communication-related overhead is ignored. Hence, total detection time is calculated with MATLAB from the detection times of the individual detectors, instead of using time stamps of the results.

Two different schemes were compared: one that starts the detectors in parallel and stops the fine detection if signal is detected with coarse detection, and one that starts the fine detection after the coarse detection if no signal was detected. Logically, starting the detectors in parallel yields better gains regarding the overall detection time, since part of the fine detection is done while energy detector finishes. However, starting the detectors in series can yield even better energy savings compared to parallel case. The difference is less significant when detection time of the coarse detector is much less than detection time of the fine detector. This is discussed in detail later in this section.

It should be noted that the fine detector used in these measurements (ADSSCC) already features very low power consumption, so the achievable benefit regarding the total energy consumption would be even more significant with an energy consuming fine detection algorithm, such as SSCC with complete test statistics computation or an FFT-based cyclostationary feature detector [46]. Furthermore, the sample

amount used for energy detection is very high compared to the minimum feasible amount, since accurate RSSI estimations are of interest in this work.

Assumed parameters for the coarse-fine measurements are presented in Table 4.

Table 4: Parameters for coarse-fine scheme

Parameter	**Value**
P_{EDT}	0.89 mW
P_{ADT}	7.25 mW
$t_{Det,EDT}$	1.8 ms
$t_{Det,ADT}$	Variable
Force Interval	5
RSSI Marginal	10 dB

The transmitter parameters are the same as presented earlier in Table 3. The occupancy level in Espoo region is relatively low, which again leads to lower benefits in terms of energy and delay (see 3.4). This time, channel 45 (Tallinn Tx) was considered true positive in addition to channels 44, 46 and 53 (Espoo Tx). Thus, the occupancy level is 25% when the sensing range is 16 channels.

Figure 28 depicts the total detection time and energy consumption compared to standalone fine detection, which was the reference case.

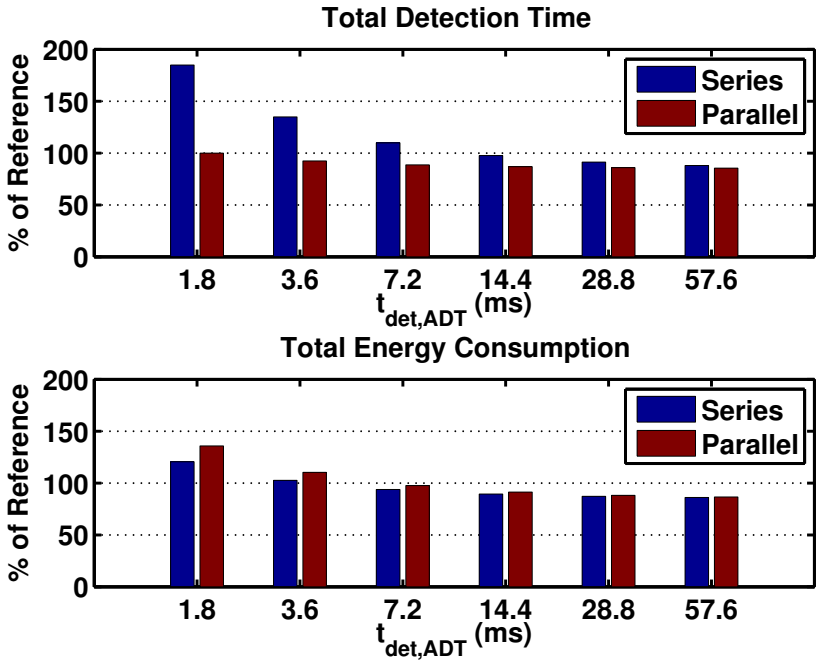

Figure 28: Achievable benefit of the coarse-fine scheme concerning detection time and energy consumption

With such low occupancy, the maximum achievable benefit from CF-scheme is close to 20%, as Figure 28 illustrates. The detection time for the energy detection was set equal to the lowest possible detection time of the feature detector, in order to see how it affects the achievable benefit. When the detection time of the fine detector approaches the detection time of energy detector, both energy consumption and detection time are actually worse compared to the reference case.

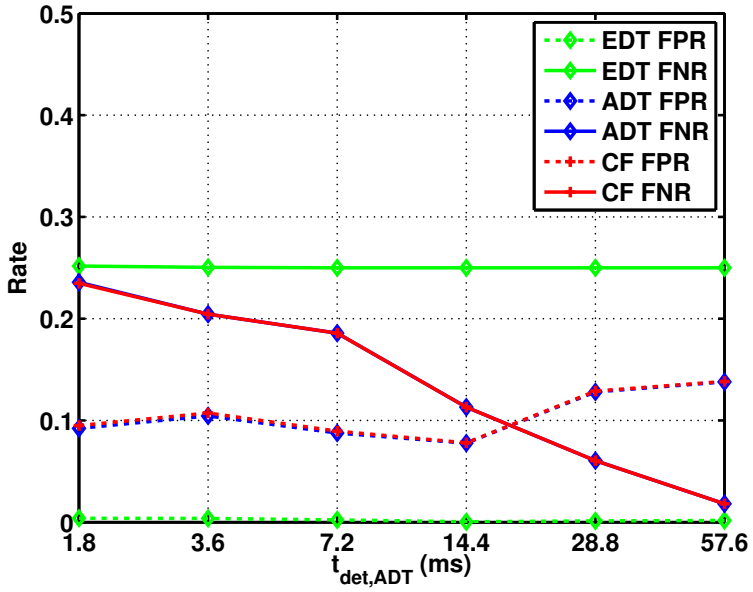

Figure 29: False negative and false positive rates of energy and angular detectors independently, as well as combined rates with coarse-fine control scheme

In addition to energy consumption and detection time rates, the final detection accuracy was measured by observing the false positive and false negative rates (FPR, FNR) of the detectors individually and combined. As can be seen from Figure 29, the FPR and FNR of the overall coarse-fine scheme follow very closely those of the angular detector. The reason for this is the pessimistic behavior of the energy detector: when the threshold is high, all the potentially vacant bands are scanned with the fine detector and only the certainly occupied channels are declared occupied with the energy detector.

If the threshold of the energy detector would be set low enough to detect channel 45, the overall detection time would decrease, but the amount of false positives would increase, leading to lost channel capacity. In presence of shadowing and multipath fading, as well as significant noise uncertainty, such threshold could lead to severe degradation in overall performance. Hence, the pessimistic behavior is justifiable; detection time and energy consumption are reduced while detection performance is not affected.

5.3 Spatial Interpolation

This section presents a practical application of a statistical spatial interpolation method, Kriging (see 2.6). The interpolated surface is calculated from real field measurement data that is measured with mobile spectrum sensor prototype (4.1). The focus of this experiment is on finding the best approximation surface for non-linear cyclostationary test statistics from the angular domain detector (ADDT) (2.4.3). The results are compared with interpolation of energy estimates from the energy detector (EDT) (2.4.1) to evaluate the feasibility of the interpolation as well as the differences of the detector algorithms. MATLAB and mGstat toolbox [81] were used for the calculation of the interpolation.

The field measurement campaign was carried out in suburban Espoo region, where four sensors were used in 31 different measurement locations covering almost uniformly the measurement area. The sensors were spread approximately 100 meters apart from each other and a total of 400 samples from both of the detectors (EDT and ADDT) were collected at each location. Detection time of 14.4 ms was used for the ADDT.

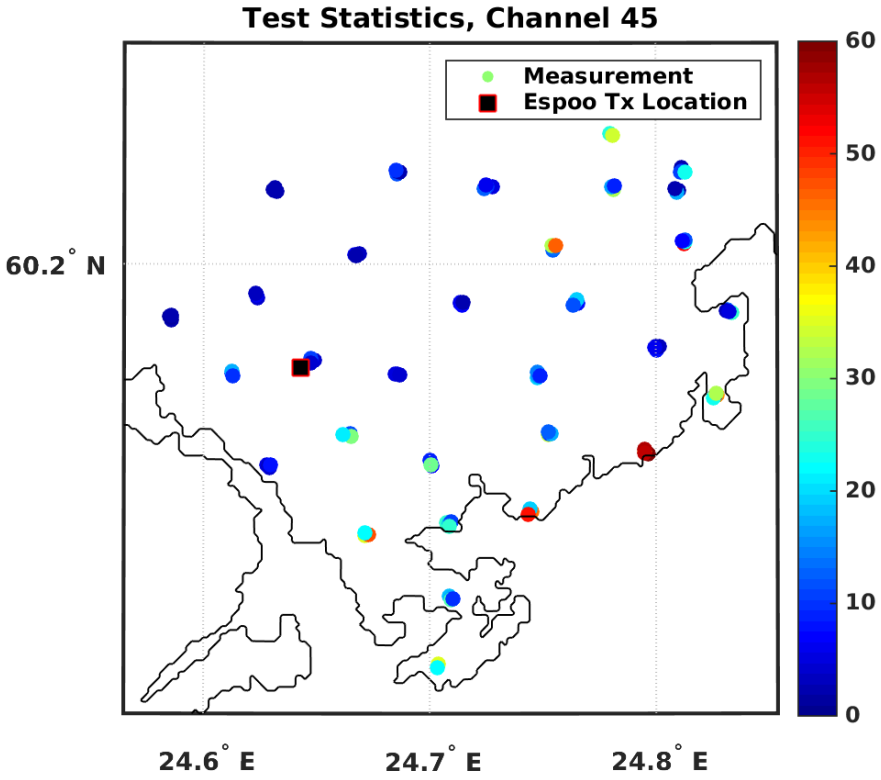

Figure 30: .

