



**Miguel Filipe Batista Duarte**

Licenciado em Ciências da Engenharia Electrotécnica e de  
Computadores

## **Spectrum Sensing through Software Defined Radio**

Dissertação apresentada para obtenção do Grau de Mestre em  
Engenharia Electrotécnica e de Computadores, pela Universidade Nova  
de Lisboa, Faculdade de Ciências e Tecnologia.

Orientador : Rodolfo Oliveira, Professor Auxiliar, FCT-UNL

Júri:

Presidente: Prof. Luís Bernardo

Arguente: Prof. Luís Oliveira

Vogal: Prof. Rodolfo Oliveira



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

Maio 2014



## **Software Defined Radio**

Copyright © Miguel Filipe Batista Duarte, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



# Acknowledgements

I would like to first and foremost thank my mentor and teacher, Rodolfo Oliveira, whose availability and helpful criticism helped me learn everything needed to complete this dissertation. I would like to thank João Porém, who has kept me company throughout all these years. Also, a big thank you to all my colleagues at FCT including but not limited to Pedro Cartaxo, Fábio Passos, Telmo Ferraria and Sérgio Pinto. To my parents, Isabel and José Duarte, who allowed me to choose this path for my life, I extend the biggest gratitude possible. I hope to be able to compensate this great sacrifice. To my brother Tiago, my second father, I can only thank for helping me in whatever I needed, whenever I needed. To my uncle Mário, I wish peace and I thank for being inspiration and an example. And last but not least, a big thank you to my loving girlfriend, Rita, who helped, distracted and overall made me a better person to this day.



# Sumário

Está em curso uma mudança no paradigma de controlo de transmissões de rádio. Tarefas que já foram exclusivas de certas classes de hardware são cada vez mais realizadas por sistemas de software. Uma mudança completa para o domínio do software é previsível, criando um verdadeiro conceito de Software Defined Radio. Ao mesmo tempo que esta mudança ocorre, o espectro radioelétrico encontra-se saturado e quase completamente licenciado. No entanto, o espectro raramente é utilizado na sua totalidade ao longo do tempo, permitindo o seu uso oportunístico enquanto os utilizadores licenciados não o utilizam. Esta é uma parte da noção de Rádio Cognitivo, um tipo de rádio que permitirá acesso oportunístico ao espectro. Estes dois novos paradigmas podem ser combinados para produzir um rádio flexível e fiável, ultrapassando dessa forma alguns problemas relacionados com a escassez de espectro disponível. Esta dissertação começa uma exploração nesta área combinando estes dois paradigmas através do uso de um Detetor de Energia implementado através de um dispositivo do tipo Universal Software Radio Peripheral e da suite GNURadio. O desempenho do sistema é avaliado calculando as Probabilidades de Detecção e Falso Alarme em cenários reais e comparando-os com os valores teóricos calculados. Um método para definir limites de decisão para sinais de banda estreita é também testado com base em trabalhos de Teoria da Informação, utilizando os algoritmos Akaike Information Criteria e Minimum Description Length. Os resultados foram obtidos através de uma transmissão real utilizando dois dispositivos USRP comunicando entre si, um agindo como utilizador licenciado e outro como utilizador secundário, oportunístico. Finalmente, realça-se o trabalho tecnológico desenvolvido na dissertação, o qual permite testar de uma forma rápida outros parâmetros através dos scripts desenvolvidos, apresentando-se como uma ferramenta de trabalho para futuros trabalhos de investigação.

**Palavras Chave:** Sistemas de Rádio Cognitivo, Detecção baseada em Energia, Rádio Definido por Software.





# Abstract

A change in paradigm when it comes to controlling radio transmissions is in course. Tasks usually executed in an exclusive class of hardware systems are increasingly controlled by software systems. A deep change to the software domain is foreseeable, creating a true Software Defined Radio. At the same time this change occurs, the radioelectric spectrum is almost completely licensed. However, the spectrum is rarely used to its full extent over time, enabling its opportunistic use while the licensed devices do not communicate. This is a part of the notion of Cognitive Radio, a new kind of radio capable of using the spectrum in an opportunistic way. These two new paradigms in radio access can be combined to produce a flexible and reliable radio, overcoming the issues with radioelectric spectrum scarcity. This dissertation starts an exploration in this area by combining these two paradigms through the use of an Energy Detector implemented in a Universal Software Radio Peripheral device and using the GNURadio suite. The performance of such a system is tested by calculating the Probabilities of Detection and False Alarm in real scenarios and comparing them to the expected theoretical values. A method for defining thresholds for narrowband signals is also tested based on works in Information Theory concepts, i.e., the Akaike Information Criteria and the Minimum Description Length. The results are tested for a real transmission using two USRP platforms communicating with each other, one acting as the licensed user and the other acting as the secondary, opportunistic user. Finally, we highlight the technological work developed in this dissertation, which may support future research works through the use of the developed scripts, allowing a faster method to test algorithms with different parameterization.

**Keywords:** Cognitive Radio Systems, Spectrum Sensing, Software-Defined Radio.



# Acronyms

**ADC** Analog to Digital Converter

**AIC** Akaike Information Criteria

**ASIC** Application-Specific Integrated Circuit

**CORBA** Common Object Request Broker Architecture

**CR** Cognitive Radio

**DAC** Digital to Analog Converter

**DDC** Digital Down-Converter

**DUC** Digital Up-Converter

**DSP** Digital Signal Processor

**FPGA** Field-Programmable Gate Array

**GPP** Global-Purpose Processor

**GSM** Global System for Mobile Communications

**IF** Intermediate Frequency

**LTE** Long-Term Evolution

**MDL** Minimum Description Length

**MISD** Multiple Instruction Single Data

**MIMD** Multiple Instruction Multiple Data

**OSSIE** Open-Source Software Communications Architecture Implementation-Embedded

**PC** Personal Computer

**POSIX** Portable Operation system Interface

**RF** Radio Frequency

**SCA** Software Communications Architecture

**SDR** Software Defined Radio

**SISD** Single Instruction Single Data

**SIMD** Single Instruction Multiple Data

**SNR** Signal to Noise Ratio

**UHD** Universal Software Radio Peripheral Hardware Driver

**UMTS** Universal Mobile Telecommunications System

**USRP** Universal Software Radio Peripheral

**VHDL** Very High Speed Integrated Circuits Description Language

# Contents

<b>Dedicatory</b>	<b>iii</b>
<b>Sumário</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Acronyms</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation . . . . .	1
1.3 Objectives and Contributions . . . . .	2
1.4 Structure of the dissertation . . . . .	3
<b>2 Software Defined Radio</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Hardware Architecture of an SDR . . . . .	6
2.2.1 General Hardware Architecture . . . . .	7
2.2.2 USRP Case Study . . . . .	9
2.2.3 Processing Options . . . . .	11
2.3 Software Architecture of an SDR . . . . .	13
2.3.1 Software Communications Architecture . . . . .	13
2.3.2 GNURadio . . . . .	16
2.3.3 A Comparison of SCA (OSSIE) and GNURadio . . . . .	16
2.4 Conclusions . . . . .	17
<b>3 Cognitive Radio</b>	<b>19</b>
3.1 Introduction . . . . .	19
3.2 Spectrum Analysis . . . . .	20
3.3 Conclusions . . . . .	28
<b>4 Experiments on EBS Parametrization</b>	<b>29</b>
4.1 Testbed Description . . . . .	29

4.2	Experiment Description . . . . .	30
4.2.1	Noise and Signal Data Computation . . . . .	30
4.2.2	Data gathering with parametrized thresholds . . . . .	31
4.2.3	Detector Characterization . . . . .	32
4.3	Experimental Results . . . . .	32
4.4	Noise Floor Estimation Experiment . . . . .	35
4.4.1	First Experiment - Gaussian Noise . . . . .	35
4.4.2	Second Experiment - Sine Wave . . . . .	38
4.4.3	Third Experiment - Three-Tone Wave . . . . .	38
4.5	Conclusions . . . . .	42
<b>5</b>	<b>Conclusions</b>	<b>43</b>
5.1	Final Considerations . . . . .	43
5.2	Future Work . . . . .	44
	<b>Bibliography</b>	<b>45</b>
	<b>Appendixes</b>	<b>49</b>
<b>A</b>	<b>DOCeis Article</b>	<b>51</b>

# List of Figures

2.1	SDR Basic Schema . . . . .	6
2.2	SDR Basic Schema . . . . .	7
2.3	Digital Down Converter . . . . .	8
2.4	Digital Up Converter . . . . .	8
2.5	USRP Connectivity . . . . .	10
2.6	USRP Base Structure . . . . .	10
2.7	SCA Architecture . . . . .	14
2.8	SCA Waveform Example . . . . .	14
2.9	GNURadio Architecture . . . . .	17
3.1	Cognitive Radio Schematic . . . . .	21
3.2	Cognitive Radio Sensed Spectrum . . . . .	22
3.3	Frame of the Secondary User . . . . .	23
3.4	Periodogram for a Sine Wave with Gaussian Noise . . . . .	26
3.5	AIC and MDL values for a Sine wave with Gaussian Noise . . . . .	27
3.6	Sorted periodogram of a Sine Wave with Gaussian Noise . . . . .	27
4.1	Secondary User GNURadio block diagram . . . . .	31
4.2	Primary User GNURadio block diagram . . . . .	31
4.3	Secondary User GNURadio Threshold block diagram . . . . .	31
4.4	Histogram of First USRP Experiment . . . . .	32
4.5	Probability of Detection and Probability of False Alarm for the first USRP Experiment . . . . .	33
4.6	$P_{Miss}$ for a range of SNR . . . . .	34
4.7	$P_{Miss}$ detail for a range of SNR . . . . .	34
4.8	$P_{FA}$ for a range of SNR . . . . .	35
4.9	Periodogram for a Gaussian Noise transmission . . . . .	36
4.10	AIC and MDL values for a Gaussian Noise transmission . . . . .	37
4.11	Sorted periodogram for a Gaussian Noise transmission . . . . .	37
4.12	Periodogram for a Sine wave transmission . . . . .	38
4.13	AIC and MDL values for a Sine wave transmission . . . . .	39
4.14	Sorted periodogram for a Sine wave transmission . . . . .	39

4.15	Periodogram for a Three-Tone wave transmission . . . . .	40
4.16	AIC and MDL values for a Three-Tone wave transmission . . . . .	41
4.17	Sorted periodogram for a Three-Tone wave transmission . . . . .	41



# List of Tables

3.1	MDL and AIC Noise Estimation Values . . . . .	27
4.1	USRP Parameters for Energy Detection Experiment . . . . .	32



# Chapter 1

## Introduction

### 1.1 Introduction

Over the last 40 years, technology has validated Moore's law [Moo06]. The number of components on a chip has consistently doubled every 18 months. This breakthrough in technology allows a processor to have capacities thought impossible just a few years ago, which opens the possibility to assign functions to processors that were only assigned to application specific hardware up until now [SRLM12]. This dissertation explores the use of processors and more specifically, software architectures to solve traditional problems in the radio domain. This technology, called Software Defined Radio, is the focus of this dissertation, which is adopted to assess spectrum sensing techniques.

### 1.2 Motivation

The Software Defined Radio, introduced by Mitola [Mit95], opened up a whole new world of possibilities regarding the use of radio equipments. In a world where the main agents were specifically built on circuits, the SDR contested this dogma. The main idea is to delegate some of the hardware specific tasks to software, namely baseband frequency operations, such as the generation and detection of signals [Wir].

SDR's main objective is to add flexibility to a radio communication device, providing it with the necessary autonomy to modify radio transmission parameters through the simple loading of a script.

This flexibility allows a device to better adapt to different kinds of communication. This is something of the utmost importance in military radios [Ulv10] for example, where the need to change communication patterns in case the transmission integrity or privacy is compromised by the "enemy" is a reality.

Another advantage is the SDR's time-to-market. At the manufacturer level, when a new wireless architecture (e.g. transmitter/receiver scheme) is required, less hardware and more software development is needed to implement new features, greatly decreasing the development time. Thus, time-to-market is greatly reduced while innovation grows [PRL<sup>+</sup>12].

The Cognitive Radio is an evolution in the paradigm of spectrum usage. It relies on the fact that a Secondary, unlicensed user can identify if the spectrum is being accessed by a Primary, licensed user. The reliability of this method must be high, since the interference caused to the licensed users must be avoided. Knowing if the spectrum is being used or not allows the Secondary User to control its transmission power and frequency occupation, therefore using a licensed frequency and increasing spectrum usability.

The main objective of this work is to evaluate the actual performance of SDR technology to perform Spectrum Sensing.

### 1.3 Objectives and Contributions

The first objective of this dissertation is validating through practical SDR experiments the Probabilities of Detection and False Alarm that characterize the ability of an Energy Detector to successfully identify the presence/absence of a Primary User in the sensed spectrum. For this end, a script that tests these probabilities was designed using the GNURadio software suite. This test allows the characterization of the Energy Detector within a range of different SNR values.

The second objective of the dissertation is the evaluation of two information criteria models as a method of detecting a narrowband transmission in the spectrum through a power difference in a periodogram. These models are the Akaike Information Criteria (AIC) and the Minimum Description Length (MDL). While these algorithms are generally used with eigenvalues as an input, in this case the periodogram values were considered as

eigenvalues, avoiding the need of a square matrix to compute eigenvalues and increasing the flexibility of the scheme, allowing for an arbitrary number of FFT samples to be collected.

The practical validation of the theoretical values of  $P_D$  and  $P_{FA}$  for a predefined sensing time have already contributed to the article presented in DOCeis 2014[MD14], titled *Practical Assessment of Energy-based Sensing through Software Defined Radio Devices*.

## 1.4 Structure of the dissertation

This work is divided into five chapters. The first chapter contains the global introduction, motivation and objectives for the dissertation. The second chapter describes first theme of the dissertation, the Software Defined Radio and its state of the art regarding technologies and models. The third chapter details the use of the Cognitive Radio for spectrum access and its mathematical models. The fourth chapter contains the practical experiments done in the course of the dissertation and the fifth chapter contains the global and final conclusions regarding the work, as well as the future work suggested.



## Chapter 2

# Software Defined Radio

### 2.1 Introduction

Reconfigurability is the main characteristic at the core of the SDR concept. Over the years, the evolution of telecommunications has brought technologies such as GSM (Global System for Mobile Communication), UMTS (Universal Mobile Telecommunications System) or more recently LTE (Long Term Evolution) to life.

Although the communication schemes supporting each technology are different, their objective remains the same: transmit and receive voice and data. SDR proposes to unite some of these technologies in a single device, switching between them as needed. This proposition is achieved by adding flexibility to the radio equipment. Reconfigurability, however, is practically non-existent nowadays as radio tasks are handled by application-specific hardware devices. The SDR concept, however, allows for radios to be configurable, even programmable in software to a large extent, with the help of software-controlled technologies. The need for this concept stems from the different transmission quality and QoS needs nowadays when performing a vertical handover from one radio standard to another (e.g. Switching from GSM to LTE on a phone).

A system that satisfies these flexibility needs via software controlled components is an SDR.

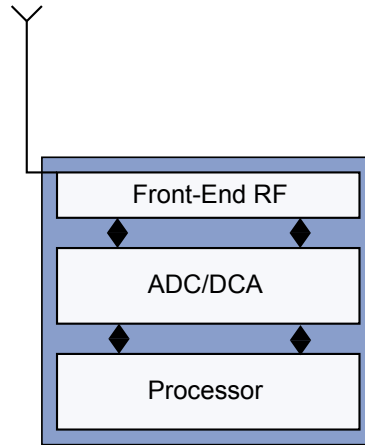


Figure 2.1: The basic schema of a SDR architecture

## 2.2 Hardware Architecture of an SDR

A basic Schema of SDR is presented in Figure 2.1. In the figure the Processor is represented as a single block. However, it can comprise several processors working in parallel. Theoretically there could be only one processor computing all the code. In practice the work of processing a signal is too demanding most of the time for a single processor, so the processor block can include an array of components that are built to manage waveforms and the processing needs associated with them.

