UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MAICON KIST

Adaptive Threshold Architecture for Spectrum Sensing

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Juergen Rochol
Advisor

Porto Alegre, April 2014

CIP – CATALOGING-IN-PUBLICATION

Kist, Maicon

Adaptive Threshold Architecture for Spectrum Sensing / Maicon Kist. – Porto Alegre: PPGC da UFRGS, 2014.

72 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Advisor: Juergen Rochol.

1. Spectrum Sensing. 2. Machine Learning. 3. Dynamic Spectrum Access. 4. Wireless Communication. I. Rochol, Juergen. II. Título.

“It’s a dangerous business, Frodo, going out your door.
You step onto the road, and if you don’t keep your feet, there’s no knowing where
you might be swept off to”
— GANDALF THE GREY

Arquitetura para Thresholds Adaptativos em Sensoriamento Espectral

RESUMO

A política atual de alocação do espectro compreende licenciar o uso de canais de radiofrequência e garantir que determinados usuários tenham acesso exclusivo a esses canais. Através dessa política, os melhores canais para comunicações sem fio de curtas e médias distâncias já foram licenciados. Sendo assim, é cada vez mais difícil disponibilizar canais adequados para novos serviços sem fio que necessitem destes canais ou para melhorar os serviços sem fio existentes. Contudo, medidas de utilização dos canais licenciados mostraram que estes são subutilizados em determinados períodos do dia e em certas regiões geográficas.

A atual subutilização dos canais de radiofrequência fez com que as agências governamentais considerassem um nova política de acesso ao espectro. Nessa nova política, um usuário não licenciado pode acessar canais subutilizados temporariamente, com a restrição de não causar interferência nas transmissões dos usuários licenciados. Dessa forma, antes de acessar um canal de rádio, o usuário não licenciado precisa analisa-lo com o objetivo de garantir que nenhum usuário licenciado esteja transmitindo. A análise dos canais é feita através do uso de soluções de sensoriamento espectral.

A maior limitação das soluções atuais de sensoriamento espectral é o uso de parâmetros estáticos para detectar a transmissão de um usuário licenciado. O uso de parâmetros estáticos é uma limitação porque o sensoriamento espectral pode encontrar diferentes níveis de ruído e interferência na análise dos canais. Nesse contexto, algoritmos de aprendizagem de máquina podem ser empregados para adaptar dinamicamente os parâmetros de detecção utilizados no sensoriamento espectral.

Neste trabalho é proposta a Arquitetura para Thresholds Adaptativos (ATA) para sensoriamento espectral. Essa arquitetura emprega algoritmos de aprendizagem de máquina para adaptar, em tempo real, os parâmetros de detecção do sensoriamento espectral. Além disso, um protótipo da ATA foi desenvolvido e avaliado em um ambiente de rádio experimental baseado na norma IEEE 802.22. Os resultados mostram que ATA alcança um desempenho melhor que as soluções atuais de sensoriamento espectral em termos de precisão na detecção do usuário licenciado e no tempo necessário para analisar o canal de rádio.

Palavras-chave: sensoriamento espectral, aprendizagem de máquina, acesso dinâmico ao espectro, comunicação sem fio.

Adaptive Threshold Architecture for Spectrum Sensing

ABSTRACT

The current spectrum allocation policy comprises licensing the use of channels of the radio spectrum and ensuring that licensed users have exclusive access to these channels. Through this policy, the best channels for short and long wireless communications were already allocated. Thus, it has become exceedingly hard to find vacant radio channels to either deploy new wireless services or to enhance existing ones. However, recent measurements of the radio spectrum showed that some allocated channels are rarely utilized in certain geographical areas.

The relatively low utilization of some radio channels made the governmental agencies consider a new spectrum access policy. In this new access policy, an unlicensed user can temporarily access underutilized radio channels, with the constraint of not interfering with the transmission of any licensed user. Thus, the unlicensed user must analyze the radio channel before accessing it, with the objective of guaranteeing that no licensed user is transmitting. The analysis of radio channels is made through spectrum sensing solutions.