Figure 30 depicts the individual measurement results from ADDT on a map of Espoo region. Clearly, the measurement results by themselves are not much of use in database-centric spectrum sensing since the whole area is not uniformly covered. It can be safely assumed that by simply using the closest measurement point does not

give the best approximation for the local occupancy, as this method would not take the spatial nature of the signal into account at all. Instead, a more sophisticated method, such as Kriging, is needed for creating the estimates for uncovered areas.

5.3.1 Interpolation of Cyclostationary Test Statistics

There are several reasons why the test statistics of a cyclostationary feature detector should be preferred to that of an energy detector (see 2.4.2), especially in non-ideal conditions. Noise uncertainty, random interference, hardware inaccuracies as well as error caused by interpolation all together make the interpolated energy estimates a highly unreliable reference for signal detection. In addition to increased robustness, the feature detectors have significantly higher sensitivity, which can also lead to increased spatial correlation in areas where signal strength is low.

Interpolation of cyclostationary test statistics has not been studied widely in the current literature, since it embodies features that may exacerbate the accuracy of the interpolation. Firstly, the value does not depend linearly on the signal power, which is naturally the varying parameter in signal detection. Secondly, the values saturate at very low and high signal powers. And finally, the variances are heterogenous, although interpolation methods such as Kriging expect constant variances along data points.

However, some publications argue that Kriging is reasonably robust against heterogenous variances [82]. Furthermore, the non-linear behavior and saturation of the values can be compensated at least partially. The estimations can be calculated only in the areas, where a primary signal is known to be possibly present, but very weak. These are basically the areas that we want to focus on; in clearly occupied or unoccupied areas the occupancy information can be obtained directly from a database, without any interpolation. The saturated and nonlinear values can also be taken into account by fitting an appropriate variogram model to the data.

Fitting a Variogram Model

The basic idea of variogram fitting was explained in 2.6. Here, we attempt to find a theoretical model based on the observed data and finding optimal parameters through least squares fitting. Two cases are tested for generating the semivariances: one that uses every data point for calculation, and one where saturated values have been filtered out. The best approximation is picked based on crossvalidation (see next subsection). The saturation point was decided by observing the histogram of the individual detection results. All the individual values that were below that threshold were simply left out from the averages that were used for variogram modeling.

Although the use of a variogram cloud, which depicts the semivariances of each location pair at the same time, is suggested by some authorities [61], it is arguably very difficult to interpret and use for finding an appropriate theoretical model. Hence, a *binned variogram* which corresponds to moving average of the data was used for clarity. After filtering the corrupted data out, the total amount of measurement

locations settled at 117 , which corresponds to 6903 location pairs. The amount of bins was chosen to be 100 in order to make the data readable.

After observing the semivariances with different parameters, the exponential model was found out to be the best approximation. Taking the *nugget effect* into account, the variogram model takes the form

$$\begin{aligned} y(h) = &\quad x_0 + x_1(1 - exp(\tfrac{-3h}{x_2})) \quad |h > 0 \\ y(0) = &\quad 0, \end{aligned} \tag{26}$$

where, in terms of variogram modeling, x_0, x_1 and $x2$ correspond to nugget, partial sill and effective range, respectively.

Now, the optimal parameters for $x0, x1, x2$ can be obtained from Non-linear Least Squares (NLS) fitting [83], which is based on iteratively minimizing the errors of each variable. Figure 31 shows the averaged semivariances of the data as a function of distance, or in this case, distance classes.

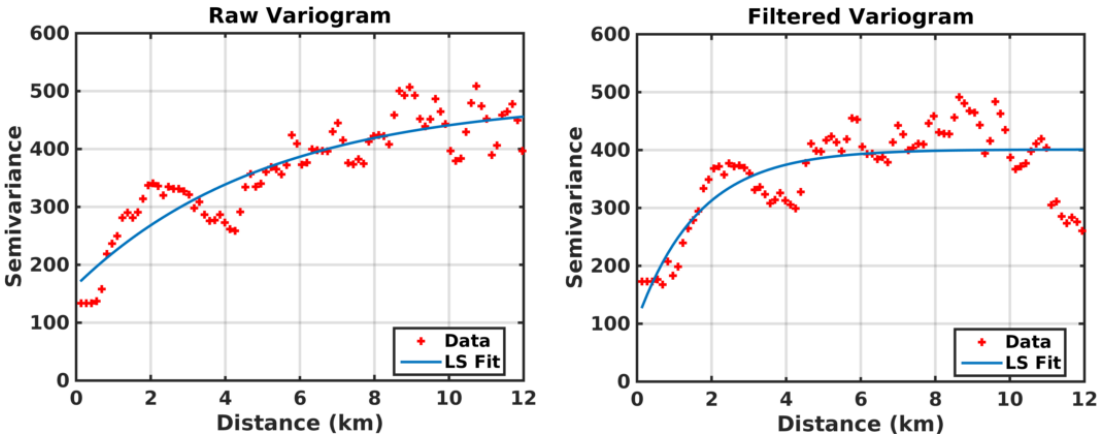

Figure 31: Variograms of cyclostationary test statistics with unfiltered and filtered data.

A window size of 1.5 km was used in the figures for the calculation of the moving average. It was chosen by carrying out the curve fit with different window sizes and observing the behavior of the parameters, so that the parameters do not vary much with surrounding window sizes. The fitted parameters as a function of the window size is depicted in Figure 32.

Table 5 presents the obtained parameters that can be used for the actual interpolation with Kriging.

Crossvalidation of Methods

A crossvalidation was carried out in order to find out which method produces the least error; variogram with all the data or variogram with the saturated data filtered out.

With Kriging, the goal is to get the best estimate in an unknown location, based on all the data in the range. Taking into account that the data may have local

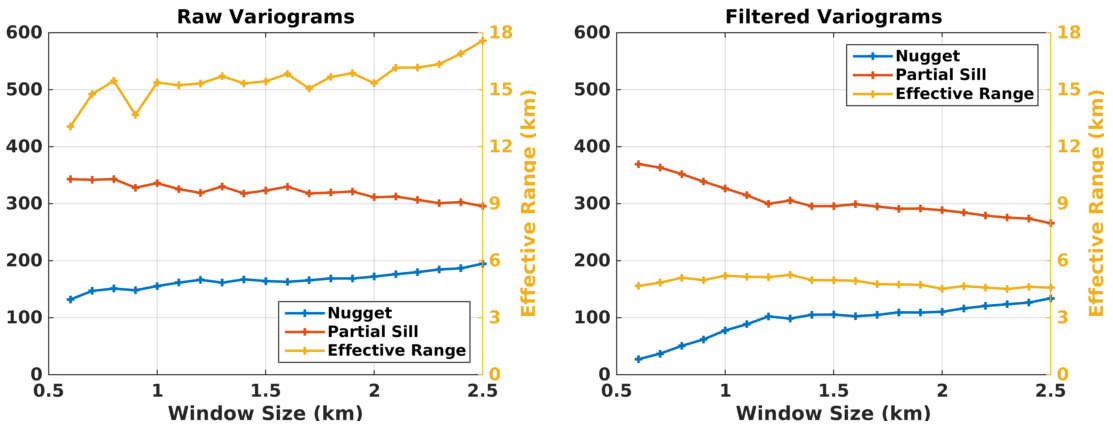

Figure 32: Parameters obtained with LS fit as a function of window size of the moving average.

Table 5: Interpolation parameters

Test cases	Raw	Filtered
Nugget	164	105
Partial Sill	323	296
Effective Range	15.4 km	5.0 km

variation (the nugget effect), we do not necessarily want the surface to pass through the data points. Hence, it would be unfeasible to compare the errors of the actual data points, but instead, we carry out the interpolation with two different data sets and compare the square error of the resulting surfaces instead.