Some of the hardware solutions featuring mostly multi-core processors are presented in [PRL<sup>+</sup>12]. There is still no ideal solution for the programmable processing associated with SDR. The technology has not allowed the SDR to reach the needed flexibility [Mit00], which joins a wide band of workable frequencies with the use of a GPP (Global Purpose Processor). This optimal state would require high processing speed, most probably in parallel, coupled with flawless component design [PRL<sup>+</sup>12]. This is not feasible nowadays.

Another issue with SDR is the power consumption [PRL<sup>+</sup>12]. Since GPPs are not built specifically for DSP (Digital Signal Processing), the power consumption is higher than in ASICs (Application Specific Integrated Circuits), which limits the options when designing portable radios that implement SDR technologies.



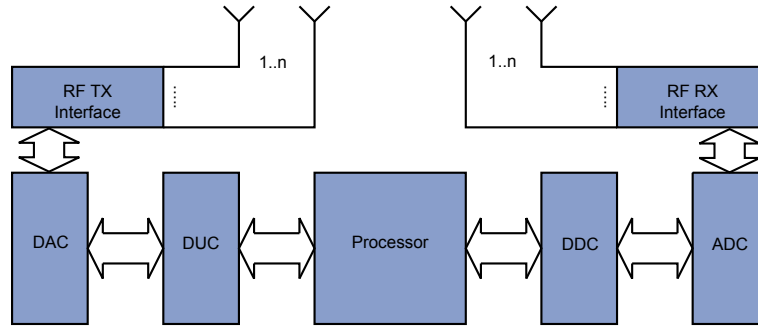


Figure 2.2: Basic Schematic of an SDR Architecture

### 2.2.1 General Hardware Architecture

The hardware architecture of a SDR system may vary, depending on its flexibility. There might be a need to change only one or all of the operational aspects of the radio. Thus, the hardware technology may change as well. A general design of an SDR is presented in Figure 2.2.

As can be seen in Figure 2.2, there may be more than one antenna attached to the RF front-end. This is due to the fact that by switching transmission protocols and supporting MIMO transmission, SDR systems can work in different frequencies covered by different antennas. Thus, SDR systems must have capabilities of switching antennas when they need to switch the operational frequency.

There are three distinct blocks for Downlink and Uplink.

#### Downlink Structure

The Downlink is responsible for receiving and decoding data. The first block in this chain is the RF front-end. This block is ideally reconfigurable for a wide range of frequencies and holds the interface between the Intermediate Frequency (IF) and Radio Frequency (RF) parts of the system. It generally consists of a Band-Pass Filter to eliminate undesired frequencies, a Low-Noise Amplifier to amplify the desired signal and a Mixer, responsible for converting the signal from RF to IF by mixing it with the Local Oscillator signal.

The next block is the ADC (Analog Digital Converter). This converter is responsible for sampling the analog signal after its processing by the RF block. It converts the analog signal into a digital representation, through the parametrization of the sample rate. Note

that for most of the signals, this conversion is only technologically possible because the central frequency of the original RF signal was decreased to the feasible (IF) levels in the RF front-end.

After the ADC the digital signal enters the Digital Down Converter (DDC), illustrated in Figure 2.3.

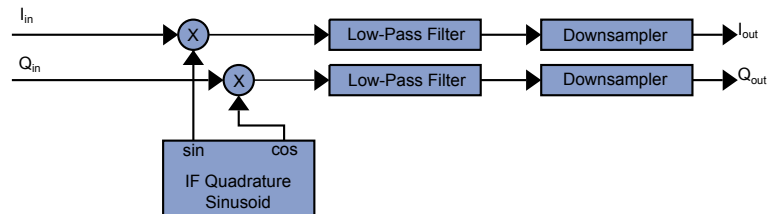


Figure 2.3: Digital Down Converter generic schematic

The DDC is responsible for bringing the signal down to baseband in order to be processed by the software. It is composed by an IF Sinusoid Multiplier, a Low-Pass Filter and a Downsampler. The Multiplier brings the signal down to baseband where digital signal processing is more feasible. The Low-Pass filter eliminates undesired clones created by the multiplication. Finally, the Downsampler throttles data so that it will achieve the necessary rate to pass through the BUS connecting it to the processor.

After this conversion the signal is brought to the processor, where it will be processed by software.

## Uplink Structure

The Uplink is responsible for generating and transmitting data. This data, as opposed to Downlink, is generated in the processor and transmitted through the BUS. It is then sent to the DUC (Digital Up Converter) (Figure 2.4).

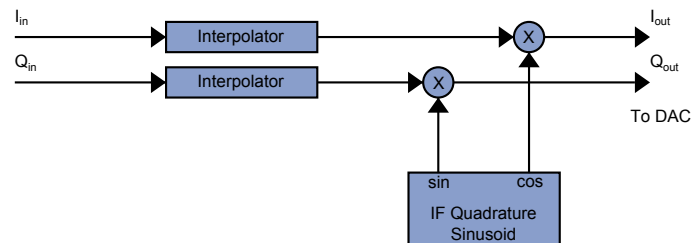


Figure 2.4: Digital Up Converter generic schematic

The DUC is responsible for increasing sample rate and converting the baseband signal

up to IF frequencies. It consists of an Interpolator to increase the sample rate and a Multiplier to convert the signal. At this stage no filter is needed because the signal will be filtered in the RF block.

The signal is then passed to the DAC (Digital to Analog Converter), which does the opposite of the ADC. It transforms the parameterized signal into an analog signal through interpolation, readying it for up-conversion to RF and transmission by the RF block.

The RF block then takes this IF analog signal, converts it to RF, filters it to match only the desired frequencies and transmits it. This block is responsible for power control of the transmission.

### **2.2.2 USRP Case Study**

The Universal Software Radio Peripheral (USRP) is a front end device designed for SDR purposes. USRPs consist of a motherboard and several possible daughterboards [Ett]. The motherboard contains an FPGA for signal processing, Digital to Analog Converters, Analog to Digital Converters, an interface to access the GPP linked to it and a power control device. The daughterboards contain the RF part of the device, converting the baseband signal to the desired frequency and transmitting it over an antenna.

The USRP is controlled by the USRP Hardware Driver, UHD. The UHD contains firmware to work with the built-in FPGA and the signal converters. The USRP uses the strategy described in Subsection 2.2.3, delegating all redundant and high bandwidth operations to the FPGA [RYZ11] to reduce sample rate sent to the peripheral processing unit. This unit could be a GPP, as mentioned before, and is connected to the USRP via an interface (e.g., USB or Gigabit Ethernet). The UHD ensures that the data is sent through the USB and returned after processed. Figure 2.5 shows the general architecture of the system.

There are, of course, several limitations in terms of the technology adopted in USRP devices. One of them is the main connection method to the processor, which adopts the USB 2.0 standard. It has been calculated that its data rate is of approximately 32MS/s (MegaSamples/second) and its bandwidth can not exceed 12MHz [RYZ11].

This means that higher bandwidth systems, like all 802.11 protocols, can not be

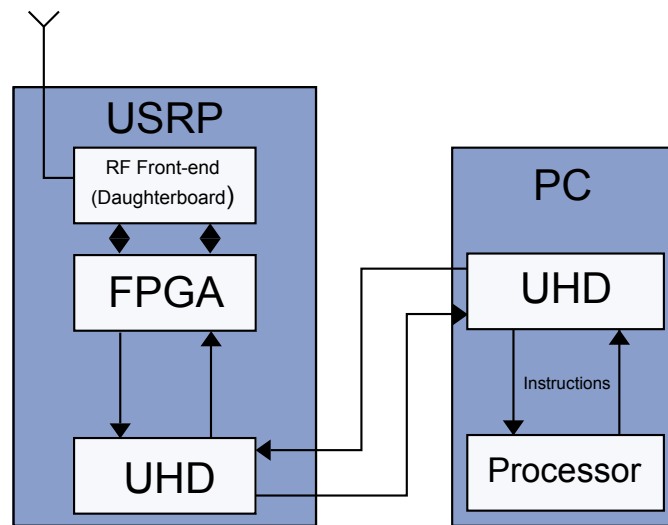


Figure 2.5: USRP connectivity to a PC and processing options

implemented using the USRP. This is one of its greatest shortcomings.

### USRP Structure

The structure of a USRP device is kept as simple as possible. Nevertheless, some explanation might be required to explain how its components work and connect to each other.

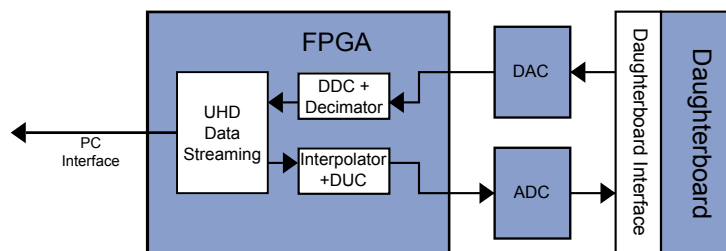


Figure 2.6: USRP basic structure. Changes apply between different models.

As can be seen in Figure 2.6, the intensive tasks of the DDC and DUC described in Section 2.2.1 are left to the FPGA. The FPGA comes with pre-defined scripts for handling this kind of operations. On each utilization, these scripts are loaded in the device and their operating parameters are properly defined (decimation and interpolation rate). It

is up to the user to change these settings according to the objectives. The figure also shows the ADC and DAC components, responsible for the Analog/Digital conversions. The performance of all these components varies from USRP model to model.

The UHD Data Stream controls the information from/to the PC and the PC Interface exchanges data between the devices through it. This interface can be a USB for some USRP models or a Gigabit Ethernet connection for others.

The daughterboards implement the RF front-end of the USRP. Several types of daughterboards are available, each one with their specific frequency range. The portfolio of different daughterboards allows the use of the USRP in most of the commonly used frequency ranges, all in the same device (same motherboard with different daughterboards).

### 2.2.3 Processing Options

The paradigm change associated with SDR brings challenges not faced before. One of them is finding the optimal hardware design for signal processing. There are many different options to handle the signal processing, and the main trade-offs involved in their choice are the level of performance and flexibility. In this section we will explore several systems already proposed for signal processing.

#### Application Specific Integrated Circuits

Application Specific Integrated Circuits (ASICs) are the *de facto* devices adopted in Digital Signal Processing. ASICs are circuits designed and built for one purpose specifically and can not be reprogrammed, meaning they are not a good choice for more flexible solutions. ASICs are, therefore, not compliant with the notion of SDR. However, some authors [SIW06] suggest that similar operations between waveforms should be studied and applied in ASICs whenever possible, in detriment of waveform global portability.

#### Reconfigurable Processing Elements

Field Programmable Gate Arrays (FPGA) are powerful programmable hardware systems. These kind of systems allow the user to reconfigure its hardware in a matter of milliseconds, being almost instantaneous in reconfiguring itself in several scenarios, such

as the need of processing a different waveform. These are widely used pieces of machinery for digital systems, with the downside of consuming more energy and occupying more space than ASICs.

The processing capabilities of FPGAs are similar to ASICs', but FPGAs are on average 3.2 times slower than ASICs [KR07]. This is much of the tradeoff between a fixed and programmable processing unit and one of the technical challenges with SDR, the processing speed.

An advantage of using FPGAs is their massive use nowadays. There are many useful toolsets already available in the market to work with SDR and FPGAs. One of these tools translates c-code into FPGA code, i.e. Very High Speed Integrated Circuits Hardware Description Language (VHDL)[LBJ06]. This implies the possibility of side-processing done by FPGAs alongside General Purpose Processors (Subsection 2.2.3). It could be especially useful for mathematical intensive computations.

### **Global Purpose Processor**

These are everyday processors one might find in almost every piece of electronic equipment nowadays. The implementation of the functions associated with waveforms is done at a much higher level than the hardware-based solutions. This implies higher latency but much higher flexibility.

The recent evolution of the GPP has migrated from increasing its clock speed to increasing the parallelism between components and increasing the number of processors in a single socket. This means a more suited type of processing for digital signals, due to the need of multiple and redundant operations when dealing with radio.

Currently there are four types of data processing associated with GPPs: Single Instruction Single Data (SISD), Multiple Instruction Single Data (MISD), Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD).

SISD processors were the first to appear: instructions are taken from the stack and only those are executed in only one set of data per cycle. This is however a low efficiency way of processing, as it relies only on higher clock rates to improve the computation speed.

Single Instruction Multiple Data processors handle instructions according to a de-

fined order, but only once at a time independently of how many processors are available. This opens the possibility to distribute common processes through different processors, imposing parallelism on the system when needed. An example of the application of such a feature would be sensing (verifying data being transmitted) through two different antennas, for two different frequency ranges. This would imply Fourier Transforms computed in parallel with two data sets, one per antenna. It would therefore benefit from the SIMD advantages.

MIMD processors distribute data and instructions through different processors as long as they are in line to be executed. This allows more flexibility in executing parallel tasks, allowing for a greater variation in parallel operations. Most of the processors available today are MIMD.

Despite its advantages, GPPs exhibit high energy consumption. These processors were not built for portable devices, at least not as small as a cell phone. Therefore, a battery would not last enough time to make GPPs a good choice to use in portable devices.

On the other hand, GPPs are most appropriate for academic and amateur applications. Most people own a computer, and that is a fact that plays strongly in GPPs' favor. Also, a GPP can be used together with an FPGA, for example, to compensate for any shortcomings it might have.

## 2.3 Software Architecture of an SDR

The Software Architecture of an SDR system specifies the way the software is organized to handle the data received from the antenna and the way it generates the data for transmission. It is inevitably a complex system, taking into account that all operational circumstances must be tackled. Three examples will be given in this section.

### 2.3.1 Software Communications Architecture

The Software Communications Architecture (SCA) [SCA] is a framework created by the Joint Program Executive Office (JPEO), an organization established in the United States, having the design of new SDR radio systems for the US Armed Forces in mind. The advantages brought by the use of this kind of software framework come from the

features listed in Section 1.2, namely the ability and speed to adapt to different radio parameters. The need to change the pattern of communication when it is compromised is a reality in the theater of war. Thus, JPEO created this framework which consists of a set of rules to obey when constructing a robust radio system compatible with SDR. SCA is not the only framework available, as seen later in the section, but it is a reference architecture that must be studied due to its importance in the current state of the art.

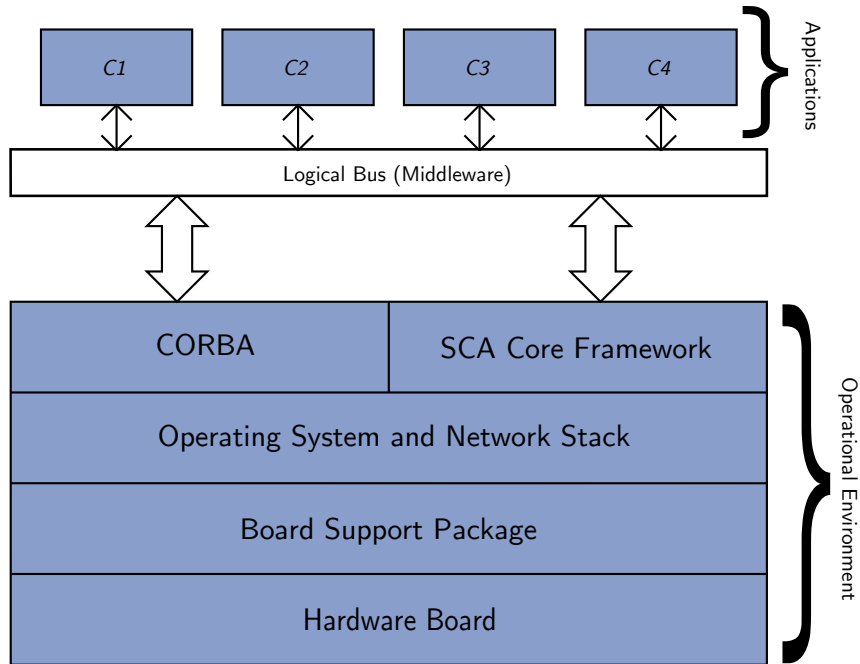


Figure 2.7: SCA Architecture, with the correspondent Core Framework.[Adapted from [SCA]]

Figure 2.7 presents the general structure of the framework. Each application represented in the figure (i.e.  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$ ) is a waveform. Waveform is the generic term for a complex radio structure. An example of a waveform is represented in Figure 2.8.