The major drawback of current spectrum sensing solutions is the use of static parameters to detect the transmission of a licensed user. The usage of static parameters is a drawback because the spectrum sensing may encounter different noise and interference levels during the channel analysis. In this context, machine learning algorithms can be employed to dynamically adapt the detection parameters used in the spectrum sensing.

In this dissertation, we propose the Adaptive Threshold Architecture (ATA) for spectrum sensing. This architecture employs machine learning algorithms to adapt the detection parameters of the spectrum sensing in real time. Furthermore, a prototype of ATA was developed and evaluated in an experimental radio environment based in the IEEE 802.22 standard. The results show that ATA achieves a better performance than current spectrum sensing solutions in terms of the accuracy in detecting the licensed user and in the time required to analyze the radio channel.

Keywords: spectrum sensing, machine learning, dynamic spectrum access, wireless communication.

LISTA DE ABREVIATURAS

ADC	Analog-to-Digital Converter
Anatel	Agência Nacional de Telecomunições
ATA	Adaptive Threshold Architecture
CFAR	Constant False Alarm Rate
CFD	Cyclostationary Feature Detection
CDMA	Code Division Multiple Access
DAC	Digital-to-Analog Converter
DDC	Digital Down Converver
DSA	Dynamic Spectrum Access
DSP	Digital Signal Processor
DUC	Digital Up Converver
ED	Energy Detection
EBN0	Energy per bit to Noise Spectral Density Radio
FCC	Federal Communications Commision
GSM	Global System for Mobile Communications
FPGA	Field Programmable Gate Array
FFT	Fast Fourier Transform
FAM	Cyclostationary Time Smoothing Fast Fourier Transform Accumulation Method
FPGA	Field Programmable Gate Array
GMSK	Gaussian Minimum Shift Keying
ISP	Internet Service Provider
GPL	GNU Public License

GPM	General Purpose Machine
GPP	General Purpose Processor
OFDM	Orthogonal Frequency Division Multiplexing
PU	Primary User
ROC	Receiver Operating Characteristic
RF	Radio Frequency
SARSA	State-Action-Reward-State-Action
SDR	Software Defined Radio
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
SS	Spectrum Sensing
SU	Secondary User
TSHA	Two-Stage Hierarchical Architecture
WFD	Waveform Detection
USRP	Universal Software Radio Peripheral

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Current spectrum allocation policies comprise licensing the usage of channels of the radio spectrum during a long time and in large geographical areas. The allocation usually is responsibility of regulatory agencies, such as the Federal Communications Commision (FCC) in the USA and Agência Nacional de Telecomunições (Anatel) in Brazil. These agencies grant to licensed users, such as television broadcaster and mobile network operators, exclusive access to their allocated channels. On the one hand, with most of the useful radio spectrum already allocated, it became exceedingly hard to find vacant radio channels to either deploy new wireless services or enhance existing ones (GHASEMI; SOUSA, 2008). On the other hand, recent measurements showed that some allocated channels are rarely utilized (FCC, 2002).

The low utilization of some radio channels suggests that the scarcity of available channels for allocation is largely due the inefficient allocation policy rather than any physical shortage (GHASEMI; SOUSA, 2008). This observation led regulatory agencies to propose a new policy in which unlicensed radios are allowed to temporarily access underutilized radio channels, with the constraint of not interfering with the transmission of licensed users. In this new access policy, the licensed users of a radio channel are the Primary Users (PU) and unlicensed radios are the Secondary Users (SU), who seek to temporarily use a vacant channels, *i.e.* channels in which no PU is transmitting. The SUs that operate under this policy are said to perform dynamic spectrum access. Since the SU is not allowed to interfere with the PU transmissions, it must analyze the radio channel to evaluate its occupancy status, *i.e.* the channel is said to be "occupied" if a PU is transmitting, otherwise the channel is said to be "vacant" (YUCEK; ARSLAN, 2009). This analysis is performed by the Spectrum Sensing (SS).