The data sets are divided by using data from half of the sensors for other set and from the other half for the other set. Since there are four sensors, two of them are used for both sets in each location (cluster). In locations where part of the data is corrupted, an even number of sensors is used. That is, if only three valid measurement exists in a cluster, only one is used for both data sets to make the weighting equal for the interpolation. The interpolated surfaces are presented in Figure 33.

The surfaces on the left correspond to unfiltered variogram model and on the right to filtered. The square error that is calculated from each of the 10000 data points that form the surfaces is shown at the bottom of the figure. As could be expected, unfiltered case produces a smaller error. This is because the nugget and the range is clearly higher compared to the filtered case, which means that less weight is put on nearby data points and more on the further ones. Hence, we choose to use the unfiltered variogram model for the final interpolations with all the data.

However, this is not necessarily the absolute truth about which of the methods is better; smaller range and nugget gives individual measurements more weight and thus could reveal more potential spectrum opportunities, whereas larger values can

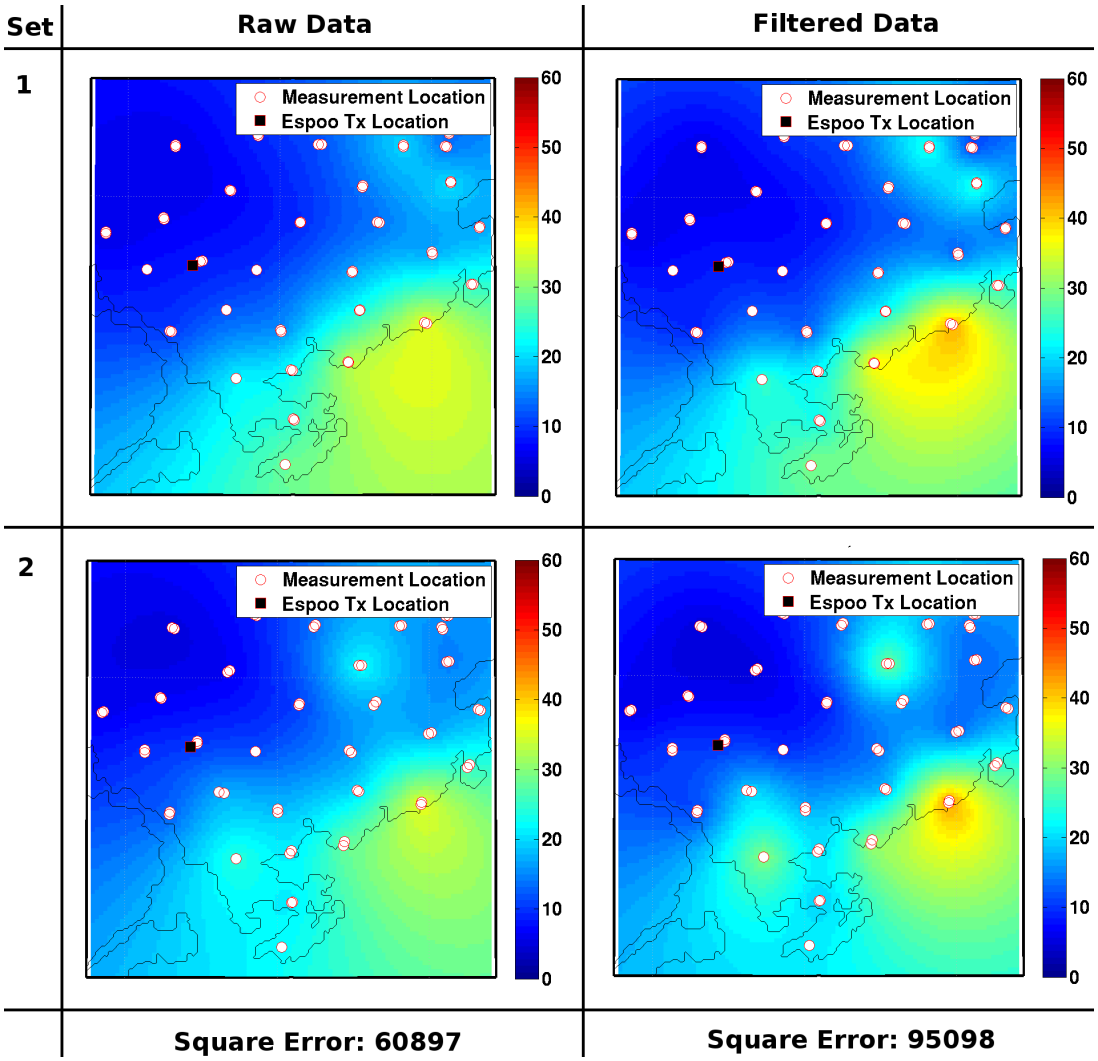

Figure 33: Crossvalidation surfaces with unfiltered and filtered variogram models.

mitigate channel conditions and measurement errors better.

5.3.2 Interpolation Results

Figure 34 shows the final interpolation results from channels 45 to 47. Here, the previously obtained parameters are used for the interpolation of cyclostationary test statistics, and the same method is used for modeling the variogram and calculating the estimates for the RSSI.

As mentioned in previous sections, channel 45 is transmitted across the sea from Tallinn and thus shows the highest variance and is affected by the surface the most. Channel 46 is transmitted from Espoo and is therefore very strongly visible with both detectors. Conversely, channel 47 should be completely unoccupied and thus shows low values in the cyclostationary test statistics.

The interpolation results for the cyclostationary test statistics seem to correspond

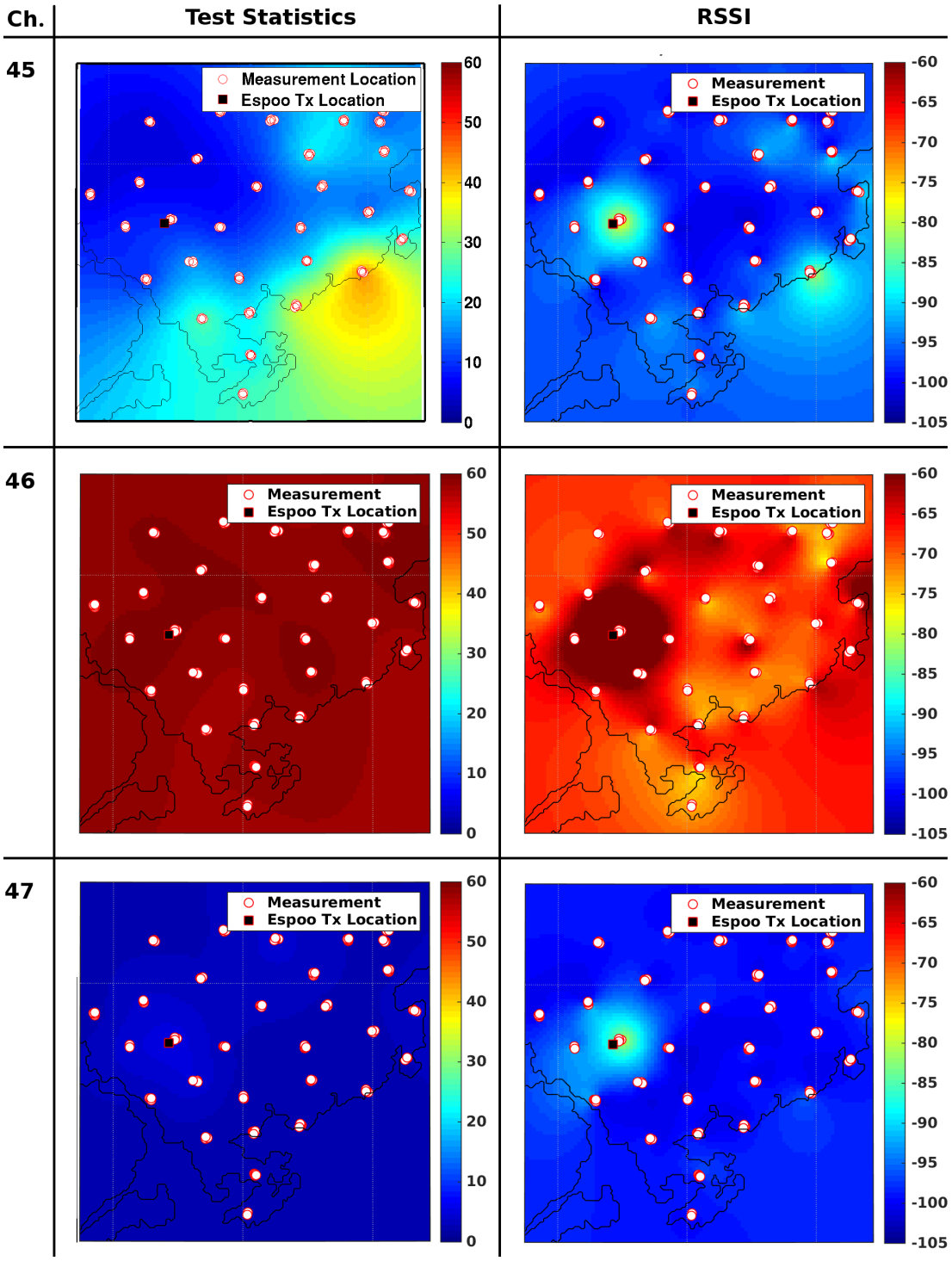

Figure 34: Spatial interpolation of cyclostationary test statistics and RSSI with Kriging method in Espoo region with the obtained parameters.

very closely to what was expected. Channel 46 is clearly detected and 47 not detected at all in the whole measurement area. Channel 45 shows higher test statistics near

the coast and even extrapolated outside of the measurement hull, which is caused by the transmitter location that is almost directly to the South in the figure.