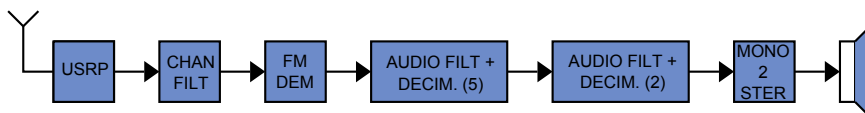


Figure 2.8: SCA Waveform Example, with high granularity

Software Communications Architecture defines how these waveforms are handled.

The hardware itself is generic. This is one of the objectives of SCA, hardware abstrac-



tion and independence by using common instructions in the hardware, supported by the Board Support Package.

The Operating System must be a Real Time Operating System, compatible with POSIX [Pos]. The Core Framework (CF) components have complete access to the OS, in order to call the services needed to execute the application.

The CF and the Common Object Request Broker Architecture (CORBA) standard are probably the most important parts of SCA. CORBA is used due to its flexibility, allowing the CF to distribute applications with platform independence and openness. This is achieved by using an Object Request Broker, or ORB, to manage applications and resources.

While ORBs are mainly supported by GPPs, special ORBs for working with SCA that provide a faster access to resources have been designed. These ORBs use DSPs and FPGAs in order to raise the system's portability [Hum06]. However, since all resources rely on CORBA to be handled, there is a significant overhead associated with each application. This is due to CORBA's characteristics such as message marshalling and location variation for resources. So, on one hand there is the mobility and flexibility earned by using CORBA. On the other, CORBA may degrade the computational time required to execute a given task. This issue can be mitigated by switching transport protocols with TCP/IP, CORBA's default protocol, which minimizes overhead [Doh02][BCMR02]. Also, CORBA has known issues with computation and memory resources, diminishing its popularity in some domains [Ulv10]. This fact raises the question: should SCA continue to exclusively use CORBA?

The CF is defined by a set of interfaces that dictate how waveforms and their components are put together. Its description is lengthy and goes beyond the scope of this thesis. However, all the interfaces are defined by CORBA's Interface Description Language. Therefore, working with SCA is much like working with CORBA, only with specific classes and not the whole array.

## OSSIE

Open-Source SCA Implementation-Embedded (OSSIE) is a project started by Wireless @ Virginia Tech, with the intent to distribute and facilitate access to the SCA framework. Most of the interfaces present in SCA are part of OSSIE [SCA]. However, some of them are already configured, which means that OSSIE is not as flexible as SCA is due to its complexity

OSSIE is written in C++ and uses omniORB and Xerces XML for parsing. It is developed for Linux. There is also the OSSIE Waveform Workshop, a set of tools designed to ease the work of a radio engineer.

### 2.3.2 GNURadio

GNURadio (GR) is an open source project that consists of a software-only approach for radio applications design. It can be connected to hardware for real world use or simulated in a conditioned environment.

GNURadio implements simple modules described by C++. Digital Signal Processing (DSP) is almost all done in software, hence there is a need to separate and categorize these modules.

GNURadio provides signal generators (*sources*), receivers (*sinks*) and intermediate blocks (*filters*). These blocks are linked through Python, defining the final schema to be tested. Python does nothing more than connect these blocks, which has generated some discussion regarding efficiency and the need for this extra layer of Python [TT09].

The general structure of the GNURadio architecture is represented in Figure 2.9. As depicted in the figure, graphical blocks and connections are built in Python, as opposed to all other DSP blocks which are built in C++.

Since GR is an open source project any user, amateur or professional, can contribute to extend the existing libraries, even to change the way GR works from its root.

### 2.3.3 A Comparison of SCA (OSSIE) and GNURadio

There are several available papers about using each of these systems to build a practical radio application. Due to its professional nature and roots, OSSIE has a higher complexity

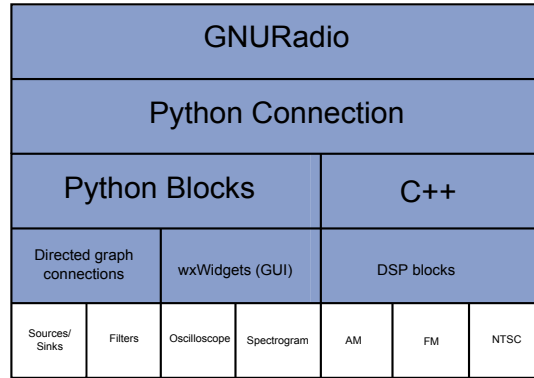


Figure 2.9: GNURadio blocks and applications, divided by layer.

and, therefore, difficulty. Its greatest advantages come from seamless distributed processing thanks to CORBA [BCMR02] and its numerous layers of security [MBF05]. However, some of its drawbacks are also due to CORBA.

In a comparative study between OSSIE and GR [ALRD<sup>+</sup>08] both systems are evaluated according to different features that are independent of the hardware schemes that can be adopted. The main conclusion highlighted in the study is that OSSIE requires more processing power and memory usage, exhibiting 5 times more CPU load and 2 times more memory for the same waveform.

It is also important mentioning that in a GNURadio environment the latency is 25 times shorter than that present in OSSIE. This larger latency in OSSIE is mostly due to the overhead introduced by CORBA. Therefore, for low budget (low hardware capacity) and non-commercial tests on waveforms, GNURadio seems like the most appropriate setup.

## 2.4 Conclusions

The use of the SDR proves to be an efficient method of changing communication parameters as needed. This technique is not yet in an ideal state due to low efficiency of Global Purpose Processors, but coupling this processing with FPGA processing for repetitive tasks that need a higher processing speed is proven to be a good solution. Given that FPGA processing methods are reconfigurable, the flexibility is maintained and the processing is executed faster. The hardware solutions prove to be effective even if there is margin for evolution. The USRP is shown to be a good compromise between

price and functionality. Since the platform allows for a great flexibility with the change of daughterboards, the platform is guaranteed to be a good choice for experiments.

The software solutions regarding SDR are still limited but flexible and complete. With investment in the area and the contribution of the open source community, solutions like SCA and GNURadio will see an evolution in the future and will permit model designing and testing for a wide array of platforms and communication patterns. Since SCA is a more complex solution, for low budget operations and investigations GNURadio will probably be chosen as a testing platform for most academic experiments, enlarging the user base and the number of contributions to the system.

Given the constant evolution in telecommunication technologies and the lower Time-To-Market SDR entails, it will be a good choice when testing and implementing new communication paradigms in the near future.

## Chapter 3

# Cognitive Radio

### 3.1 Introduction

The radioelectric spectrum is a scarce resource. With the exponential increase verified in the last years in consumer electronics, there is a dangerous spectrum shortage right now. This fact needs to be overcome. However, as noted in [MSB06], most of the spectrum is underused, but most of the bands are allocated to several licensed users. This means that there is an opportunity to use those bands while they are vacant. One of the most recent and important applications for the use of this opportunity is the dynamic use and allocation of radio spectrum. The radios that can accomplish this are called Cognitive Radios (CR).

Cognitive Radio, a term introduced in [MMJ99], proposes to analyze the radio spectrum, estimate the state of the channels, predict what is going to happen and manage its spectrum, transmissions and power according to that information. Simon Haykin has defined for Cognitive Radio as follows [Hay05].:

*Cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world), 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:*

- *highly reliable communications whenever and wherever needed;*
- *efficient utilization of the radio spectrum.*

Because of these characteristics, a CR needs reconfigurability. This is achieved by using SDR platforms, introduced in Chapter 2.

Cognitive Radio has been studied widely in [MMJ99] and [Mit00] and it improves the band's utilization. In this dissertation we will focus on the first part of a cognitive radio operation: the Spectrum Sensing.

## 3.2 Spectrum Analysis

The basis of the cognitive radio lies on the analysis of the spectrum, also known as *Spectrum Sensing*. A CR must know the level of occupancy of the radio spectrum in order to take action, and in order to do this it must sweep large ranges of frequencies. This raises the first problem: How does a radio effectively sweep such a large range of spectrum? This dissertation evaluates a response to the question by using a Software Defined Radio platform.

There are two classes of spectrum users, Primary Users (PUs) and Secondary Users (SUs). A PU has licensed access to the spectrum while an SU needs to use it opportunistically in order not to interfere with any PU. Consider the scenario illustrated in Figure 3.1. SU A wants to communicate with SU B but it first has to check if there is spectrum available for its transmission. It is a simple case of A-B transmission.

Considering that radios C and D are communicating in a part of the band sensed by the SUs, the spectrum should appear to these SUs as shown in Figure 3.2. After the sensing period and the conclusion that Channel 2 is being used, the SUs would switch to Channel 1 and transmit their data.

Spectrum sensing warrants thought. What should be the value of the Noise Floor, or Decision Threshold, presented in Figure 3.2? How will a SU be absolutely sure that no one is transmitting on a given frequency? How big a range of frequencies can be sensed in real time? Methods for this depend on what frequencies the CR is sensing, since modulations change for a given technology and technologies change for different frequencies. The

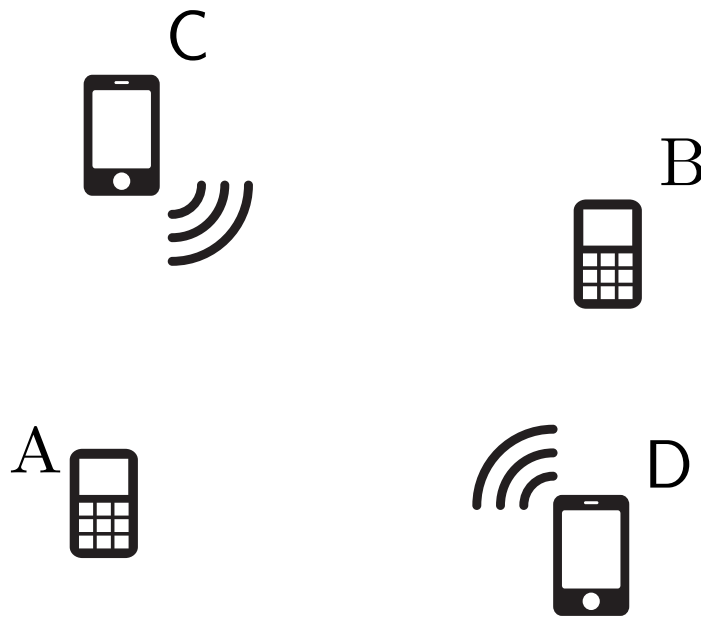


Figure 3.1: Secondary Users A, B and Primary Users C,D.

objective is, however, being able to utilize *all* the spectrum, so one technique is unlikely to work for all technologies and frequencies.

There are limits concerning Spectrum Sensing. Due to the sample rate being entirely connected to the range of frequencies analyzed (Nyquist's theorem), the rate at which data can be gathered, transferred and processed is linked to the frequencies that a cognitive radio can analyze. For example, in a USRP device connected by USB 2.0, the theoretical limit is closely tied to the connection's top speed at 35MB/s. The antenna configuration is another important point. For sub-optimal receiving areas, an antenna might detect a non-existent transmission. The problem is minimized when an array of antennas cooperating in sweeping the frequencies is adopted.

The spectrum sensing technique studied in this dissertation is the Energy Detection, which was theoretically characterized in [Urk67]. It relies on energy sensing and threshold comparison in order to detect the presence of a transmission.

According to [ALL10], the ability to detect a signal involves the evaluation of two hypothesis:

$$\begin{aligned}
 H_0 : y(t) &= n(t) \\
 H_1 : y(t) &= n(t) + s(t)
 \end{aligned}
 \tag{3.1}$$

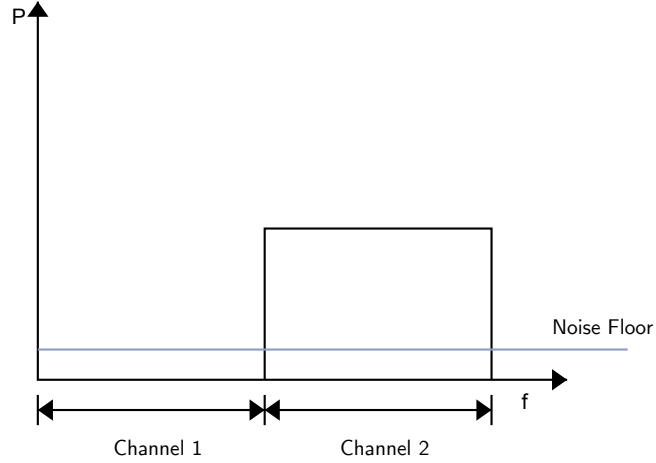


Figure 3.2: RF Spectrum, as sensed by A and B.

where  $y(t)$  represents the received signal,  $s(t)$  denotes the transmitted signal and  $n(t)$  represents the noise experienced during the signal reception.

The noise vector  $n(t)$  is assumed to be additive, white and Gaussian (AWGN) with zero mean and variance  $\sigma_n^2$ . For the purpose of this work, the signal  $s(t)$  is also assumed to be a Gaussian random variable with variance  $\sigma_s^2$ .

In other words,

$$Y \sim \begin{cases} N(\mu_n, \sigma_n^2), & H_0 \\ N(\mu_s + \mu_n, \sigma_s^2 + \sigma_n^2), & H_1 \end{cases} \quad (3.2)$$

The probabilities associated with detecting a transmission are the Probability of Detection ( $P_D$ ) and Probability of False Alarm ( $P_{FA}$ ). These are defined as follows:

$$P_D(\gamma) = Pr[Y > \gamma | H_1] \quad (3.3)$$

$$P_{FA}(\gamma) = Pr[Y > \gamma | H_0], \quad (3.4)$$

$\gamma$  is a decision threshold used to compare the received energy.

Assuming  $N_S$  samples collected during the detection,  $P_D$  and  $P_{FA}$  can be rewritten as [LFO<sup>+</sup>13]:



$$P_D(\gamma) = Q\left(\frac{\gamma - N_S + \theta}{\sqrt{2(N_S + 2\theta)}}\right) \quad (3.5)$$

$$P_{FA}(\gamma) = Q\left(\frac{\lambda - N_S}{\sqrt{2N_S}}\right) \quad (3.6)$$

where  $\theta = \sum_{k=1}^{N_S} (\mu_s / (1 + \sigma_s))^2 = \sum_{k=1}^{N_S} \theta'$  and the Q function is the tail probability of the standard normal distribution:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \quad (3.7)$$

In this dissertation it is assumed that there is *a priori* knowledge of the signal and noise probability distributions. The purpose of this kind of testing is evaluating the performance of an SDR device, in this case a USRP, when performing these tasks.

### Proposed Energy-Based Detection Model

The proposed model in this dissertation is called a **periodogram**. This model computes the Fast Fourier Transform (FFT) of the samples collected by the USRP, translating them into the frequency domain.