The fundamental task of the SS is to decide whether or not a particular radio channel is being occupied by a PU. To do this, the SS performs three operations: (*I*) analyzes the signal received in a radio channel for a period of time. (*II*) summarizes the signal in a single value, and (*III*) compares the value obtained against a decision threshold. The decision threshold is a numerical value, preconfigured in the sensing technique, used to determine the channel occupancy status. More precisely, the channel is considered vacant only when the decision threshold is greater than the single value generated in operation (*II*). Based on the aforementioned operations, we can define the two main metrics of the SS: sensing duration and sensing accuracy. The first is the time required to analyze the radio signal and summarize it in a single value. The second is the percentage of correct decisions when compared to the real channel occupancy status. This metric is highly related with the value defined for the decision threshold.

Different sensing techniques were proposed in the literature to analyze and estimate the sensed channel occupancy status, such as Energy Detection (ED), Waveform De-

tection (WFD) and Cyclostationary Feature Detection (CFD). These techniques present different trade-offs regarding the sensing accuracy and sensing duration. For example, the ED is the least accurate technique and the fastest one, while the CFD is the most accurate and the slowest one. Overcoming the trade-off between sensing accuracy and duration has proved to be one of the main challenges in developing new sensing techniques (AXELL et al., 2012).

Research efforts in SS led to the proposal of the cooperative and multi-stage approaches to improve the performance regarding the sensing accuracy and sensing duration. In the cooperative approach, multiple SUs send their decisions to a common receiver through a reporting radio channel. Although this approach might increase the sensing accuracy in certain radio environments, some authors showed that the overhead required to exchange information among SUs and the need for a dedicated reporting channel can make this approach infeasible in some scenarios (PEH et al., 2009), (ATAPATTU; TELLAMBURA; JIANG, 2011), (AKYILDIZ; LO; BALAKRISHNAN, 2011). The multi-stage approach was proposed in the literature to improve the sensing of an individual SU without the need of cooperation. This approach combines SS techniques in a hierarchical structure, maintaining a high sensing accuracy and a low sensing duration (LUO et al., 2009), (MALEKI; PANDHARIPANDE; LEUS, 2010), (NAIR; VINOD; KRISHNA, 2011), (EJAZ; HASAN; KIM, 2012).

The major drawback of cooperative and multi-stage approaches is the use of static decision thresholds to determine if a radio channel is vacant or occupied. This is a drawback because the SU may encounter different noise or interference levels when switching between different channels, requiring different values for the decision threshold. Moreover, SUs may be operating in unknown radio environments and may not have perfect knowledge of the characteristics of the other existing PUs or SUs (BKASSINY; LI; JAYAWEERA, 2013). Hence, the decision threshold of SUs needs to be adapted accordingly to each particular situation. Machine learning algorithms have been highlighted as a solution to properly adapt the decision threshold in such radio environments (BKASSINY; LI; JAYAWEERA, 2013).

To the best of our knowledge, no solution considering the use of machine learning to adapt the decision threshold has been proposed so far. In this sense, we propose the Adaptive Threshold Architecture (ATA), a solution that allies machine learning and multi-stage SS to overcome the identified constraints. ATA improves the current state-of-the-art solutions by enabling the unlicensed user to learn and adapt the decision threshold in real time, achieving high sensing accuracy in different radio environments. In addition, ATA utilizes a mechanism that reduces the sensing duration when the sensing accuracy is high. The proposed architecture comprises two components: *Sensing Component* and *Machine Learning Component*. The first component executes any two independent SS techniques, named *First Stage* and Second Stage. The second component executes the *Feedback Algorithm*, which reduces the sensing duration, and the *Threshold Learning Algorithm*, which dynamically adapts the decision threshold.