What is also worth noticing, is that the channel cannot be clearly declared unoccupied anywhere in espoo, which is mostly caused by the range of the variogram. The effect can be understood for example as follows: If the range of the spatial correlation is more than 10 kilometers, and we clearly detect a signal 10 kilometers away with several detectors or in several locations, we cannot declare the channel locally unoccupied based on individual measurements. It could be assumed from the figure, that the channel would be usable further in the inner land (North), given that no channel 45 is only transmitted from the South.

By observing the prediction results of the energy estimates (RSSI), we can clearly see the benefit of using a feature detector: RSSI is highly distorted by the transmission on channel 46 in the vicinity of the transmitter on both of the neighboring channels, whereas the effect is negligible in the cyclostationary test statistics. The increased values around the transmitter are caused by the self-modulation of channel 46 due to non-linearity of the sensors, and would cause both of the channels 45 and 47 seem strongly occupied in that area. Furthermore, based on variogram and the results on channel 45, the RSSI values show poor spatial correlation, since the noise floor, intermodulation and random interference distort the results. This makes it also difficult to choose an optimal variogram model.

6 Conclusions

The goal of this master's thesis was to introduce the basic principles of the cognitive radio and spectrum sensing techniques in real-world conditions. Various methods for improving energy efficiency and performance in mobile sensing was introduced and mobile device related non-idealities were discussed. Detection algorithms and schemes along with methods for compensating the hardware issues were implemented on a spectrum sensor prototype, and their feasibility was tested with laboratory and field measurements.

In chapter 2, we focused on the operation principle of the cognitive radio and the theoretical side of spectrum sensing. An overview of various detection methods and standalone spectrum sensing algorithms along with their classifications was given, and their suitability to different use cases was discussed. Two detection algorithms, energy detector and spatial sign cyclic correlation (SSCC) detector, were introduced in detail in order to give theoretical background for the implementations presented in chapter 4. In addition, cooperative sensing and spatial interpolation were introduced.

Clearly, each of the detection algorithms is fit for a purpose and has their advantages and drawbacks. For example, energy and waveform based detectors can be extremely fast, computationally simple and hence also very energy efficient compared to other alternatives. The downside is that they are not robust to noise uncertainty and cannot differentiate signals of different systems, or even primary signal from the interference. This makes these algorithms unreliable in practical scenarios, where radio environment and system parameters are never ideal. However, the aforementioned downsides do not necessarily make these detectors unfeasible: the efficiency makes them a great option for two-stage detectors or simple applications, where top-grade accuracy is not crucial.

Feature detectors and matched-filter detectors, on the other hand, can be very robust and have excellent detection performance, even under low signal-to-noise ratio. However, most of these algorithms tend to have high energy consumption and/or delay, since they are computationally complex. Still, exceptions exist: the introduced spatial sign cyclic correlation detector can be implemented relatively simply using angular domain calculation and uniform distribution test. Although this method appears to be slightly less sensitive compared to its similar alternatives (given the same detection time), it was found out to be very energy efficient and thus a considerable choice for devices with constrained power supply capacity.

Given the non-ideality of real-world scenarios, even the most robust standalone detection schemes cannot provide reliable decisions in every situation. Thus, it is often crucial to use external data sources. Such sources can be either geographically neighboring sensors, or geolocation-based databases. The former scenario is called cooperative sensing, where occupancy information is combined from neighboring sensors, thus increasing the reliability of an individual sensor. In addition to reliability, cooperative sensing can even increase energy efficiency by censoring out non-informative data prior to combination. The geolocation databases are based on data that is stored from earlier measurements. Normally, it is geographically

mapped, i.e. each measurement result corresponds to a certain geographical area. In order to cover the areas where no data exists or cannot be even obtained, spatial interpolation can be a feasible option. The idea of the method is to generate data in between measurement locations using a mathematical model, such as Kriging or deterministic interpolation.

The weakness in both of the aforementioned methods is that they are applicable only when the data is available: when there are other sensors in the proximity or when a database connection can be established. Furthermore, the information in the database may not be up-to-date and does not take into account the fact that channel conditions and interference sources may vary over time. Thus, it could be said that practically none of the sensing methods alone is optimal for every situation, and the best overall performance can usually be achieved with a combination of several methods.

Chapter 3 focused on the aspects of mobile spectrum sensing. In this chapter, a practical "playground" for the rest of the work was given: which hardware sections need attention, what to take into account in the implementations, and what can be expected from the measurement results. The most significant non-idealities that possess mobile hardware were introduced, with emphasis on non-linearity and antenna-related issues. In addition, channel conditions that may affect field measurements were discussed and a brief overview on energy-efficiency issues was given. Finally, a two-stage (coarse-fine) spectrum sensing scheme, which intends to reduce the overall energy consumption and delay, was introduced.

Among all of the non-idealities of mobile-enabled hardware, non-linearity was found out to be among the most problematic issues. The increasing trend in the bandwidth of new communication systems make the issue even more relevant: the more signals fall on the receiver bandwidth, the more they will cause interference due to non-linearity. Since the price and form of the front-end chips are strictly constrained, the problem cannot be fully addressed by hardware design. Thus, digital post-processing methods as well as suitable algorithm choices are essential.

Regarding the form factor of hand-held devices, the shape and placement of the antenna may be a major challenge. There are several important factors in the design of miniature antennas: size versus sensitivity, shape versus radiation pattern, and both of them versus flexibility in terms of antenna placement in the size-constrained enclosures. In terms of antenna optimization, not only external signals need to be considered, but also the coupling of interference from electronics to the antenna. It was discussed, as well as measured, that the hardware may produce very problematic spurious emissions that lead to significant increase in the false alarm rate. This, in turn, causes loss of channel capacity if the antenna is not designed and placed carefully.

Furthermore, a wide variety of other non-ideal factors may affect the performance of spectrum sensing. Practically each part of the analog signal chain distorts the signal in a way or another and adds noise. Consequently, digital algorithms may produce unexpected results when the input signal has unexpected features. In addition to hardware-related degradation, even more so, the external interference and especially the unknown channel conditions that are present in real-worlds ra-

dio environment hampers the performance. What makes these conditions especially challenging, is that they cannot be completely characterized; location, strength and propagation route of the interferers varies depending on location and time. All of the aforementioned issues sum up and lead to false alarms and missed spectrum opportunities, as well as sensitivity degradation and possible interference with the primary system. Hence, the cognitive radios need to be designed to take advantage of external data, such as other sensors and databases, in addition to meeting the strict sensitivity requirements set by standards.

Another very important aspect of mobile spectrum sensing is the energy efficiency. It was concluded that two design parameters dominate the energy consumption: the choice of the detection algorithm and its run time. However, there are methods for optimizing the required run time as well as to reduce the overall need for local spectrum sensing. Considering local sensing, a coarse-fine sensing scheme is such a method. In the scheme, the operation of a computationally heavy fine detector is controlled by the decisions from a coarse detector. It was proven by measurements that depending on the chosen behavior of the coarse detector, the overall result can be pure improvement in the overall performance, that is, the run time and energy consumption. Hence, it always makes sense to use the scheme when the chip area is not limited.

Chapter 4 focused on the implementations of the introduced spectrum sensing algorithms, a coarse-fine controller, and improved controls for the RF front-end. Their functionality was verified and performance evaluated in chapter 5. In addition to the implementations, the most relevant non-idealities and environmental conditions were evaluated, including non-linearity, noise uncertainty, interference propagation from hardware to the RF input, stabilization delays and antenna-related issues. Finally, a practical use case of spatial interpolation using MATLAB and real measurement data was presented.