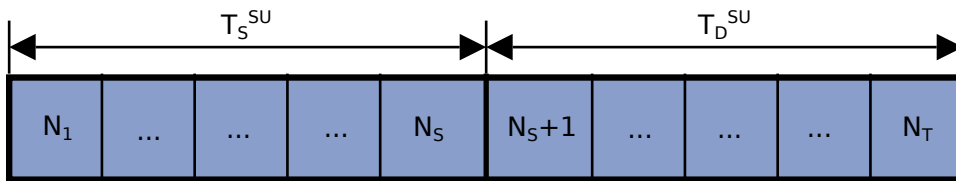


Figure 3.3: Secondary User Frame

In the proposed model the SU has a frame structure that distinguishes two different tasks: *Sampling* and *Transmission*. Sampling happens during the  $T_S^{SU}$  period, which depends on the number of Sensing Samples and the Sampling Rate of the SU. The transmission may occur during the  $T_D^{SU}$ , if the SU detects the spectrum is not occupied. To detect the state of the spectrum, the SU collects  $N_S$  samples and calculates their FFT:

$$X_k = \sum_{n=0}^{N_{FFT}} x_n e^{-i2\pi k \frac{n}{N}}, \quad k = 0, \dots, N_S - 1 \quad (3.8)$$

This spectrum description can allow for some criteria to be used. In this model, these values are averaged, for an average power per sample ( $\bar{Y}_n$ ). Given that the FFT size is equal to the number of sensing samples,  $N_S$ , the average is calculated by

$$\bar{Y}_n = \frac{1}{N_S} \sum_{k=0}^{N_S-1} |X_k|^2 \quad (3.9)$$

Each of these  $\bar{Y}_n$  values is then compared against the decision threshold,  $\gamma$ , to decide if a PU is present or absent.

To validate the detection there is a previous knowledge of whether the PU is present or absent. In the validation, the practical  $P_D$  and  $P_{FA}$  values are calculated directly from the comparison done in the energy detected, being its output given by  $B_n$  and  $C_n$ . To compute  $P_D$  it is assumed that:

$$B_n = \begin{cases} 0, & \bar{Y}_n < \gamma | H_1 \\ 1, & \bar{Y}_n > \gamma | H_1 \end{cases} \quad (3.10)$$

To compute  $P_{FA}$ ,  $C_n$  is defined as follows:

$$C_n = \begin{cases} 0, & \bar{Y}_n < \gamma | H_0 \\ 1, & \bar{Y}_n > \gamma | H_0 \end{cases} \quad (3.11)$$

Each of the probabilities is then calculated by:

$$\begin{aligned} P_D &= \frac{1}{M} \sum_{n=0}^M B_n \\ P_{FA} &= \frac{1}{M} \sum_{n=0}^M C_n \end{aligned} \quad (3.12)$$

where  $M$  is the number of decisions. There are some trade-offs to consider when using this approach. Parameters such as the ratio between  $N_S$  and  $N_T$ , the desired frequency band and the desired  $P_D$  and  $P_{FA}$  values.

### Decision Threshold Estimation in Presence of a Signal

The aforementioned methods work for cases where the noise power and the presence or absence of a PU are known. However, in cases where the SU knows the PU is transmitting but has no means of knowing the noise values, the threshold can not be calculated by inspection. Some methods were already defined to calculate the value for this threshold when no prior knowledge about noise is available.

The methods studied in this dissertation are the Minimum Description Length (MDL) and the Akaike Information Criteria (AIK), derived from [WP98] and [WK85]. These algorithms also depend on the presence of a signal. Both algorithms use eigenvalues to in the decision process. However, in this case the eigenvalues are considered to be the values of a sorted periodogram, calculated by:

$$P_k = |X_k|^2 \quad (3.13)$$

It is then considered that the eigenvalue of the  $k$ -th FFT is

$$\lambda_i = P_k \quad (3.14)$$

The AIC and MDL values are then calculated by [SMS12]:

$$AIC(k) = (N - k)L\log(\alpha(k)) + k(2N - k) \quad (3.15)$$

and:

$$MDL(k) = (N - k)L\log(\alpha(k)) + k/2(2N - k)\log(L) \quad (3.16)$$

where:

$$\alpha(k) = \frac{\sum_{i=k+1}^N \lambda_i}{N-k} \frac{1}{\left(\prod_{i=k+1}^N \lambda_i\right)^{\frac{1}{N-k}}} \quad (3.17)$$

and  $N$  represents the periodogram FFT size,  $L$  denotes the number of times the periodogram was averaged,  $k$  is the model number and  $\lambda_i$  represents the eigenvalues, approx-

imated by  $P_k$ .

Both algorithms evaluate the periodogram and draw conclusions in regard to its shape.

The value  $k$  is the model number and it ranges from the values 1 to  $N$  (FFT Size). For each of these values there is an MDL and AIC corresponding value. The model number with the lowest AIC and MDL value is defined as the threshold for the signal power in the averaged periodogram. Up to that  $k$  value, a primary user is considered as being active (detected). From  $k$  to  $N$  a channel is declared as idle, allowing the calculation of the average noise power.

These two models were tested for a periodogram generated in MATLAB. The periodogram represents a sine wave centered at 30 Hz and sideband AWGN Noise with its power centered at 1. This periodogram is presented in Figure 3.4.

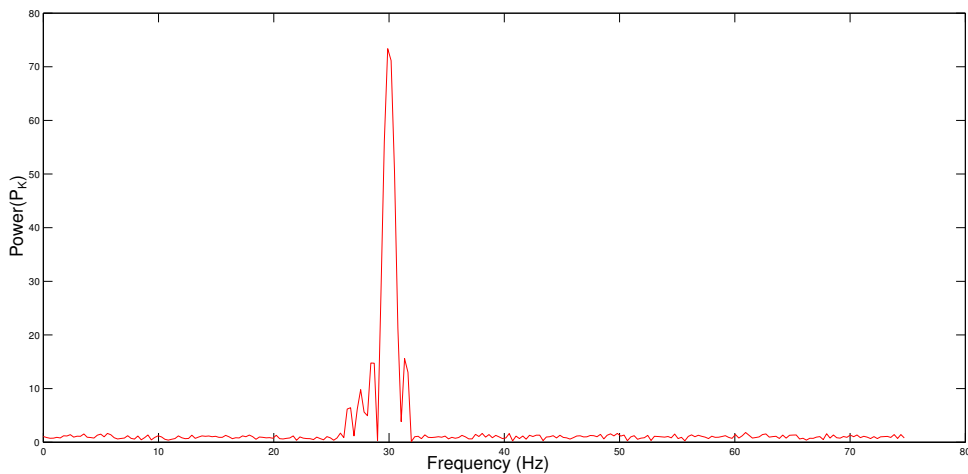


Figure 3.4: Sine wave with sideband Gaussian Noise

The AIC and MDL results, calculated with Equations 3.15 and 3.16 are presented in Figure 3.5. The minimum value for both of these methods is present in  $k = 18$ . This means that in the sorted periodogram (Figure 3.6), all values up to the 18th FFT Bin contain a signal and all values upwards contain noise, allowing for the calculation of the noise figure.

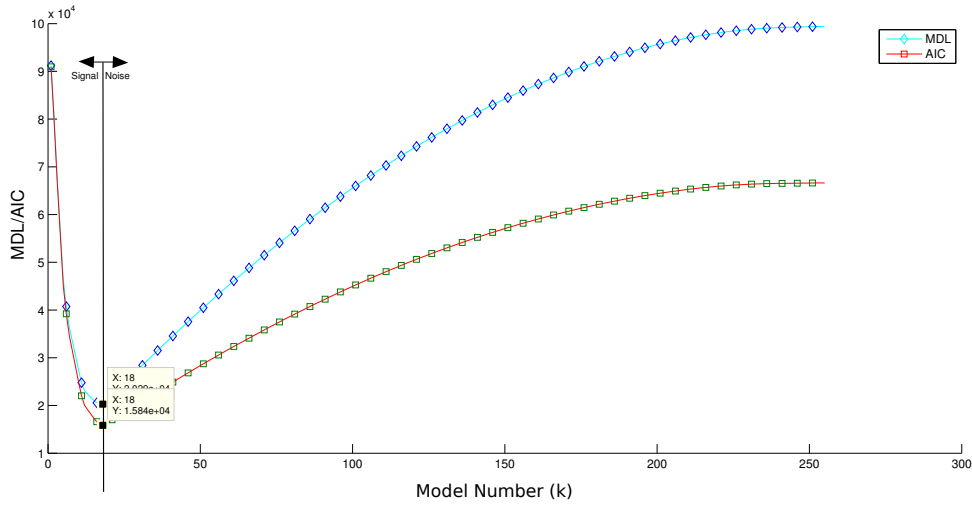


Figure 3.5: AIC and MDL values for a Sine wave with Gaussian Noise

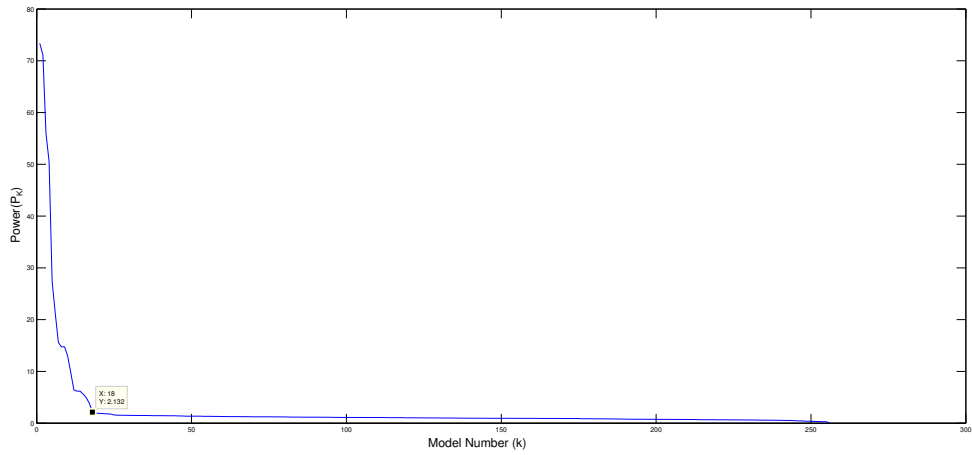


Figure 3.6: Sorted Sine wave periodogram with Gaussian Noise

With this data, the noise figure was calculated through averaging all points in Figure 3.6 from the value  $k_{AIC}$  where  $AIC(k_{AIC}) = \min(AIC)$  and  $k_{MDL}$  where  $MDL(k_{MDL}) = \min(MDL)$  upwards. The results are presented in Table 3.1.

Waveform	Real Noise Figure	Noise Estimation (AIC)	Noise Estimation (MDL)
Sine+Noise	1.00	1.00 ( $k_{AIC}=18$ )	1.00 ( $k_{MDL}=18$ )

Table 3.1: MDL, AIC Noise values estimation

The results approximate the noise figure. Since this approach is based on models that

take into account the relationship between the power throughout the whole spectrogram and not particular waveforms, the estimation is very accurate. It correctly identifies the peak as being the signal, singles out the power values for that peak and averages the other samples in the periodogram for a perfect estimation of the noise figure.

There is a parameter that can be varied with a high effect on the output,  $L$ . In this experiment,  $L$  was set to 1000. The ideal value for this parameter would always depend on a case by case scenario, since the bandwidth and the amplitude of the sensed wave influence the output of the criteria.

### 3.3 Conclusions

The Cognitive Radio is a necessary evolution for smart spectrum usage. While complex solutions might present a good option in some cases, a simple solution like an Energy Detector might be useful for situations where there is a noticeable difference between the noise and signal values. A model for energy detection was presented considering a noise and signal normal distribution for simplicity. The extension of such a model for other signal types might come from calculating other kinds of distribution regarding signal values. Two models regarding signal detection that follow information theory and use FFT values as eigenvalues were presented and tested with theoretical cases. The result using these models was positive, detecting existing narrowband signals in the spectrum.

## Chapter 4

# Experiments on EBS Parametrization

### 4.1 Testbed Description

The proposed EBS model was tested on a setup with two USRP B100 connected to a PC with the following specifications:

- OS: Ubuntu 12.04 64 bit
- Memory: 4GB DDR3
- Processor: Intel Core i5-2320 @ 3.00GHz x 4
- Hard Drive: 320GB @ 7200rpm

The GNURadio version installed was the 3.7.1 and the USRP driver was UHD 003.005.003, the latest versions available at the time of the experiment.

The two USRPs consist of a motherboard with a Xilinx Spartan 3A 1400 FPGA, 64MS/s dual ADC, 128MS/s dual DAC and USB 2.0 connection. These motherboards used as a front-end device an SBX 400-4400 MHz Rx/Tx Daughterboard, with two RF ports: a transceiver (TX/RX) and a receiver (RX2). The daughterboard provides up to 100mW of output power and a typical noise figure of 5dB.

The USRPs were connected to each other (TX/RX to RX2) through a coaxial cable and two 30dB, 50Ohm attenuators to achieve lower SNR when transmitting.

## 4.2 Experiment Description

The experiment consisted of three steps, described in detail in the following Subsections:

1. Determine, noise and signal data, determine thresholds
2. Collect data with regard to calculated thresholds
3. Detector characterization

The main objective is the comparison between the theoretical distribution of probability and the real distribution, by calculating the practical  $P_D$  and  $P_{FA}$  using the USRP in real-time.

To achieve that, a python script was built in which the USRP parameters are defined. The steps were done as follows:

### 4.2.1 Noise and Signal Data Computation

The script allows the user to select parameters such as:

- Frequency (Hz) - Operational frequency of the SU and the PU (in case of PU transmission);
- Sample Rate (Samples/s) - The sample rate at which the SU and the PU receive and transmit data. The observed spectrum bandwidth follows Nyquist's rule,  $F_s/2$ ;
- Packet Size (Samples) - The number of samples of a packet ( $N_F$ );
- Sensing Samples (Samples) - The number of sensing samples in a packet ( $N_S$ );
- Transmitter Gain (dB) - The output gain on the PU in case of transmission;
- Receiver Gain (dB) - The gain of the SU;
- Sensing Time (s) - The required for calibration;
- Noise File Name (String) - The name of the file where noise data is stored;
- Signal File Name (String) - The name of the file storing PU's signal data;



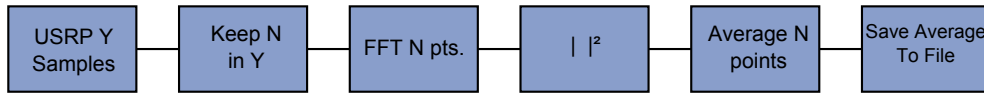


Figure 4.1: Secondary User GNURadio block diagram



Figure 4.2: Primary User GNURadio block diagram

The script then does two data collections: one where the PU is transmitting and one where it is not. The GNURadio design blocks for the PU and the SU are illustrated in Figures 4.1 and 4.2.

The  $\mu, \sigma$  values of a normal approximation of the values present in the averaged periodogram are then calculated through the use of the Python NumPy library.

Three files are obtained: a binary file containing the signal spectrogram averages, a binary containing the noise spectrogram averages and a file containing the parameters used in the transmission.

The decision threshold range is defined on the lower bound by the minimum value of the noise spectrogram and on the higher bound by the maximum value of the signal spectrogram.

#### 4.2.2 Data gathering with parametrized thresholds

This part of the script loads the .dat files associated with noise and signal and the properties.p file related with the operational parameters. In this part of the experiment, a threshold block is added to the signal processing unit. This threshold was set for a  $P_{FA}$  value of 5%, according to the theoretical  $P_{FA}$  curve. For each previously defined threshold the block diagram in Figure 4.3 is executed and the results registered, first with the PU transmitting and then with the PU silent.