A prototype of ATA was developed using the GNU Radio toolkit (GNU Radio, 2004) and the Universal Software Radio Peripheral (USRP) front-end (Ettus Research, 2008). The prototype was evaluated considering different combinations of sensing techniques in an experimental radio environment emulating the behaviour of public safety radio channels. Gathered results showed that the proposed architecture increased the sensing accuracy and reduced the sensing duration when compared to other multi-stage architectures. Therefore, the main contributions of this dissertation are:

1. An architecture that integrates machine learning to SS solutions. This enables the SU to learn and adapt the decision threshold in real time, achieving high sensing accuracy in unknown radio environments.

2. A mechanism to reduce the sensing duration when the sensing accuracy is high. This enables the SU to provide fast and accurate decisions in radio environments where the decision threshold is easy to learn.

3. Development of a prototype in a real radio device. With this we were able to verify the applicability of ATA in current radio platforms. In addition, the prototype enables the analysis of other aspects of the sensing, such as processing and energy consumption.

The remainder of this dissertation is organized as follows. Chapter 2 details the dynamic spectrum access and the fundamentals of signal detection by SS. In Chapter 3 we show other research efforts that aim to improve the sensing accuracy and sensing duration of the SS. Chapter 4 details every aspect of the proposed architecture. The hardware and software, as well as the prototype used to evaluate the proposed architecture are presented in Chapter 5. Chapter 6 presents and discusses the evaluation methodology and the results obtained. Final remarks and future work are presented in Chapter 7.

2 BACKGROUND

Through this chapter we discuss the fundamental concepts to correctly understand this dissertation. Although other references to other articles are made, the text of this chapter is a compendium of the articles of Ghasemi and Souza (GHASEMI; SOUSA, 2008), Yucek and Arslan (YUCEK; ARSLAN, 2009), and Axell *et al.* (AXELL et al., 2012). The underlying concepts of Dynamic Spectrum Access (DSA) and the regulatory policies to prevent interference in PUs are presented in Section 2.1. The Software Defined Radio (SDR) is described as a technology that enabled the design and evaluation of Spectrum Sensing (SS) techniques in Section 2.2. The fundamental concepts for Spectrum Sensing (SS) and the main techniques developed to detect the PU signal are presented in Section 2.3.

2.1 Dynamic Spectrum Access

Current spectrum allocation policy comprises licensing the usage of channels of the radio spectrum during a long time and in large geographical areas. The allocation usually is responsibility of regulatory agencies. The regulatory agencies grant the licensed holders, such as television broadcaster and mobile network operators, exclusive access to their allocated channels. On the one hand, with most of the useful spectrum already allocated, it became exceedingly hard to find vacant radio channels to either deploy new wireless services or enhance existing ones (GHASEMI; SOUSA, 2008). On the other hand, recent measurements showed that the allocated channels are rarely utilized, as illustrated in Figure 2.1 (FCC, 2002).

The relatively low utilization of some radio channels suggests that the scarcity of available channels for allocation is largely due the inefficient allocation policy rather than any physical shortage (GHASEMI; SOUSA, 2008). This observation led regulatory agencies to propose a new policy in which unlicensed users are allowed to opportunistically access underutilized channels, with the constraint of not interfering with the signal of any licensed user. In this policy, the PUs are the licensed users of a radio channel and the SUs are unlicensed users, who seek to temporarily use vacant radio channels. One major milestone towards this access policy is the IEEE 802.22 Standard (STEVENSON et al., 2009), in which a base station acts as a SU, accessing vacant television channels to provide Internet access in rural areas. The SUs that operate under this policy are said to perform dynamic spectrum access.

A device performing dynamic spectrum access increases the overall usage of the radio spectrum by taking advantage of underutilized channels, commonly referred to as “white spaces”. The conventional definition of white space is “a band of frequencies that are not being used by the PU of that band at a particular time in a particular geographic area”

(KOLODZY et al., 2001). However, this definition only exploits three dimensions of the spectrum: frequency, time, and space. Yucek and Arslan (YUCEK; ARSLAN, 2009) added two more dimensions to this definition: code and angle.