To account for the inconvenient stabilization time of front-end, a delay system and a new gain controller was implemented. The gain controller was measured to outperform the built-in controller of the front-end chip. It seems, that discrete gain steps can yield significantly better performance compared to continuous (IIR) gain adjustment loop when a consistent signal level is not needed, especially if the front-end is designed for static operation instead of rapid spectrum sensing.

Non-linearity was verified to be the most significant problem in the front-end. The effect of it was verified by a separate measurement by observing the false detections on adjacent channels. However, the effect was also visible in most of the other measurements. Another very problematic phenomenon is the spurious emissions caused by the electronics. Coupled with a poorly placed antenna and a detection algorithm that is distorted by the emissions, the false alarm rate increased close to 100% on certain channels, which is clearly unacceptable in any practical implementation. Although the on-board FPGA may be a significant contributor to the emissions, attention needs to be paid to the component and antenna choices and placement when designing robust spectrum sensing hardware.

Finally, a use-case of the spatial interpolation algorithms was presented. Kriging method was used to generate predictions on the areas that were not covered by the

measurements. The resulting figures showed clearly the expected behavior of the signal propagation as well as the differences between energy and feature detectors. The intermodulation that was showing in the results of the energy detectors in the vicinity of the transmitter was not visible in the results of feature detector. Furthermore, the spatial correlation of the cyclostationary test statistics was found out to be more clear and consistent, which improved the extrapolation capability, mitigation of individual errors in the measurement and in theory could mitigate the channel conditions, too. Thus, it could be said that the interpolation of cyclostationary test statistics can potentially give more reliable predictions from signal detection point of view, compared to interpolation of energy estimates.

The research outcomes can be enhanced by numerous ways in the future. For the coarse-fine scheme, it would be beneficial to evaluate the gains in varying terrain versus static measurements to verify the feasibility of the scheme in realistic mobile applications. Furthermore, the spatial interpolation methods should be compared with each other using the measurement data in different locations and study their feasibility in different use-cases. The time-aspect should also be researched more thoroughly: the calculation rate of the interpolation data needs to be optimized not to waste any resources or cause overhead to the spectrum sensing.

References

[1] A. Cárdenas-González, "Implementation of signal detector for cognitive radio," Master's thesis, Aalto University School of Electrical Engineering, 2013.

[2] I. Rognoni, "Geostatistics in cognitive radios," Master's thesis, Politechico di Milano, 2014.

[3] S. Kallionen, M. Vaarakangas, P. Hui, J. Ollikainen, I. Teikari, A. Parssinen, V. Turunen, M. Kosunen, and J. Ryynanen, "Multi-mode, multi-band spectrum sensor for cognitive radios embedded to a mobile phone," in *Cognitive radio oriented wireless network and communications*, pp. 236–240, 2011.

[4] V. Turunen, M. Kosunen, A. Huttunen, S. Kallioinen, P. Ikonen, A. Parssinen, and J. Ryynanen, "Implementation of cyclostationary feature detector for cognitive radios," in *Cognitive radio oriented wireless network and communications*, pp. 1–4, 2009.

[5] M. Vaarakangas, S. Kallioinen, A. Parssinen, V. Turunen, and J. Ryynanen, "Trade-offs in primary detection using a mobile phone as a sensing device," in *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2011 Sixth International ICST Conference on*, pp. 241–245, 2011.

[6] S. Chaudhari, M. Kosunen, S. Mäkinen, A. Cárdenas-González, V. Koivunen, J. Ryynänen, M. Laatta, and M. Valkama, "Measurement campaign for collaborative sensing using cyclostationary based mobile sensors," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 283 – 290, 2014.

[7] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mähönen, "Cognitive radio networking and communications: An overview," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 3386 – 3407, September 2011.

[8] D. Danev, E. Axell, and E. G. Larsson, "Spectrum sensing methods for detection of DVB-T signals in AWGN and fading channels," in *2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pp. 2721 – 2726, 2010.

[9] S. Haykin, "Cognitive radio: brain-empowered wireless communications," in *Selected Areas in Communications*, vol. 23:2, pp. 201–220, 2005.

[10] V. Valenta, R. Marsalek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on spectrum utilization in europe: Measurements, analyses and observations," in *2010 Proceedings of the Fifth International Conference on Cognitive Radio Oriented Wireless Network & Communications (CROWNCOM)*, pp. 1 – 5, 2010.

[11] F. C. Commission, "Report of the spectrum efficiency working group," tech. rep., Federal Communications Commission Spectrum Policy Task Force, November 2002.

[12] B. A. Fette, *Cognitive radio technology*. Burlington, MA, USA: Newnes, 2nd ed., 2009.

[13] A. M. Wyglinski, *Cognitive radio communications and networks: principles and practice*. Amsterdam: Academic Press, 2010.

[14] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom of wireless x networks," *IEEE Transactions on Information Theory*, vol. 55, pp. 3893 – 3908, September 2009.

[15] N. Devroye, P. Mitran, and V. Tarokh, "Achievable rates in cognitive radio channels," *IEEE Transactions on Information Theory*, vol. 52, pp. 1813 – 1827, May 2006.

[16] P. Grover and A. Sahai, "On the need for knowledge of the phase in exploiting known primary transmissions," in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007.*, pp. 462 – 471, 2007.

[17] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 4 – 17, February 2008.

[18] T. Yucek, "A survey of spectrum sensing algorithms for cognitive radio applications," in *Communications Surveys & Tutorials, IEEE*, pp. 116–130, 2009.

[19] E. Hossain, D. Niyato, and Z. Han, *Dynamic spectrum access and management in cognitive radio networks*. Cambridge, United Kingdom: Cambridge University Press, 2009.

[20] DARPA, "Next generation communications: Kickoff meeting." Proceedings of the Defense Advanced Research Projects Agency (DARPA'01), 2001.

[21] D. Arianandа, M. Lakshmanan, and H. Nikookar, "A survey on spectrum sensing techniques for cognitive radio," in *Second International Workshop on Cognitive Radio and Advanced Spectrum Management, 2009. CogART 2009.*, pp. 74 – 79, 2009.

[22] L. Xing, "Comparison of reliability, delay and complexity for standalone cognitive radio spectrum sensing schemes," *IET Communications*, vol. 7, no. 9, pp. 799 – 807, 2013.

[23] J. Candy, *Model-Based Signal Processing*, ch. Discrete Random Signals and Systems, pp. 21 – 134. Wiley-IEEE Press, 1 ed., 2006.

[24] J. Renard, "Spatial sign cyclic-feature detection," *IEEE Transactions on Signal Processing*, vol. 61, pp. 4521 – 4531, September 2013.

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory.* Prentice Hall Signal Processing Series, Prentice-Hall PTR, 1998.

[26] H. V. Poor, *An Introduction to Signal Detection and Estimation.* A Dowden & Culver book, Springer, 1994.

[27] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests on statistical hypotheses," *Philosophical Transactions of the Royal Society of London*, vol. 231, pp. 289 – 337, 1933.

[28] "IEEE standard for information technology–telecommunications and information exchange between systems wireless regional area networks (wran)–specific requirements - part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures foroperation in the tv bands amendment 1: Management and control plane interfaces and procedures and enhancement to the management information base (mib)."

[29] S. Shellhammer, "Numerical spectrum sensing requirements," June 2006.

[30] C. R. Stevenson, C. Cordeiro, E. Sofer, and G. Chouinard, "Functional requirements for the 802.22 wran standard," January 2006.

[31] Digita, "Kanavat ja taajuudet." Webpage, November 2012.

[32] F. Khan and K. Nakagawa, "Comparative study of spectrum sensing techniques in cognitive radio networks," in *2013 World Congress on Computer and Information Technology (WCCIT)*, 2013.

[33] Z. Tian and G. B. Giannikis, "A wavelet approach to wideband spectrum sensing for cognitive radios," in *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2006*, pp. 1 – 5, 2006.

[34] B. Zayen, A. M. Hayar, and D. Nussbaum, "Blind spectrum sensing for cognitive radio based on model selection," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008.*, pp. 1 – 4, 2008.

[35] Y. Zeng and Y.-C. Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," in *IEEE Transactions on Communications*, pp. 1784 – 1793, 2009.

[36] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, pp. 523 – 531, April 1967.

[37] H. Wang, E. hui Yang, Z. Zhao, and W. Zhang, "Spectrum sensing in cognitive radio using goodness of fit testing," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 5427 – 5430, November 2009.

[38] H.-S. Chen, W. Gao, and D. G. Daut, "Signature based spectrum sensing algorithms for IEEE 802.22 WRAN," in *IEEE International Conference on Communications, 2007. ICC '07.*, pp. 6487 – 6492, 2007.

[39] F. F. Digham, "On the energy detection of unknown signals over fading channels," in *IEEE International Conference on*, pp. 3575–3579, 2003.