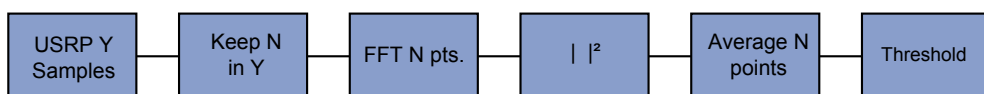


Figure 4.3: Secondary User Threshold block diagram

### 4.2.3 Detector Characterization

The Energy Detector was characterized by calculating its Probability of Miss-Detection,  $P_{Miss} = (1 - P_D)$ , after setting a threshold value (in this experiment, at  $P_{FA} = 5\%$ ) and changing the transmitter power, thus changing the SNR. The script allows the user to choose what range of SNR is desired in comparison with the original, as well as the number of periodogram averages that should be collected. It then calculates the  $P_{Miss}$  values for each experiment and its corresponding value of SNR, increasing the SNR value for each iteration. This allows the evaluation of the Energy Detector's performance under a wide range of SNR.

## 4.3 Experimental Results

The first experiment was done assuming the following set of parameters:

Frequency (GHz)	$N_S$	$N_T$	Sample Rate	SU Gain (dB)	PU Gain (dB)	SNR (dB)	Primary AWGN Amplitude
1.3	256	5120	1,000,000	10	10	-6.5	0.0001

Table 4.1: USRP Parameters for Experiment

The histogram of the collected data and the approximated normal distribution calculated with basis in this collected data for noise and signal characterization can be seen in Figure 4.4.

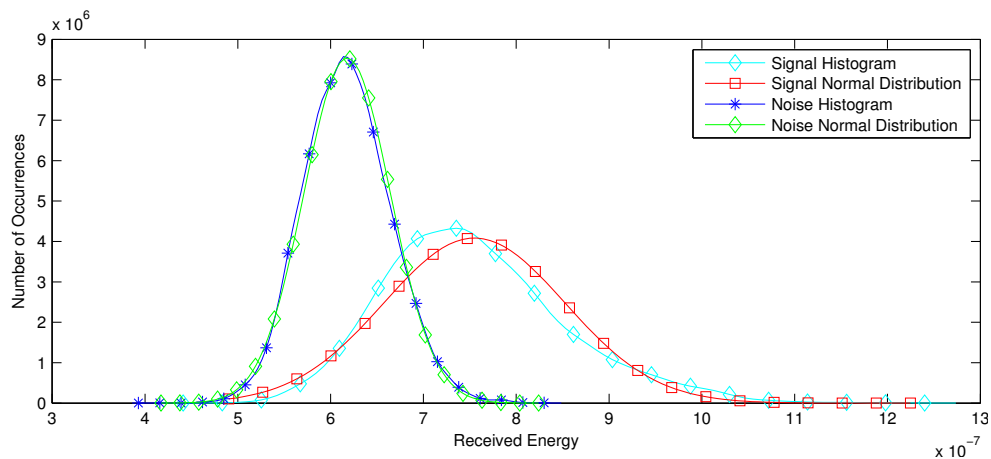


Figure 4.4: Histogram and Normal Distribution of First USRP Experiment

These distributions allowed the  $P_D$  and  $P_{FA}$  values to be calculated through Equations (3.5) and (3.6). These are the theoretical values. The next test consisted of sweeping these threshold values from the minimum noise energy value to the maximum signal energy value. Each of these thresholds was tested for 2.5 seconds of sensing. The resulting  $P_D$  and  $P_{FA}$  are represented in Figure 4.5. As can be seen, the practical value also

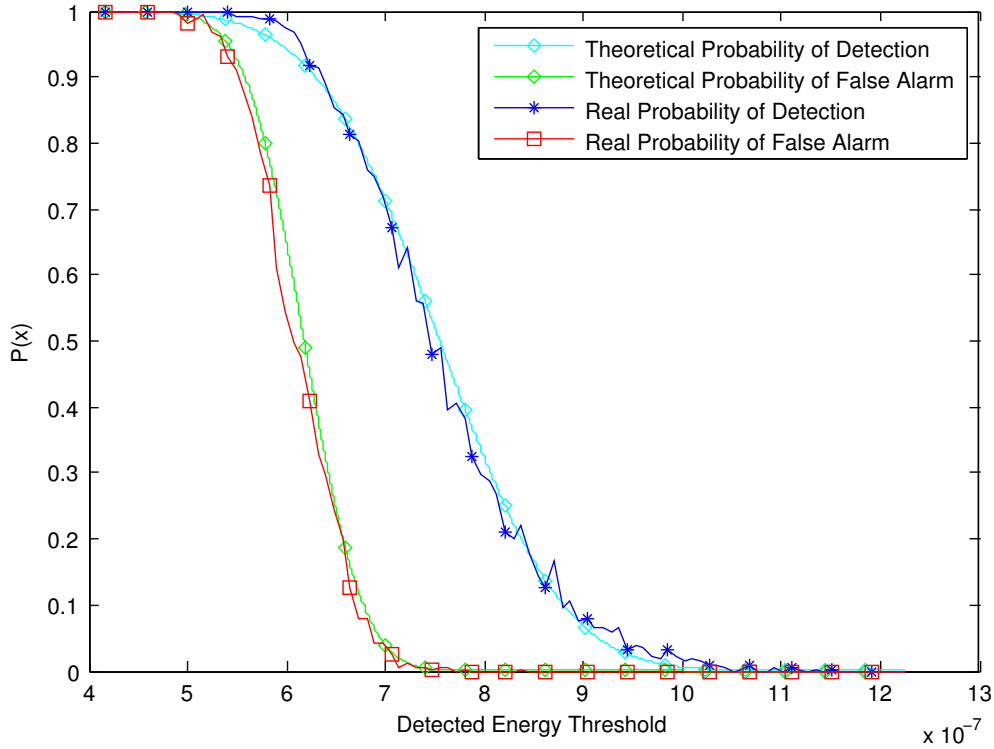
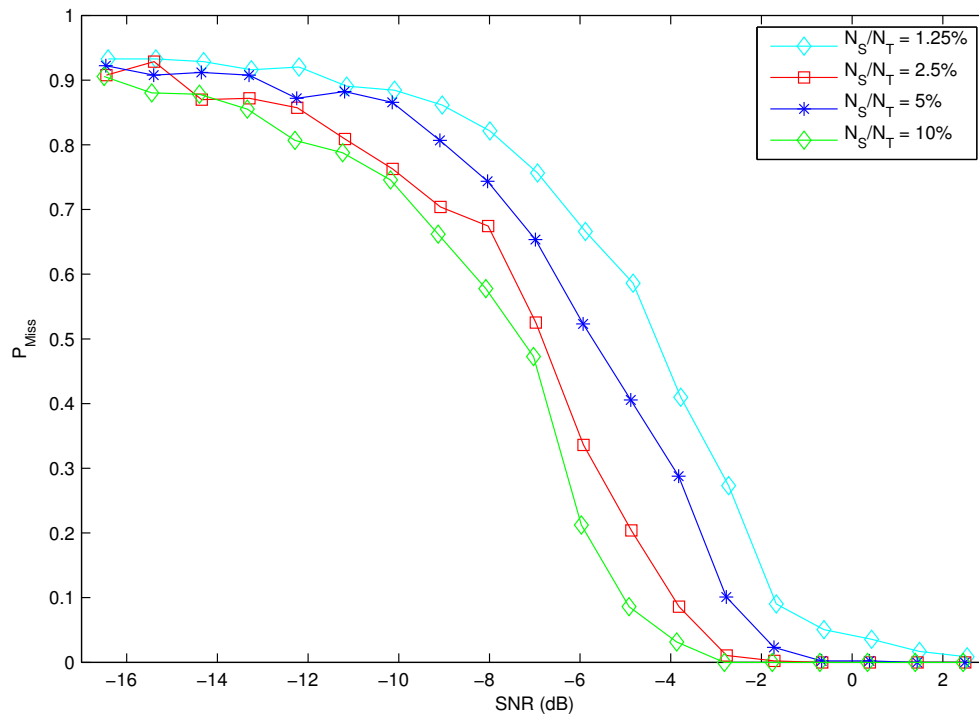
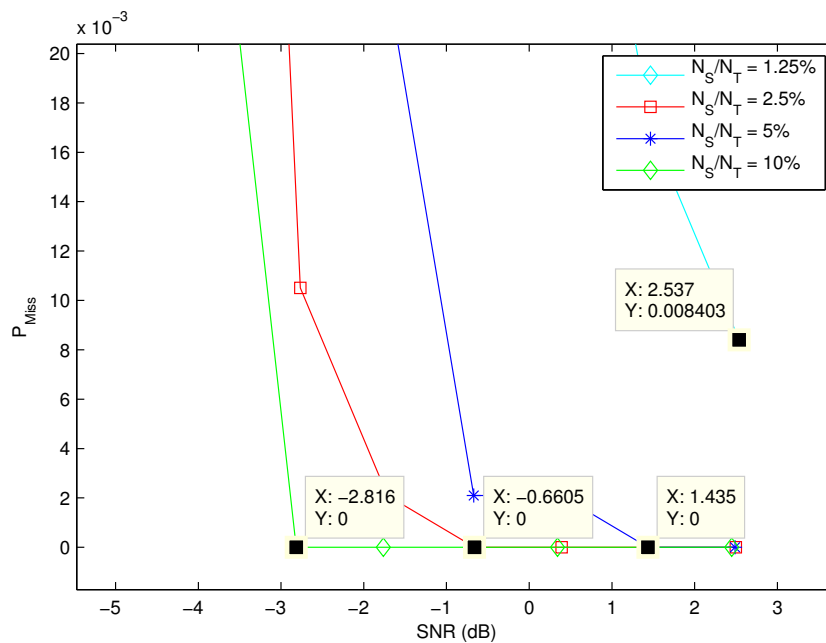


Figure 4.5:  $P_D$  and  $P_{FA}$  for the first Experiment

approaches the theoretical one. The detector was then characterized for a range of SNR. For this same experiment, the threshold value at which the theoretical value of  $P_{FA}$  is 5% was inspected from Figure 4.5. This value was then set as the threshold for the characterization experiment. The SNR was swept and the  $P_{Miss}$  values were computed varying the  $N_S/N_T$  ratio. The results are depicted in Figure 4.6. As can be seen, the probability of miss-detection decreases as the SNR increases. Moreover, the utilization of more sensing samples per frame (higher  $N_S/N_T$  increases the performance in terms of miss-detection but also increases the sensing overhead per frame. As can be seen in Figure 4.7, which presents a zoom of the data presented in Figure 4.6 near 0 dB, if the goal is to use the sensing to achieve approximately 0% in terms of probability of miss-detection ( $P_{MISS}$ ), that is, always detect the PU, using a ratio  $N_S/N_T$  of 10% gives the user a gain

Figure 4.6:  $P_{Miss}$  for a range of SNRFigure 4.7:  $P_{Miss}$  detail for a range of SNR

of 2.155dB over a ratio of 5%.

While the  $P_{Miss}$  performance of the detector is stable and reaches the desired results, the  $P_{FA}$  performance deviates from the initial desired value of 5%, as illustrated in Figure 4.8. This is due to the difference between the approximated normal distribution and the histogram in regards to the noise and signal figures (Figure 4.4). While theoretically the noise value for all experiments would be the same, with lower  $N_S/N_T$  values there is a slight deviation. For a real  $P_{FA}$  of 5%, only the  $N_S/N_T = 10\%$  ratio ensures a safe interval.

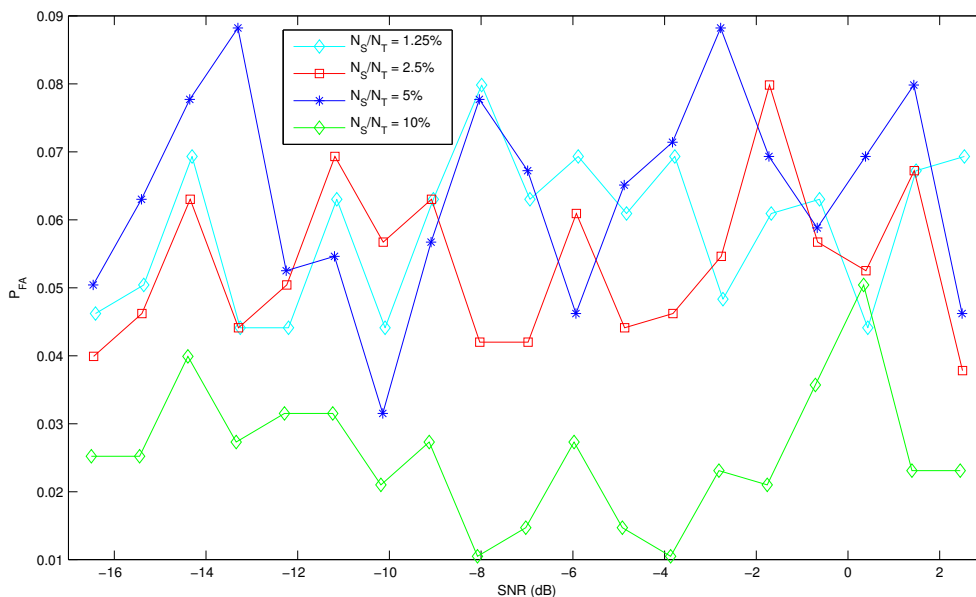


Figure 4.8:  $P_{FA}$  for a range of SNR

## 4.4 Noise Floor Estimation Experiment

The noise floor estimation techniques presented in Subsection 3.2 were evaluated in a real scenario. FFT values were computed during a given amount of time and then their AIC and MDL noise estimation were computed using the computed FFT values. This was done with the same USRP parameters as in Section 4.3.

### 4.4.1 First Experiment - Gaussian Noise

The First Experiment was conducted without considering PU's transmission and was performed in a Gaussian noise channel to observe the results generated by the algorithms

where only Gaussian noise is considered. The power figures needed to be normalized for the algorithm to work, since the product of the low range of power values received by the USRP would invariably produce results converging to -infinite. The averaged periodogram of the first channel sampling is illustrated in Figure 4.9.

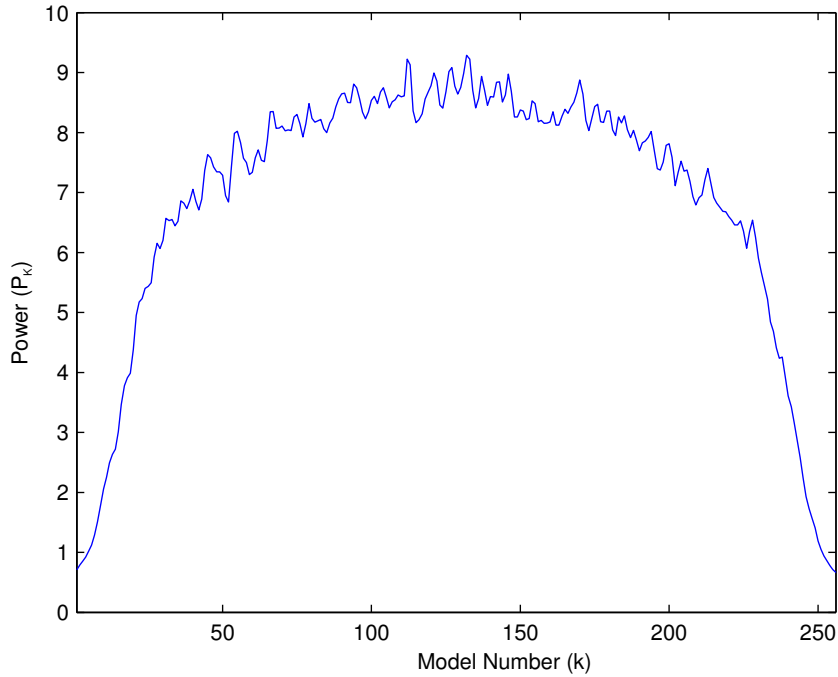


Figure 4.9: Periodogram for a Gaussian Noise transmission

The figure shows the filtering done by the USRP itself. While the amplitude of the Gaussian noise is high, the algorithm still determines the whole signal to be noise by both the AIC and MDL criteria. This is justified by the fact that the minimum value of both models (AIC and MDL) occurs at  $k = 0$ , hence no part of the data is detected as being a signal). This can be seen in Figure 4.10.