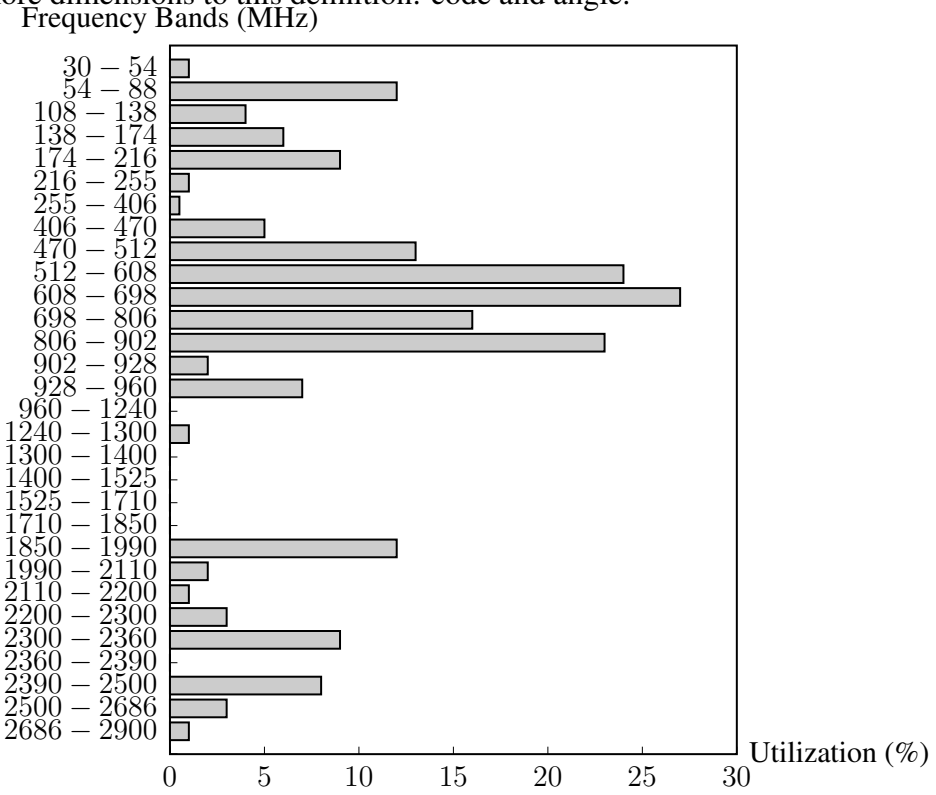
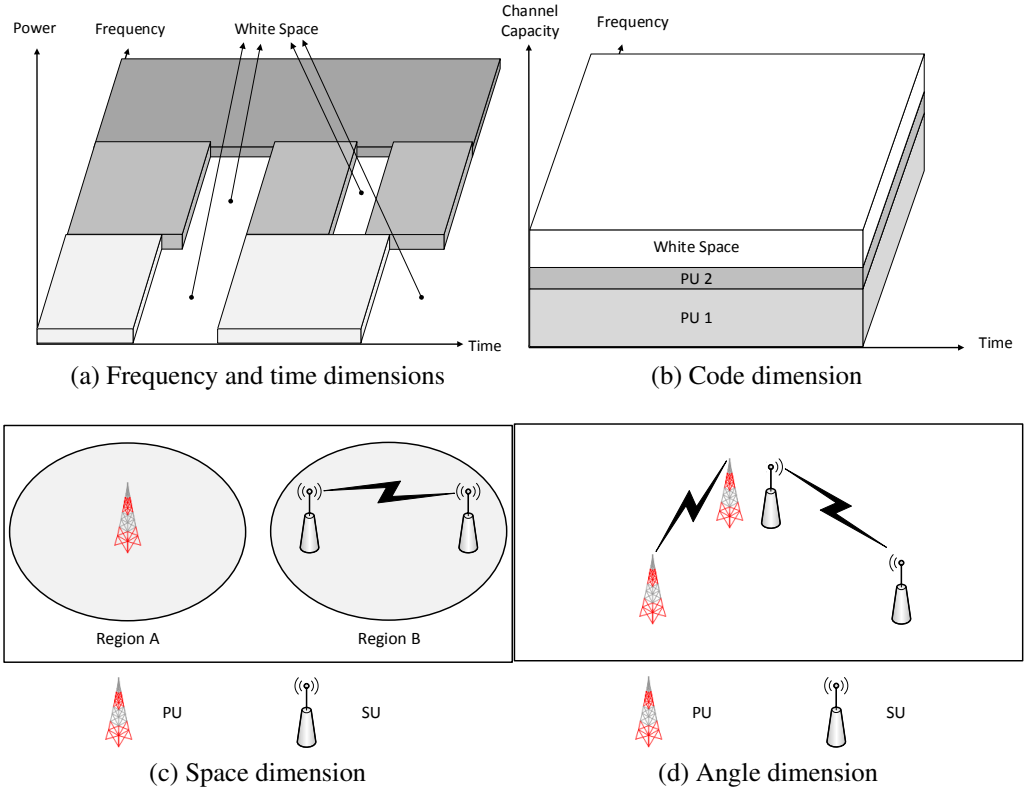