[40] B. Lawton and C. C. Murphy, "Coarse-fine spectrum sensing for cognitive radio for minimum sensing time," in *2010 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1 – 7, 2010.

[41] A. V. Dandawaté, "Statistical tests for presence of cyclostationarity," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2355 – 2369, 1994.

[42] W. A. Gardner, "Spectal correlation of modulated signals: Part II–digital modulation," *IEEE Transactions on Communications*, vol. 35, pp. 595 – 601, June 1987.

[43] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86(4), pp. 639–697, April 2006.

[44] R. Prasad, *OFDM for Wireless Communications Systems*. Artech House, 2004.

[45] ETSI, "Digital video broadcasting (DVB). framing structure, channel coding and modulation for digital terrestrial television," 1997.

[46] M. Kosunen, "Survey and analysis of cyclostationary signal detector implementations on FPGA," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 4, pp. 541–551, 2013.

[47] V. Turunen, "Correlation-based detection of ofdm signals in the angular domain," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 951–958, March 2012.

[48] J. Lundén, S. A. Kassam, and V. Koivunen, "Robust nonparametric cyclic correlation-based spectrum sensing for cognitive radio," *IEEE Transactions on Signal Processing*, vol. 58, pp. 38 – 52, January 2010.

[49] M. G. Sanchez, L. de Haro, M. C. Ramon, A. Mansilla, C. M. Ortega, and D. Oliver, "Impulsive noise measurements and characterization in a uhf digital tv channel," *IEEE Transactions on Electromagnetic Compatibility*, vol. 41, pp. 124 – 136, May 1999.

[50] K. L. Blackard, T. S. Rappaport, and C. W. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 991 – 1001, September 1993.

[51] S. Visuri, V. Koivunen, and H. Oja, "Sign and rank covariance matrices," *Journal of Statistical Planning and Inference*, vol. 91, pp. 557 – 575, December 2000.

[52] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, pp. 1893 – 1907, September 2009.

[53] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communications*, vol. 4, pp. 40 – 62, March 2011.

[54] P. K. Varshney, *Distributed Detection and Data Fusion.* Springer, 12 ed., 1997.

[55] J. N. Tsitsiklis, "Decentralized detection," *Advances in Statistical Signal Processing*, vol. 2, pp. 297 – 344, 1993.

[56] S. Chaudhari, *Spectrum Sensing for Cognitive Radios: Algorithms, Performance, and Limitations.* PhD thesis, Aalto University School of Electrical Engineering, 2012.

[57] R. Viswanathan, "Cooperative spectrum sensing for primary user detection in cognitive radio," in *2011 Fifth International Conference on Sensing Technology*, 2011.

[58] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: A low-communication-rate scheme for distributed detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 554 – 568, April 1996.

[59] S. Maleki, A. Pandharipande, and G. Leus, "Energy-efficient distributed spectrum sensing with convex optimization," in *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009*, pp. 396 – 399, 2009.

[60] M. Umer, L. Kulik, and E. Tanin, "Spatial interpolation in wireless sensor networks: Localized algorithms for variogram modeling and kriging," *Geoinformatica*, vol. 14, pp. 101 – 134, 2010.

[61] N. Cressie, *Statistics for Spatial Data.* John Wiley and Sons, Inc, 1993.

[62] U.-A. J. E. Range, B.-A. Program, and I. C. of The Nature Conservancy, "The landscapetoolbox." Webpage, March 2012.

[63] D. Zimmerman, C. Pavlik, A. Ruggles, and M. Armstrong, "An experimental comparison of ordinary and universal kriging and inverse distance weighting," *Mathematical Geology*, vol. 31, pp. 375 – 390, 1999.

[64] P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty, Volume 497.* John Wiley and Sons, Inc, 2009.

[65] FCC News, "Cc crafts proposals to boost spectrum research and spectrum efficient wireless technologies." webpage, November 2010.

[66] K. Kim, J. Min, S. Hwang, S. Lee, K. Kim, and H. Kim, "A CR platform for applications in tv whitespace spectrum," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008.*, pp. 1 – 6, 2008.

[67] S. W. Oh, A. Naveen, Y. Zeng, V. Kumar, T. Le, K. Kua, and W. Zhang, "White-space sensing device for detecting vacant channels in tv bands," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008.*, 2008.

[68] S. K. Jones, T. W. Phillips, H. L. V. Tuyl, and R. D. Weller, "Evaluation of the performance of prototype tv-band white space devices phase ii," tech. rep., Technical Research Branch Laboratory Division Office of Engineering and Technology, Federal Communications Commission, 2008.

[69] A. A. Abidi, "Direct-conversion radio transceivers for digital communications," *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 1399 – 1410, December 1995.

[70] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference Record of the Thirty-Eighth Asilomar Conference on*, pp. 772–776, 2004.

[71] K. Kokkinen, "D3.2 - test report on sensing algorithms." Webpage, September 2010.

[72] M. Allen, J. Marttila, and M. Valkama, "Digital post-processing for reducing A/D converter nonlinear distortion in wideband radio receivers," in *Signals, systems and computers*, pp. 1111–1114, 2009.

[73] K. M. Ghairaibeh, *Nonlinear distortion in wireless systems: modeling and simulation with Matlab*. Wiley, 2011.

[74] D. H. Mahrof, E. A. M. Klumperink, and B. N. J. C. Haartsen, "On the effect of spectral location of interferers on linearity requirements for wideband cognitive radio receivers," in *IEEE DySPAN 2010 proceedings*, pp. 1–9, 2010.

[75] C. Rowell and E. Y. Lam, "Mobile-phone antenna design," *Antennas and Propagation Magazine, IEEE*, vol. 54, pp. 14 – 34, 2012.

[76] S. Chaudhari, "D3.2 - spectrum sensing algorithm evaluation." Webpage, May 2010.

[77] L. Li, X. Zhou, H. Xu, G. Y. Li, D. Wang, and A. Soong, "Energy-efficient transmission in cognitive radio networks," in *IEEE CCNC 2010 Proceedings*, 2010.

[78] S. Maleki, A. Pandharipande, and G. Leus, "Two-stage spectrum sensing for cognitive radios," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 2946 – 2949, 2010.

[79] Altera, *Cyclone III Device Datasheet.* Altera, July 2012.

[80] Analog Devices, *ADMTV102 Datasheet*, 2008.

[81] T. M. Hansen, "mgstat : A geostatistical matlab toolbox." Webpage, 2004 - 2008.

[82] J. P. Kleijnen and W. C. van Beers, "Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments," *European Journal of Operational Research*, vol. 165, pp. 826 – 834, 2005.

[83] W. MathWorld, "Nonlinear least squares fitting." Webpage, 1999 - 2014.

A Appendix A: C-Code for the Gain Controller

```
1  void dvbh_adjust_gains(void)
2  {
3      int rssi, gainmode = 0;
4      int timeout = 0;
5
6      // Set gains initially to maximum
7      dvbh_write_gainmode(gainmode);
8      IOWR(RSSI_OFFSET_TO_DBM_BASE, 0, (dS.rssi_offset_base));
9
10     // Request the RSSI estimation
11     IOWR(READ_RSSI_BASE, 0, 1);
12     usleep(20);
13     while (!IORD(RSSI_READY_BASE, 0) && timeout++ < 1000) {
14         usleep(10);
15     }
16     IOWR(READ_RSSI_BASE, 0, 0);
17     usleep(20);
18
19     // Read RSSI value
20     rssi = IORD(RSSI_IN_BASE, 0);
21     while ((rssi-256) > -87 && gainmode < 7) {
22         // Increase gain mode (decrease gains)
23         // until output is within acceptable limits
24         gainmode++;
25         dvbh_write_gainmode(gainmode);
26
27         IOWR(READ_RSSI_BASE, 0, 1);
28         usleep(20);
29             timeout = 0;
30         while (!IORD(RSSI_READY_BASE, 0) &&
31                 timeout++ < 1000) {
32             usleep(10);
33         }
34         IOWR(READ_RSSI_BASE, 0, 0);
35         usleep(20);
36
37         rssi = IORD(RSSI_IN_BASE, 0);
38     }
39
40     // Write offset to the estimator to get the real RSSI
41     IOWR(RSSI_OFFSET_TO_DBM_BASE, 0, (dS.rssi_offset_base +
42             gainmode*10 + (freq - 640) / 40));
43 }
```