The conclusion that the whole spectrum contains noise is correct in this case. However, for a small window and signals where that window is fully occupied by, the algorithm will not detect the signal as it should, requiring a higher sample rate for the signal to be noticeable in the spectrum.

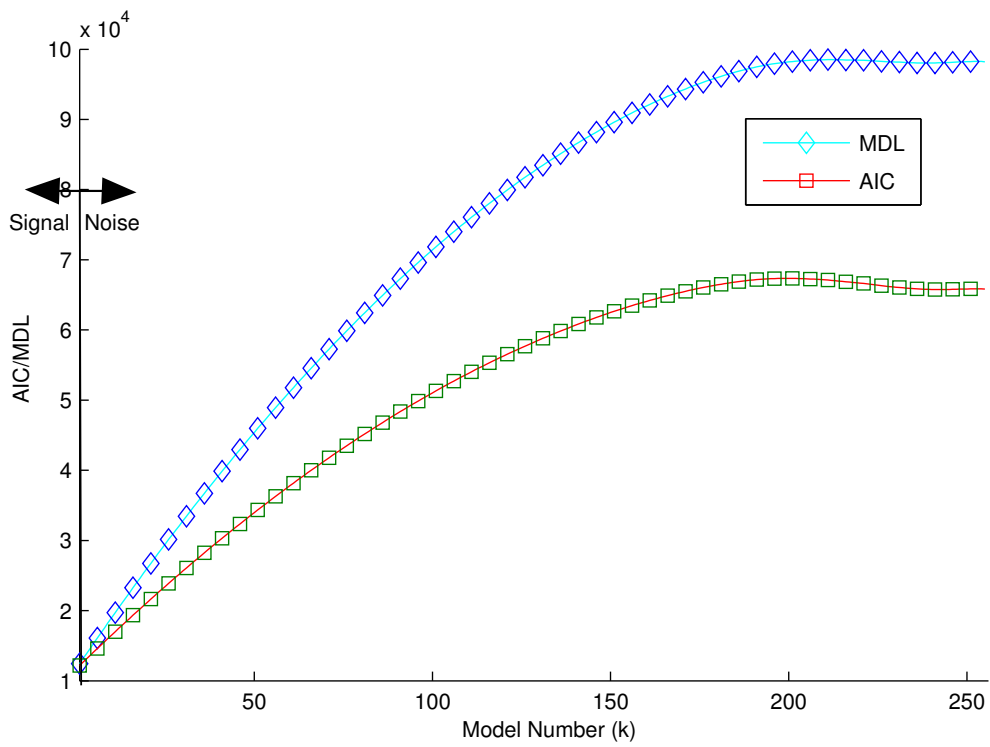


Figure 4.10: AIC and MDL values for a Gaussian Noise transmission

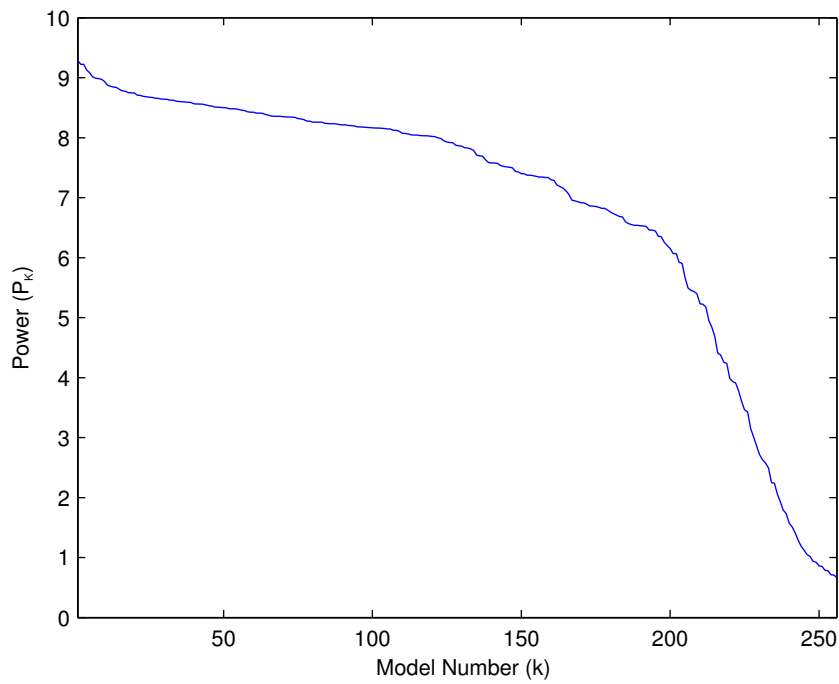


Figure 4.11: Sorted periodogram for a Gaussian Noise transmission

#### 4.4.2 Second Experiment - Sine Wave

The Second Experiment consisted of transmitting a single sine wave from one USRP to the other. The periodogram is represented in Figure 4.12 and the results for the criteria are represented in Figure 4.13. Figure 4.14 represents the sorted periodogram of the transmission.

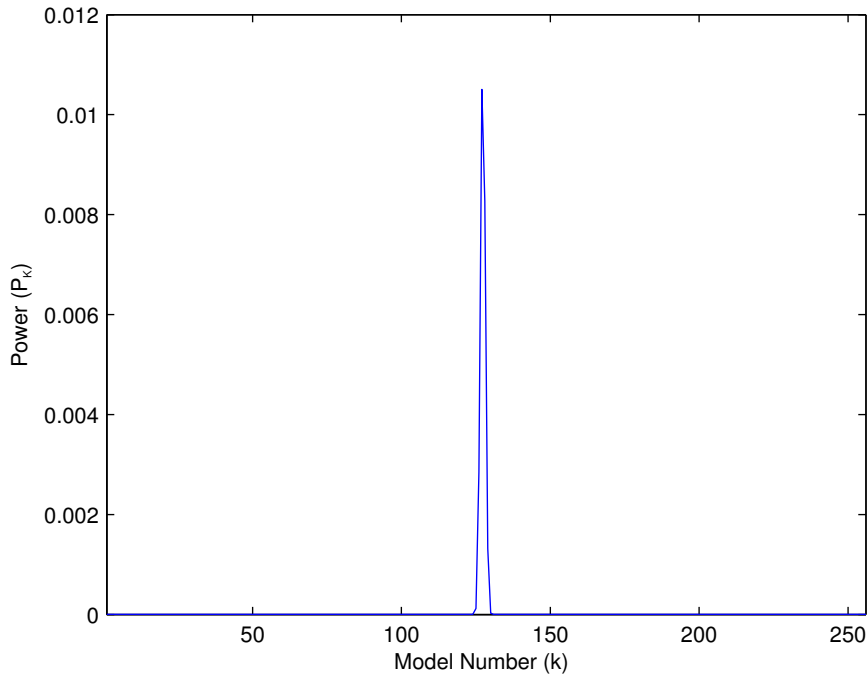


Figure 4.12: Periodogram for a Sine wave transmission

As can be seen in Figure 4.13, the division between the noise and signal values is at  $k=7$ , due to the fact that only 7 bins in the FFT contain the signal values. This means that in the ordered periodogram the power of the noise value is the average for all samples at  $k=7$  bins. The Sine Wave was correctly detected due to the high sample rate with regards to the signal's spectral occupation. This demonstrates that for a signal whose power is noticeable on the spectrum, the algorithm shows good results.

#### 4.4.3 Third Experiment - Three-Tone Wave

The third experiment consisted of transmitting a three-tone wave, consisting of a sum of three Sine waves. The experiment's objective was to obtain more information on



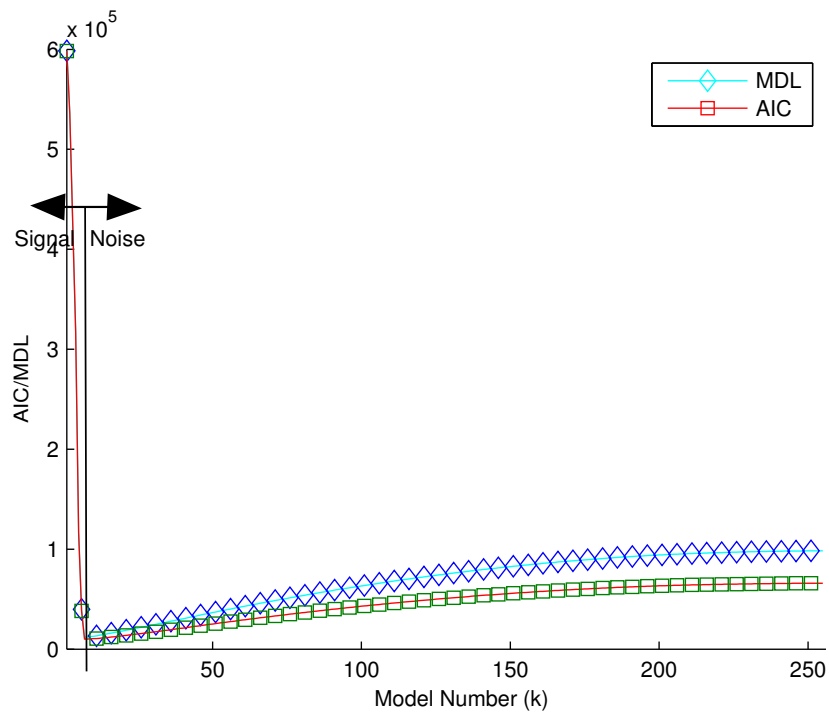


Figure 4.13: AIC and MDL values for a Sine wave transmission

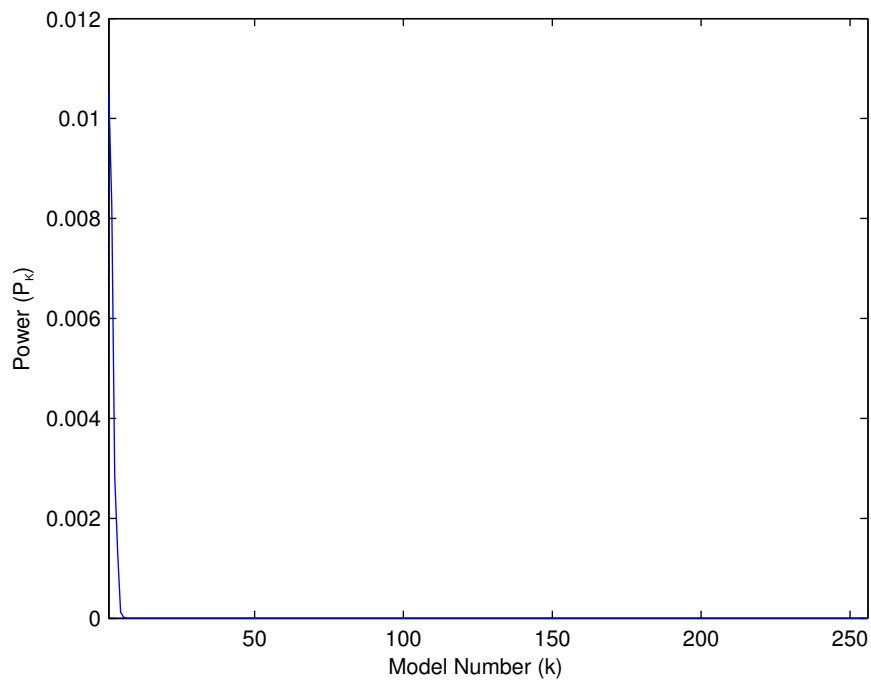


Figure 4.14: Sorted periodogram for a Sine wave transmission

how the models would react to a higher number of narrowband signals present in the periodogram. The periodogram is presented in Figure 4.15.

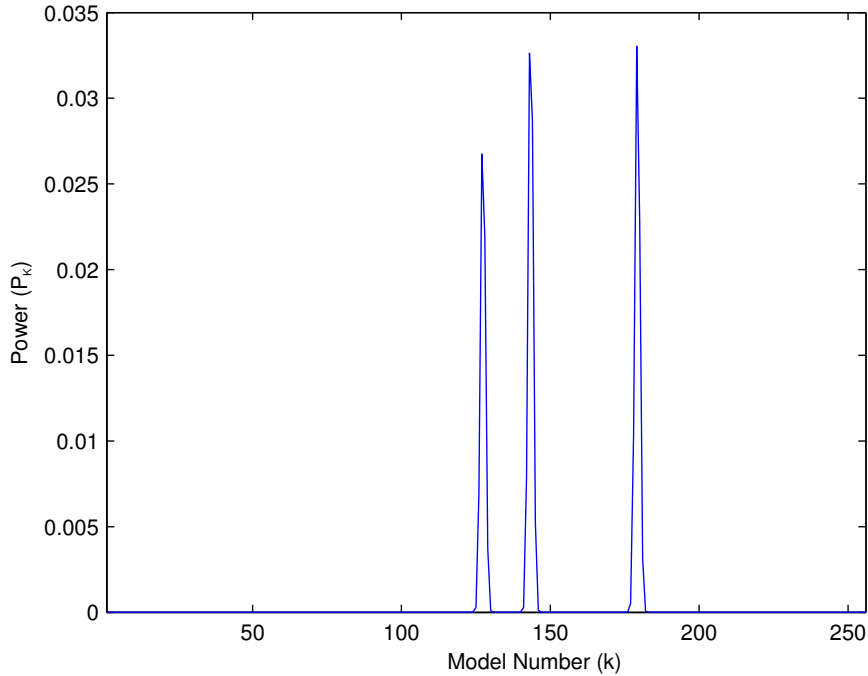


Figure 4.15: Periodogram for a Three-Tone wave transmission

As can be seen in 4.16, the values for the number of bins start to variate for the two algorithms. For the MDL model, the value at which the minimum can be located is  $k=15$ . For the AIC model, this value is  $k=18$ . This means that for the MDL model, every value in the ordered periodogram from the bin number 15 upwards is classified as noise and for the AIC model every bin starting at the number 17 is also noise. An average between the two can be used later for a better approximation.

Figure 4.17 represents the ordered periodogram for the third experiment. A conclusion can be made that the models react properly to the increase of signals present in the periodogram, since the value of  $k$  is higher than the one where only one sine was transmitted.