B Appendix B: VHDL-Code for the Coarse-Fine Controller

```
1
2  LIBRARY IEEE;
3  USE IEEE.std_logic_1164.ALL;
4  USE IEEE.numeric_std.ALL;
5  USE work.basic_lib_comp.ALL;
6
7
8  ENTITY det_ctrl IS
9    PORT (
10     clk              : IN  std_logic;
11     resetn           : IN  std_logic;
12     ena              : IN  std_logic;
13     read_rssi        : IN  std_logic;
14     e_done           : IN  std_logic;
15     e_dec            : IN  std_logic;
16     a_done           : IN  std_logic;
17     a_dec            : IN  std_logic;
18     e_resetn         : OUT std_logic;
19     a_resetn         : OUT std_logic;
20     fifo_write       : OUT std_logic;
21     fifo_dec         : OUT std_logic;
22     rssi_in          : IN  std_logic_vector(7  DOWNTO 0);
23     tstat_in         : IN  std_logic_vector(15 DOWNTO 0);
24     reftime_in       : IN  std_logic_vector(31 DOWNTO 0);
25     thres_out        : OUT std_logic_vector(7  DOWNTO 0);
26     tstat_out        : OUT std_logic_vector(15 DOWNTO 0);
27     timestamp_out    : OUT std_logic_vector(31 DOWNTO 0);
28
29     -- Next two considered static during operation
30     force_int        : IN  std_logic_vector(4  DOWNTO 0);
31     rssi_mrg         : IN  std_logic_vector(7  DOWNTO 0)
32   );
33 END det_ctrl;
34
35
36 ARCHITECTURE rtl OF det_ctrl IS
37
38 ---------------------------------------------------
39 -- Internal signals
40 SIGNAL ena_reg                          : std_logic;
41 SIGNAL read_rssi_reg                    : std_logic;
```

```
42 SIGNAL read_rssi_d1, read_rssi_d2      : std_logic;
43 SIGNAL read_rssi_d3, read_rssi_d4      : std_logic;
44 SIGNAL e_done_reg, e_dec_reg           : std_logic;
45 SIGNAL e_done_reg1                     : std_logic;
46 SIGNAL a_done_reg, a_dec_reg           : std_logic;
47 SIGNAL e_resetn_reg, a_resetn_reg      : std_logic;
48 SIGNAL e_resetn_reg1                   : std_logic;
49 SIGNAL fifo_write_reg, fifo_dec_reg    : std_logic;
50 SIGNAL e_dec_overridable               : std_logic;
51 SIGNAL rs1_out, e_self_reset           : std_logic;
52 SIGNAL rssi_in_reg     : std_logic_vector(7 DOWNTO 0);
53 SIGNAL thres_out_r     : std_logic_vector(7 DOWNTO 0);
54 SIGNAL tstat_in_reg    : std_logic_vector(15 DOWNTO 0);
55 SIGNAL tstat_out_reg   : std_logic_vector(15 DOWNTO 0);
56 SIGNAL count           : std_logic_vector(4 DOWNTO 0);
57 SIGNAL reftime_reg     : std_logic_vector(31 DOWNTO 0);
58
59 BEGIN
60
61 ---------------------------------------------------
62 -- Input registers
63 inreg: PROCESS(clk) IS
64 BEGIN
65   IF clk'event AND clk = '1' THEN
66     e_done_reg1     <= e_done;
67     read_rssi_reg   <= read_rssi;
68     read_rssi_d1    <= read_rssi_reg;
69     read_rssi_d2    <= read_rssi_d1;
70     read_rssi_d3    <= read_rssi_d2;
71     read_rssi_d4    <= read_rssi_d3;
72     e_dec_reg       <= e_dec;
73     a_done_reg      <= a_done;
74     a_dec_reg       <= a_dec;
75     tstat_in_reg    <= tstat_in;
76     reftime_reg     <= reftime_in;
77   END IF;
78 END PROCESS inreg;
79
80 -- When only reading RSSI, disable
81 -- e_done_reg and override e_resetn_reg
82 e_done_reg      <= e_done_reg1 AND NOT read_rssi_d4;
83 e_resetn_reg1   <= e_resetn_reg OR read_rssi_reg;
84
85 ---------------------------------------------------
86 -- Adjustment processes
```

```
87
88 -- Capture RSSI when angular detection is done
89 rssi_reg_i : n_bit_register
90     GENERIC MAP (
91       n => 8)
92     PORT MAP (
93       d => rssi_in,
94       c => a_done,
95       q => rssi_in_reg);
96
97 -- Force angular detector at certain interval
98 force_counter: PROCESS(e_done_reg, resetn) IS
99 BEGIN
100   IF resetn = '0' THEN
101     count <= (OTHERS => '0');
102   ELSIF rising_edge(e_done_reg) THEN
103     IF unsigned(count) < unsigned(force_int) THEN
104       count <= std_logic_vector(unsigned(count) + 1);
105       e_dec_overridable <= e_dec_reg;
106     ELSE
107       count <= (others => '0');
108       e_dec_overridable <= '0';
109     END IF;
110   END IF;
111 END PROCESS force_counter;
112
113 -- RSSI threshold adjustment
114 thres_adj: PROCESS(a_done_reg, resetn) IS
115 BEGIN
116   IF resetn = '0' THEN
117     thres_out_r  <= std_logic_vector(to_signed(-70, 8));
118   ELSIF rising_edge(a_done_reg) AND a_dec_reg = '0' THEN
119     IF e_dec_reg = '1' THEN
120       thres_out_r <=
121         std_logic_vector(signed(thres_out_r) + 6);
122     ELSE
123       thres_out_r <=
124         std_logic_vector(signed(rssi_in_reg)
125         + signed(rssi_mrg));
126     END IF;
127   END IF;
128 END PROCESS thres_adj;
129
130 -- Teststat filtering
131 tstat_proc: PROCESS(e_dec_overridable) IS
```

```
132 BEGIN
133   IF e_dec_overridable = '0' THEN
134     tstat_out_reg <= tstat_in_reg;
135   ELSE
136     tstat_out_reg <= (OTHERS => '0');
137   END IF;
138 END PROCESS tstat_proc;
139
140 -- Reset signal handling
141 control: PROCESS(clk, resetn) IS
142 BEGIN
143   IF resetn = '0' THEN
144     ena_reg <= '0';
145   ELSIF clk'event AND clk = '1' THEN
146     ena_reg <= ena;
147   END IF;
148 END PROCESS control;
149
150 -- Reftime to timestamp
151 reftime: PROCESS(fifo_write_reg) IS
152 BEGIN
153   IF rising_edge(fifo_write_reg) THEN
154     timestamp_out <= reftime_reg;
155   END IF;
156 END PROCESS reftime;
157
158 ------------------------------------------------
159 -- Control logic
160
161 -- RS latch
162 rs_proc: PROCESS(ena_reg, e_done_reg, a_done_reg,
163     e_dec_overridable, rs1_out) IS
164 BEGIN
165   IF a_done_reg = '1' OR ena_reg = '0' THEN
166     rs1_out <= '0';
167   ELSIF e_dec_overridable = '0' AND e_done_reg = '1' THEN
168     rs1_out <= '1';
169   END IF;
170 END PROCESS rs_proc;
171
172 -- Energy decector self-reset
173 e_self_reset  <= e_dec_overridable AND e_done_reg;
174
175 -- Resetn to detectors
176 e_resetn_reg  <= ena_reg AND NOT (e_self_reset OR rs1_out);
```

```
177 a_resetn_reg  <= ena_reg AND rs1_out;
178
179 -- FIFO write signal & combined decision
180 fifo_write_reg  <= a_done_reg OR e_self_reset;
181 fifo_dec_reg    <= e_dec_reg OR a_dec_reg;
182
183 ------------------------------------------------
184 -- Output registers
185 outreg: PROCESS(clk) IS
186 BEGIN
187   IF clk'event AND clk = '1' THEN
188     fifo_write  <= fifo_write_reg;
189     fifo_dec    <= fifo_dec_reg;
190     e_resetn    <= e_resetn_reg1;
191     a_resetn    <= a_resetn_reg;
192     thres_out   <= thres_out_r;
193     tstat_out   <= tstat_out_reg;
194   END IF;
195 END PROCESS outreg;
196
197 END rtl;
```

C Appendix C: MATLAB-Code for Spatial Interpolation

```
clear all;
close all;
clc;

CH = 45; % DVB-T channel number, 45 = 666 MHz
ST = 3.6; % The sensing time, 3.6 or 7.2 or 14.4
pfa = 0.05; % False alarm rate for collaborative detection

%% Collect the data from the 4 sensors
LAT = [];
LON = [];
MZ = [];
cc = [];
sid = [];
loc_count = 0; % Counts the number of measurement locations
for caseIdx = [1:4]
    load(['ls_testCase' num2str(caseIdx+18) '.mat'])
    chind = find(gc.DVBChanFreqs.chNo==CH);
    lat = gv.tcOp.sensorResult(chind).latitude;
    lon = gv.tcOp.sensorResult(chind).longitude;
    senTime = gv.tcOp.sensorResult(chind).sensingTime;
    rssi = gv.tcOp.sensorResult(chind).rssi;
    testStat = gv.tcOp.sensorResult(chind).ls.testStats;

    LAT(end+1) = lat(1);
    LON(end+1) = lon(1);
    MZ(end+1)  = rssi(1);
    lid = length(MZ);
    cc(end+1) = 1;
    sid(end+1) = caseIdx;
    zz = [];
    zz(end+1) = rssi(1);
    loc_count = loc_count+1;
    tst(loc_count).zz = [];
    if senTime(1)==ST
        tst(loc_count).zz(end+1) = testStat(1);
    end
    tst(loc_count).rr = [];
    tst(loc_count).rr(end+1) = rssi(1);
    for i=2:length(lat)
        % If the next lat or lon value changes more than
```

```
42          % 0.0001 start a new location
43          if abs(lat(i)-lat(i-1)) > 0.0001 || ...
44                  abs(lon(i)-lon(i-1)) > 0.0001
45              loc_count=loc_count+1;
46              LAT(end+1) = lat(i);
47              LON(end+1) = lon(i);
48              MZ(end+1)  = rssi(i);
49              lid = length(MZ);
50              cc(end+1)  = 0;
51              sid(end+1) = caseIdx;
52              tst(loc_count).zz = [];
53              tst(loc_count).rr = [];
54          else
55              if senTime(i)==ST
56                  tst(loc_count).zz(end+1) = testStat(i);
57                  cc(end) = cc(end)+1;
58              end
59              tst(loc_count).rr(end+1) = rssi(i);
60          end
61      end
62  end
63
64  %% Mean test statistics
65  mts = [];
66  for i = 1:length(tst)
67      mts(end+1) = mean(tst(i).zz);
68  end
69
70  %% Cluster 4 closest sensors or not
71  clustering_on = 1;
72  I=1:length(LAT);
73  LATf  = [];
74  LONf  = [];
75  RSSIf = [];
76  TSTf  = [];
77  while not(isempty(I))
78      if clustering_on
79          % Lump 4 sensors to one location
80          % (31 measurement locations)
81          idx = find(abs(LAT(I(1))-LAT(I)) < 0.004 & ...
82              abs(LON((I(1)))-LON(I)) < 0.004);
83      else
84          % 4 sensors to different locations
85          % (117 measurement locations)
86          idx = find(abs(LAT(I(1))-LAT(I)) < 0.000004 & ...
```

```
87                  abs(LON((I(1)))-LON(I)) < 0.000004);
88          end
89          LATf(end+1)  = mean(LAT(I(idx)));
90          LONf(end+1)  = mean(LON(I(idx)));
91          RSSIf(end+1) = mean(MZ(I(idx)));
92          TSTf(end+1)  = mean(mts(I(idx)));
93          I(idx)=[];
94      end
95
96      %% Collaborative detection
97      coop_plot_on = 0;
98      LAT = LAT(cc>=400); % Location with at least 400 samples
99      LON = LON(cc>=400);
100     tst = tst(cc>=400);
101     MZ  = MZ(cc>=400);
102     for i=1:length(LAT)
103         Dl = [];
104         Tl = [];
105         d  = pos2dist(LAT(i), LON(i), LAT, LON, '1') * 1000;
106         % Only SUs within 200 m can cooperate
107         ind = find(d <= 200);
108         [s, ind] = sort(d, 'ascend');
109         nc = 1; % Number of collaborators (including oneself)
110         ind = ind(1:nc); % Fixed number of cooperating SUs
111         Nind(i) = length(ind);
112         % OR-rule, To have just local sensor set K=1 and nc=1
113         K  = 1; os = 0; leg = ['OR_' num2str(nc)]; color = 'b';
114         pf = linspace(0,1,10000);
115         p  = 1-binocdf(K-1,Nind(i),pf);
116         % Search for the correct local Pfa for K-out-of-N rule
117         [val, idx] = min(abs(p-pfa));
118         if ST==3.6 && i==1
119             trh = addt_thr(pf(idx), 2^15);
120         end
121         if ST==7.2  && i==1
122             trh = addt_thr(pf(idx), 2^16);
123         end
124         if ST==14.4 && i==1
125             trh = addt_thr(pf(idx), 2^17);
126         end
127         for k=1:length(ind)
128             % Cyclic test statistics
129             Dl(:,k)= tst (ind(k)).zz(1:400)>=trh;
130             Tl(:,k)= tst (ind(k)).zz(1:400);
131     end
```

```
132         Pdc(i) = mean(sum(Dl,2)>=K); % K-out-of-N
133     end
134     if coop_plot_on
135         figure; hold on;
136         plot(os+nc/10, mean(Pdc), 'o', 'color', color);
137         plot(os+nc/10, max(Pdc), '*', 'color', color);
138         plot(os+nc/10, min(Pdc), '*', 'color', color);
139         plot((os+nc/10)*[1 1], [min(Pdc) max(Pdc)], ...
140             'color', color, 'linewidth', 2);
141         legend(leg);
142     end
143
144     %% convert (LAT,LON) [deg] to (X,Y) [m] coordinates
145     a0 = min(LATf)-0.01;
146     o0 = min(LONf)-0.01;
147     [xm,ym] = Spherical2AzimuthalEquidistant(LATf, LONf, ...
148         a0, o0, 0, 0, 10);
149     xm = xm*10^6;
150     ym = ym*10^6;
151     lat_grid = linspace(min(LATf)-0.01, max(LATf)+0.01, 100);
152     lon_grid = linspace(min(LONf)-0.01, max(LONf)+0.01, 100);
153     [x, y] = Spherical2AzimuthalEquidistant( ...
154         lat_grid, lon_grid, a0, o0, 0, 0, 10);
155     x = x*10^6;
156     y = y*10^6;
157     X = meshgrid(x);
158     Y = meshgrid(y)';
159
160     %% Kriging for RSSI
161     options.polyfit = 1; % Fit a trend (Universal Kriging)
162     variogrammodel = '59 Exp(3000)';
163     [d_est, d_var, lambda_sk, K_dd, k_du, inhood] = ...
164         krig([xm' ym'], RSSIf', [X(:) Y(:)], ...
165         variogrammodel, options);
166     H = reshape(d_est, length(y), length(x));
167
168     %% Kriging for TS
169     options.polyfit = 1;
170     variogrammodel2 = '164 Nug(0) + 323 Exp(15400)';
171     [d_est2, d_var2, lambda_sk2, K_dd2, k_du2, inhood2] = ...
172         krig([xm' ym'], mts', [X(:) Y(:)], ...
173         variogrammodel2, options);
174     H2 = reshape(d_est2, length(y), length(x));
175
176     %% Plot the interpolated field
```

```
177 Hv = cat(3, H, H2);
178 fontsize = 16;
179 for i = 1:2
180     latlim = [min(LATf)-0.01 max(LATf)+0.01];
181     lonlim = [min(LONf)-0.01 max(LONf)+0.01];
182     figure, hold;
183     % Make map centered in ESPOO
184     map = worldmap(latlim, lonlim);
185     setm(map, 'FontSize', fontsize, 'FontWeight', 'Bold');
186     % Read coastline
187     land = shaperead('gshhs_coastline.shp', ...
188         'UseGeoCoords', true);
189     % Plot the Helsinki coastline
190     geoshow(map, land, 'facecolor', [1 1 1]);
191     geoshow(map, meshgrid(lat_grid')', ...
192         meshgrid(lon_grid), double(Hv(:,:,i)), ...
193         'DisplayType', 'texturemap');
194     mloc = geoshow(map, LATf, LONf, ...
195         'DisplayType', 'point', 'Marker', 'o', ...
196         'MarkerFaceColor', 'w', 'markersize', 9);
197     set(gca, 'FontSize', fontsize, 'FontWeight', 'Bold');
198     colorbar;
199     set(gca, 'FontSize', fontsize, 'FontWeight', 'Bold');
200     txloc = geoshow(map,gc.EspTxLoc.lat, gc.EspTxLoc.long, ...
201         'DisplayType', 'point', 'Marker', 's', ...
202         'MarkerFaceColor', 'k', 'markersize', 11);
203     legend([mloc, txloc], ...
204         'Measurement Location', 'Espoo Tx Location');
205     if (i == 1)
206         title(['RSSI, Channel ' num2str(CH)])
207         % Colormap scaling
208         caxis([-105 -55]);
209         str = 'rssi';
210     else
211         title(['Test Statistics, Channel ' num2str(CH)])
212         caxis([0 60])
213         str = 'ts';
214     end
215     set(gcf, 'Color', 'w');
216     print('-dpng', '-r150', [str, num2str(CH), '.png']);
217 end
```