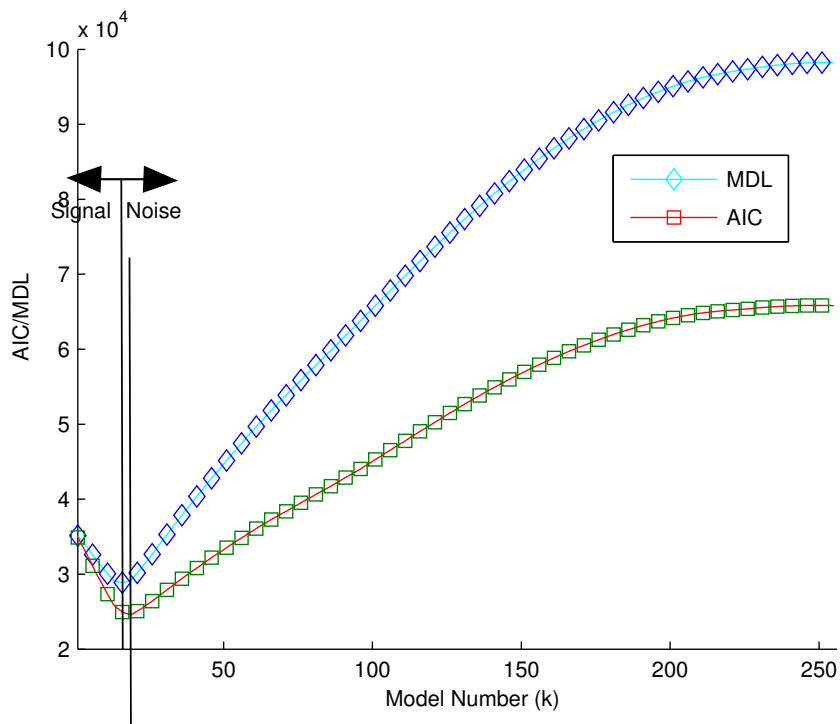


Figure 4.16: AIC and MDL values for a Three-Tone wave transmission

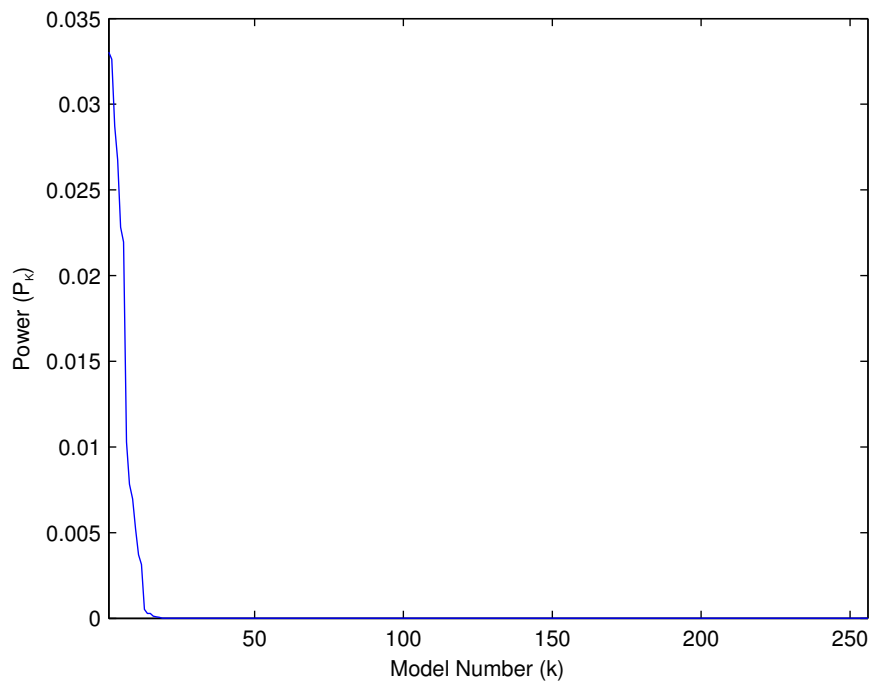


Figure 4.17: Sorted periodogram for a Three-Tone wave transmission

## 4.5 Conclusions

This chapter presented experiment results regarding threshold calculation through inspection, comparison between theoretical and practical probabilities of detection and false alarm, and testing of the theory information criteria AIC and MDL. The practical values of  $P_D$  and  $P_{FA}$  approached the theoretical values. The characterization of the energy detector showed that for a relatively high SNR the energy detector may be used as an effective and accurate detection method. The use of the theory information criteria allowed the detection of narrowband signals, which implies it might be a good way to detect a signal by itself.

# Chapter 5

## Conclusions

### 5.1 Final Considerations

This work was produced with the objective of testing the feasibility of using an SDR system for Spectrum Sensing for cognitive radios. While the processing tools available nowadays do not allow for communication and processing as fast as application specific circuits, the solutions existent nowadays present a good compromise between speed and flexibility. The use of an FPGA for some functions allows the GPP to focus on other non-priority functions, presenting a good solution for the majority of communication schemes.

The performance by the USRP devices while the tests were being run shows that they technically provide the required features for testing systems. The software suite GNURadio is still underdeveloped and recommended mostly for practical testing in laboratory. However, there has been a big push by the open source community to improve the overall system and allow it to be more accessible for new users. The proposed solution for an Energy Detector showed useful for certain circumstances. The theoretical and practical values for the probabilities of detection and false alarm showed a good convergence for a low sensing time. This was, however, for an AWGN signal, where the whole spectrum shows increased power. For narrowband signals, the energy detector might underperform. However, for these narrowband signals, the MDL and AIC solutions were tested and proven to be useful. These algorithms, originally used with eigenvalues, when used in FFT values showed that the signals were correctly detected and a good noise floor estimation was achieved. The objective of building a know-how regarding the GNURadio, USRP

and SDR systems in general was also accomplished. An easy-to-use script was built using the GNURadio platform, still allowing for change in the algorithms and further testing on different circumstances by rewriting the Python code of the script. This allows for an easier use for future work in the area.

## 5.2 Future Work

The coupling of the two studied detection techniques could be used. The SUs could collect the data, calculate the FFT, and then apply both detection methods for both narrow and wideband signals, allowing for a more robust system. This could be achieved by changing the scripts built during the dissertation.

A conclusion can also be made regarding different signal types and the use of the Energy Detector. For this dissertation the signal was considered to be Gaussian. However, by adding options to the PU signal type, more characterizations can be made.

# Bibliography

- [ALL10] Erik Axell, Geert Leus, and Erik G Larsson. Overview of spectrum sensing for cognitive radio. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 322–327. IEEE, 2010.
- [ALRD<sup>+</sup>08] G. Abgrall, F. Le Roy, J.P. Delahaye, J.P. Diguët, and G. Gogniat. A comparative study of two software-defined radio environments. In *SDR'08 Technical Conference and product Exposition*, page Session 4.2, Oct. 2008.
- [BCMR02] J. Bertrand, John W. Cruz, B. Majkrzak, and T. Rossano. Corba delays in a software-defined radio. *IEEE Communications Magazine*, pages 152–155, Feb. 2002.
- [Doh02] Dave Dohse. Successfully introducing corba into the signal processing chain of a software-defined radio, 2002.
- [Ett] <http://www.ettus.com/>, 2013.
- [Hay05] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, VOL. 23, NO. 2,, pages 201–220, Feb. 2005.
- [Hum06] F. Humcke. Making fpgas 'first class' sca citizens. In *2006 Software Defined Radio Technical Conf. Product Exposition*, volume -, Mar. 2006.
- [KR07] I. Kuon and J. Rose. Measuring the gap between fpgas and asics. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, pages 203–215, Feb. 2007.

- [LBJ06] D. Lau, J. Blackburn, and C. Jenkins. Using c-to-hardware acceleration in fpgas for waveform baseband processing. In *Software Defined Radio Technical Conf. Product Exposition*, Nov. 2006.
- [LFO<sup>+</sup>13] M. Luis, A. Furtado, R. Oliveira, R. Dinis, and L. Bernardo. Towards a realistic primary users' behavior in single transceiver cognitive networks. *Communications Letters, IEEE*, 17(2):309–312, February 2013.
- [MBF05] Kurdziel M., J. Beane, and J.J. Fitton. An sca security supplement compliant radio architecture. In *Military Communications Conference*, pages 2244–2250 Vol. 4, Jan. 2005.
- [MD14] L. Bernardo R. Dinis R. Oliveira M. Duarte, A. Furtado. Practical assessment of energy-based sensing trhogh software defined radio devices. In *5th Doctoral Conf. on Computing, Electrical and Industrial Systems (DoCEIS 14)*, Apr. 2014.
- [Mit95] Joe Mitola. The software radio architecture. *Communications Magazine, IEEE*, 33(5):26–38, 1995.
- [Mit00] Joseph Mitola. Cognitive radio—an integrated agent architecture for software defined radio. *Royal Institute of Technology (KTH)*, 2000.
- [MMJ99] Joseph Mitola and Gerald Q Maguire Jr. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.
- [Moo06] Gordon E et al Moore. Cramming more components onto integrated circuits. In *Electronics, Volume 38, Number 8*, Apr. 2006.
- [MSB06] Shridhar Mubaraq Mishra, Anant Sahai, and Robert W Brodersen. Cooperative sensing among cognitive radios. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 4, pages 1658–1663. IEEE, 2006.
- [Pos] <http://http://posixcertified.ieee.org/>, 2013.



- [PRL<sup>+</sup>12] M. Palkovic, P. Raghavan, M. Li, A. Dejongue, P. Liesbet Van der, and F. Catthoor. Future software-defined radio platforms and mapping flows. *IEEE Signal Processing Magazine*, March 2010, pages 22–33, Mar. 2012.
- [RYZ11] Yi Ren, Dongping Yao, and Xianhui Zhang. The implementation of tetra using gnu radio and usrp. In *Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications (MAPE), 2011 IEEE 4th International Symposium on*, pages 363–366. IEEE, 2011.
- [SCA] [http://jpeojtrs.mil/sca/Documents/SCAv4\\_0/SCA\\_4.0\\_20120228\\_ScaSpecification.pdf](http://jpeojtrs.mil/sca/Documents/SCAv4_0/SCA_4.0_20120228_ScaSpecification.pdf), 2013.
- [SIW06] J.L Shanton III and H. Wang. Design considerations for size, weight and power constrained radios. In *2006 Software Defined Radio Technical Conf. Product Exposition*, Nov. 2006.
- [SMS12] S. Sequeira, R.R. Mahajan, and P. Spasojevic. On the noise power estimation in the presence of the signal for energy-based sensing. In *Sarnoff Symposium (SARNOFF), 2012 35th IEEE*, pages 1–5, 2012.
- [SRLM12] A.F.B Selva, A.L.G. Reis, K.G. Lenzi, and L.G.P. Meloni. Introduction to the software-defined radio approach. *IEEE Latin America Transactions*, Vol. 10, No. 1, pages 1156–1161, Jan. 2012.
- [TT09] D.C. Tucker and G.A. Tagliarini. Prototyping with gnu radio and the usrp - where to begin. In *Southeastcon, 2009. SOUTHEASTCON '09. IEEE*, pages 50–54, March 2009.
- [Ulv10] T. Ulversøy. Software defined radio: Challenges and opportunities. *IEEE Communications survey and tutorials*, Vol. 12, No. 4, pages 531–550, Oct. 2010.
- [Urk67] Harry Urkowitz. Energy detection of unknown deterministic signals. *Proceedings of the IEEE*, 55(4):523–531, 1967.
- [Wir] <http://wirelessinnovation.org/>, 2013.

- [WK85] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):387–392, 1985.
- [WP98] Xiaodong Wang and H.V. Poor. Blind multiuser detection: a subspace approach. *Information Theory, IEEE Transactions on*, 44(2):677–690, 1998.

# Appendixes



# Appendix A

## DOCeis Article

This appendix contains the Article presented in the DOCeis 2014 Conference.



# Practical Assessment of Energy-based Sensing through Software Defined Radio Devices

Miguel Duarte<sup>1</sup>, A. Furtado<sup>1,2</sup>, L. Bernardo<sup>1,2</sup>, R. Dinis<sup>1,2</sup>, R. Oliveira<sup>1,2</sup>

<sup>1</sup> CTS, Uninova, Dep.º de Eng.ª Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516, Caparica, Portugal

<sup>2</sup> IT, Instituto de Telecomunicações, Portugal

**Abstract:** The Cognitive Radio is a solution proposed for the increasing demand of radio spectrum. Usually cognitive radios are adopted by the non-license wireless users, which have a certain degree of cognition in order to only access to a given frequency band when the band is sensed idle. However, these bands and their use must be assessed, to avoid interfering with licensed users (primary users). The way to assess band's occupancy is by discerning between just noise or noise plus signal. In this paper, energy-based sensing (EBS) is considered through the use of a classical energy detector. The work proposes and describes an implementation of an energy detector using a software defined radio (SDR) testbed and, after computing the probability of detection and false alarm from a real set of samples obtained with the SDR devices, we successfully validate a theoretical model for the probabilities. EBS' performance is validated for several points of operation, i.e. for different signal-to-noise-ratio (SNR) values. These findings may be useful for building an EBS detector that defines its own decision threshold in real time given the target probabilities, since the formal probabilities are successfully validated. Moreover, our contribution also includes a detailed description of the implemented blocks using GNU Radio's open-source software development toolkit.

**Keywords:** Software Defined Radio, Energy Sensing, System Performance.

## 1 Introduction

Cognitive Radio (CR) has been proposed as an effective answer to alleviate the increasing demand for radio spectrum [1]. CR nodes, usually denominated Secondary Users (SUs) due to its non-licensed operation, must be aware of the activity of the licensed users, denominated Primary Users (PUs), in order to dynamically access the spectrum without causing them harmful interference.

Spectrum Sensing (SS) aims at detecting the availability of vacant portions (holes) of spectrum and has been a topic of considerable research over the last years [1]. It plays a central role in CR systems. The traditional SS techniques include Waveform-based sensing (WBS) [2], a coherent technique that consists on correlating the received signal with *a priori* known set of different waveform patterns; Matched Filter-based sensing (MFBS) [3], an optimal sensing scheme where the received signal is also correlated with a copy of the transmitted one; and Cyclostationarity-based sensing (CBS) [4], a technique that exploits the periodic characteristics of the

received signals, *i.e.*, carrier tones, pilot sequences, etc. MFBS assumes prior knowledge of the primary's signal, while WBS assumes that the received signal matches with one of the patterns previously known. This means that these sensing techniques are not feasible in some bands, where several communication technologies may operate without *a priori* knowledge. On the other hand, CBS is impracticable for signals that do not exhibit cyclostationarity properties.

Energy-based sensing (EBS) [5], [6] is the simplest spectrum sensing technique and its main advantage is related with the fact that it does not need any *a priori* knowledge of PU's signal. At the same time, it is well known that EBS can exhibit low performance in specific comparative scenarios [7], or when noise's variance is unknown or very large. EBS has been studied in several CR scenarios, namely on local and cooperative sensing schemes [1]. More recently, several EBS schemes adopting sub-Nyquist sampling have been proposed, which are advantageous in terms of the sensing duration [8].

This work evaluates the performance of the EBS technique in a real Software Defined Radio (SDR) platform. EBS' theoretical probabilities of detection and false alarm are compared with practical ones obtained with the SDR system. The comparison indicates that the theoretical probabilities are successfully validated.

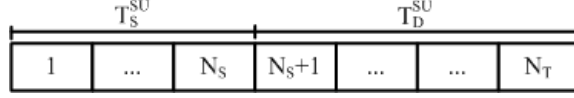
## 2 Relationship to Collective Awareness Systems

As it is well known, collective systems massively rely on communication that is most of the times supported by wireless links. Wireless communication technologies allow high level of device's mobility, and are indicated for scenarios where mobility is required. Simultaneously, wireless technologies avoid the use of physical (wired) connections, being an effective solution for collective awareness systems relying on mobile players. As mentioned in the introduction, Cognitive Radio was proposed as a solution to alleviate the increasing demand for radio spectrum and, consequently, as more radio spectrum becomes available more wireless devices can be used. Consequently, CR will help the development of novel collective awareness systems by increasing the number of wireless devices that may access the network.

## 3 Energy-based Sensing

This work considers a cognitive radio network with a pair of PUs accessing the channel and a pair of SUs that access the channel in an opportunistic way. SUs are equipped with a single radio transceiver. However, because SUs are unable to distinguish SUs and PUs' transmissions, SU's operation cycle includes the sensing and transmission periods, which facilitates the synchronization of the sensing task. Sensing and transmission period durations are represented by  $T_S^{SU}$  and  $T_D^{SU}$  respectively, as illustrated in Figure 1.





**Fig. 1.** SU's frame structure representing SU's operation cycle.

The SU's frame,  $T_F^{SU} = T_S^{SU} + T_D^{SU}$ , contains  $N_T$  slots, where each slot duration is given by the channel sampling period adopted in the spectrum sensing task. The first  $N_S$  slots define the sensing period duration, and the remaining ones ( $N_S + 1$  to  $N_T$ ) represent the transmission period duration. To distinguish between occupied and vacant spectrum bands, SUs sample the channel during the sensing period  $T_S^{SU}$ , and for each sample  $k$  two hypotheses can be distinguished

$$\begin{aligned} \mathcal{H}_{00}: x(k) &= w(k) & k &= 1, 2, \dots, N_S \\ \mathcal{H}_{11}: x(k) &= w(k) + s(k) & k &= 1, 2, \dots, N_S, \end{aligned} \quad (1)$$

where  $s(k)$  denotes the signal transmitted by the PUs, with distribution  $\mathcal{N}(\mu_s, \sigma_s^2)$ .  $w(k)$  is assumed to be a zero-mean variable with standard deviation  $\sigma_n^2 = 1$  representing additive Gaussian white noise. The received signal is given by

$$Y \sim \begin{cases} \mathcal{N}(\mu_n, \sigma_n^2), & \mathcal{H}_{00} \\ \mathcal{N}(\mu_n + \mu_s, \sigma_n^2 + \sigma_s^2), & \mathcal{H}_{11} \end{cases} \quad (2)$$

Therefore, for a single SU, the probability of detection  $P_D^{\mathcal{H}_{11}}$  and the probability of false alarm  $P_{FA}^{\mathcal{H}_{00}}$  are represented by

$$P_D^{\mathcal{H}_{11}} = Pr(Y > \gamma | \mathcal{H}_{11}) = Q\left(\frac{\gamma - (\mu_n + \mu_s)}{\sigma_n^2 + \sigma_s^2}\right) \quad (3)$$

$$P_{FA}^{\mathcal{H}_{00}} = Pr(Y > \gamma | \mathcal{H}_{00}) = Q\left(\frac{\gamma - \mu_n}{\sigma_n^2}\right), \quad (4)$$

where  $Q(\cdot)$  is the complementary distribution function of the standard Gaussian. In the testbed  $L$  time samples are used to determine the fast Fourier Transform (FFT). The FFT is represented in  $L$  frequency bins and  $N_S/L$  FFTs are computed during the sensing period ( $T_S^{SU}$ ). During this period, the average energy received per sample,  $Y_n$ , is defined as

$$Y_n = \frac{1}{N_S} \sum_{n=1}^{N_S/L} X_n, \quad (5)$$

where  $X_n$  is the sum of the power of each individual FFT bin, given by

$$X_n = \sum_{k=1}^L \left| x_n e^{-i2\pi k \frac{n}{L}} \right|^2, \quad k = 1, \dots, L. \quad (6)$$

The decisions performed in the testbed are according to the following conditions,

$$\begin{aligned} C_0 &= 0, & \mathcal{H}_{11} | Y_n < \gamma \\ C_1 &= 1, & \mathcal{H}_{11} | Y_n > \gamma \end{aligned} \quad (7)$$

$$\begin{aligned} B_0 &= 0, & \mathcal{H}_{00}|Y_n < \gamma \\ B_1 &= 1, & \mathcal{H}_{00}|Y_n > \gamma, \end{aligned}$$

where  $C_1$  contributes for the measured probability of detection, while  $B_1$  represents the case of false alarm.

## 4 System Implementation

### 4.1 Testbed

The EBS was implemented using the GR (GNURadio) [9] system and the USRP [10] (Universal Software Radio Peripheral) platform. These tools are compliant with the notion of SDR (Software Defined Radio), mostly due to their flexibility when it comes to DSP (Digital Signal Processing). This flexibility opens up options when dealing with Digital Signal Processing. In this case, a range of FFT sizes can be chosen, instead of a fixed one like in most DSP devices. This allows for a variation in the  $N_S/N_T$  ratio.

The setup consisted of two USRP B100 devices connected to one computer. The USRPs are connected by a coaxial cable and two 30dB attenuators to achieve the desired signal-to-noise (SNR) values. One USRP implements a Primary User while the other behaves as the Secondary user, sensing for the presence of the first one. The basic blocks of the GR system are written in C++. These blocks link directly with the USRP and its FPGA. Python is used as a “glue” to connect these blocks and as a controller to the system flow (issuing start and stop commands). The script used to build the EBS was entirely written in python, also making use of the language's math tools.

### 4.2 EBS Structure

A conventional EBS consists of a low pass filter, an A/D converter, a square-law device and an integrator (see Fig. 1). The detector model used in this work consists of an A/D converter followed by an L-Point FFT (6). The FFT provides the windowing needed to filter out unnecessary frequency bands. This data is then averaged (see Fig. 3), and the sensing output is determined according to  $C_0/C_1$  or  $B_0/B_1$  conditions (eq. 7).

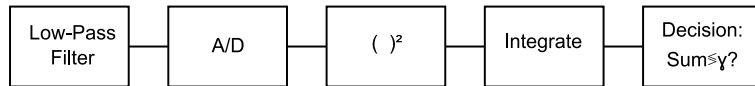


Fig. 2. Typical Energy-based Detector

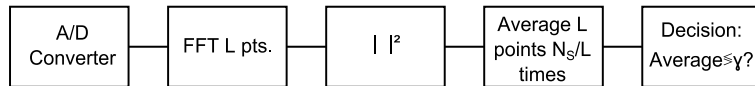


Fig. 3. FFT Energy-Based Detector

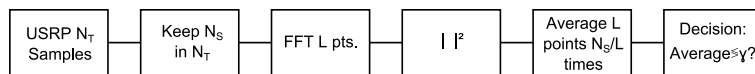


Fig. 4. GNURadio Blocks

### 4.3 Primary User Signal

The Primary signal is assumed to obey a Normal Distribution with higher amplitude than the noise distribution for sake of simplicity. The generation of its signal is achieved using the blocks in Fig.5.

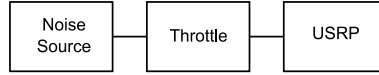


Fig. 5. PU signal generator

## 5 Comparison Results

### 5.1 Calibration and Parameter Calculation

This subsection validates the noise  $w(k)$  and PU's signal  $s(k)$ . The USRP's parameters used in the validation are shown in Table 1.

Table 1. USRP Configuration

Freq. (GHz)	L	Sample Rate (Sps/s)	SU Gain (dB)	PU Gain (db)	SNR (dB)	Primary Amplitude
1.3	256	1,000,000	0	17	0.11	0.05

Noise ( $w(k)$ ) and Signal plus Noise ( $w(k) + s(k)$ ) signals were sampled during a 10 second interval. The histogram of the received energy is plotted in Fig. 6. In this figure the green curve approximates the noise distribution, while the red one approximates the noise plus signal distribution. Since the overlapping area of the approximate distributions is negligible, a decision threshold can be defined to identify the two conditions with low error.

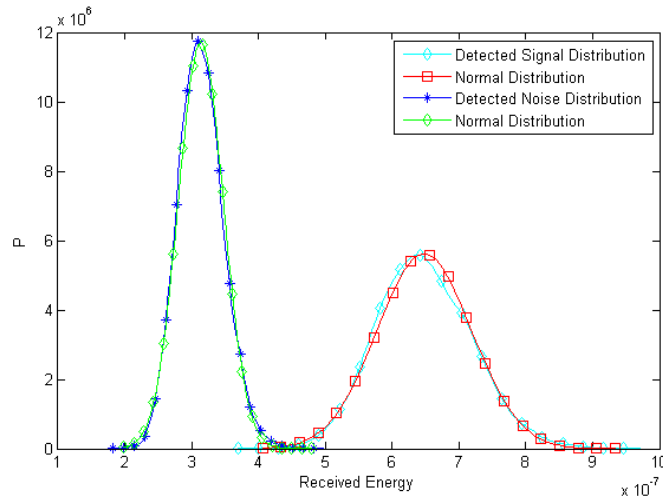
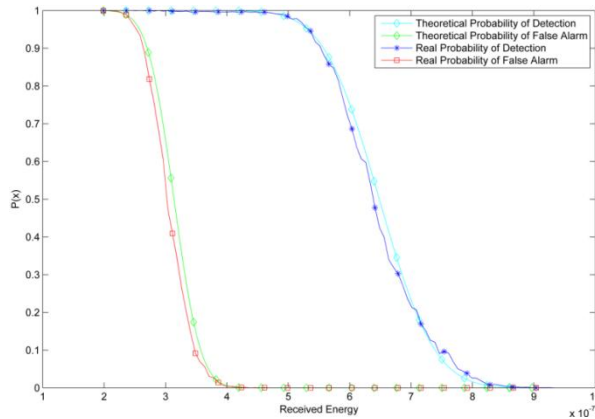


Fig. 6. Histogram of Signal and Signal plus Noise data and respective normal approximations. P is the number of occurrences.

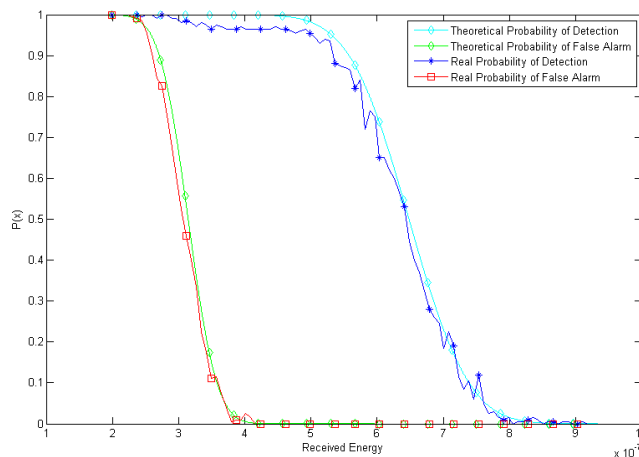
## 5.2 Practical Probabilities

Taking the x-axis values of Fig. 6 as a hypothetical range for the decision threshold, the probabilities  $P_D$  and  $P_{FA}$  were computed from the conditions expressed in equation (7). Basically,  $P_D$  was computed as the ratio of the number of times the condition  $C_1$  was observed over the total number of decisions.  $P_{FA}$  was similarly computed as the ratio of the number of times the condition  $B_1$  was observed over the total number of decisions. These probabilities are shown in Fig. 7, where the achieved results are compared with the theoretical ones.



**Fig.7.** Theoretical and practical probabilities  $P_D$  and  $P_{FA}$  ( $T_S^{SU} = 0.512s$ )

As can be seen, the registered practical values approach the theoretical ones. This is, however, for a very long sensing period (512 ms). Fig. 8 shows the results achieved for a shorter sensing period ( $N_S=512000$ , which corresponds to  $T_S^{SU} = 51.2ms$ ). In this case the practical results underperform the case of longer sensing period, but the practical results roughly follow the theoretical trend.



**Fig. 8.** Theoretical and Real Probabilities ( $T_S^{SU} = 0.0512s$ )

### 5.3 Performance Evaluation

With the knowledge of the signal and noise statistics, a threshold can be defined so that the desired conditions (a certain pair of  $P_D$  and  $P_{FA}$ ) can be achieved. By inspecting the theoretical curves for the probabilities, a threshold was defined where the  $P_{FA}$  was set to 5%. Since  $P_{FA}$  only depends on noise power, and not on noise plus signal, its value should remain constant for any SNR value.  $P_D$  does not follow the same rationale, since it depends on SNR (eq. 3). The following test evaluates the detection performance for a fixed decision threshold ( $3.7E-07$ ) and varying the SNR from -10 to 2 dB. Different sensing periods were also tested, from 0.0125s to 0.512s. The obtained results are shown in Fig 9.

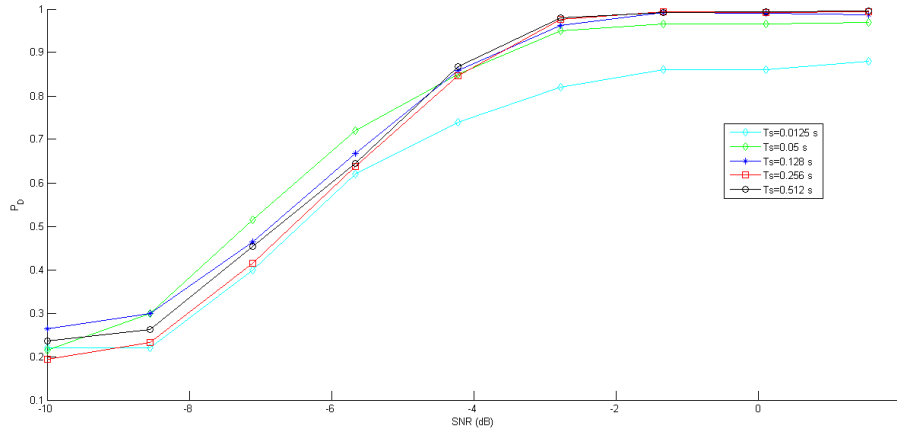


Fig. 9.  $P_D$  evolution for SNR [-10,2] dB.

As can be seen, the statistic test for sensing period durations longer than 0.05s start to be redundant, since they present almost the same performance. At 0dB we observe that the detector performance is high, and for  $T_s^{SU} > 0.05$ s the miss-detection probability is lower than 10%. Due to lack of space we do not present false alarm probabilities for the same scenario, but for sensing period durations longer than 0.05s we have observed that the false alarm probability is always lower than 5%, which confirms the expected performance for the adopted decision threshold.

## 6 Conclusions

In this work we have addressed the assessment of energy detection for cognitive radio systems. We started to characterize the performance of an energy detector through practical USRP devices. The theoretical performance was successfully validated through practical results. For higher SNR values, the detection probability is high even for short sensing periods, achieving detection probabilities that approach 1. For

lower SNR values a higher sensing time is required but even in these cases good performance can be achieved.

**Acknowledgments.** This work was partially supported by EU COST IC0902 and the Portuguese Science and Technology Foundation under the projects PTDC/EEATEL/ 115981/2009, PTDC/EEA-TEL/099074/2008, PTDC/EEA-TEL/099973/2008, PTDC/EEATEL/120666/2010, PEst-OE/EEI/UI0066/2011 and SFRH/BD/68367/2010.

## References

- [1] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *Communications Surveys Tutorials, IEEE*, vol. 11, pp. 116–130, First Quarter 2009.
- [2] Zahedi-Ghasabeh, A. Tarighat, and B. Daneshrad, "Spectrum Sensing of OFDM Waveforms Using Embedded Pilots in the Presence of Impairments," *Vehicular Technology, IEEE Transactions on*, vol. 61, pp. 1208–1221, March 2012.
- [3] A. Bouzegzi, P. Ciblat, and P. Jallon, "Matched Filter Based Algorithm for Blind Recognition of OFDM Systems," in *Proc. IEEE VTC 2008-Fall*, pp. 1–5, September 2008.
- [4] A. Al-Habashna, O. Dobre, R. Venkatesan, and D. Popescu, "Cyclostationarity-Based Detection of LTE OFDM Signals for Cognitive Radio Systems," in *Proc. IEEE GLOBECOM 2010*, pp. 1–6, December 2010.
- [5] H. Urkowitz, "Energy Detection of Unknown Deterministic Signals," *Proceedings of the IEEE*, vol. 55, pp. 523–531, April 1967.
- [6] A. Ghasemi and E. S. Sousa, "Optimization of Spectrum Sensing for Opportunistic Spectrum Access in Cognitive Radio Networks," in *Proc. IEEE CCNC 2007*, pp. 1022–1026, January 2007.
- [7] D. Bhargavi and C. Murthy, "Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing," in *Proc. IEEE SPAWC 2010*, pp. 1–5, June 2010.
- [8] Z. Tian and G. Giannakis, "Compressed Sensing for Wideband Cognitive Radios," in *Proc. IEEE ICASSP 2007*, vol. 4, pp. IV–1357–IV–1360, April 2007.
- [9] GNURadio - Free Software Toolkit for SDR - <http://www.gnuradio.org/>
- [10] Ettus Research, <http://www.ettus.com/>