Figure 2.1: Average radio spectrum usage in Atlanta, New Orleans, and San Diego

White spaces in frequency and time dimensions arise when a licensed channel of the spectrum becomes temporarily unused. Consequently, a SU can temporarily access this underutilized channel. Figure 2.2a illustrates the occurrence of white spaces in frequency and time dimensions, considering three different channels, *e.g.,* TV channels 1, 2 and 3. It is worth noticing that the duration of the occupied and vacant periods depends on several characteristics, *e.g.,* television channel or mobile network channel, night or day period, rural or urban areas. For example, in a television channel, the time scale of the occupied and vacant periods is expected to be of months, while in mobile networks this period is in the order of milliseconds.

Modern wireless systems can multiplex several transmitters in a single radio channel through the use of orthogonal codes. An orthogonal code is a exclusive sequence of bits assigned to an user. For example, an orthogonal code is given to each mobile device in mobile systems using Code Division Multiple Access (CDMA). Multiple mobile devices can access the same channel simultaneously, each one using its exclusive orthogonal code to modulate the transmitted data. Thus, the total number of available orthogonal codes represents the channel capacity, *i.e.* the maximum number of users that can access the same radio channel simultaneously. White spaces in the code dimension arise when the number of licensed users is lower than the total number of orthogonal codes. SUs can explore unused orthogonal codes to access the channel simultaneously with PUs. Figure 2.2b depicts the occurrence of white spaces in this dimension.

In some geographical areas several channels of the radio spectrum present high uti-



(a) Frequency and time dimensions

(b) Code dimension

(c) Space dimension

(d) Angle dimension

Figure 2.2: White spaces dimensions explored in the dynamic spectrum access

lization by PUs. For example, in metropolitan areas it is common that all TV channels are occupied. Any SU in such geographical area will hardly find a white space in time, frequency, and code domains. However, it is very common that the signal of the TV broadcaster does not reach remote rural areas. A SU located in such areas can be benefited from the spatial dimension. White spaces in the spatial dimension arise when a licensed channel is never used in specific geographical areas (YUCEK; ARSLAN, 2009), as illustrated in Figure 2.2c.

Modern wireless antennas are capable to focus the transmitted radio signal on specific directions, forming the so called "beams". White spaces in the angle dimension arise when the PU transmits its radio to a receiver using a signal beam. A SU located in the same geographical as the PU can use the same channel without causing interference if their beams are focused on different directions, as depicted in Figure 2.2d.

A SU accessing the spectrum dynamically must explore these five dimensions to maximize the access opportunities. We highlight some dimensions are clearly more simple to explore than others. For example, any secondary user performing SS will naturally be benefited from the spatial dimension. Whereas the code characteristic is the most complex, probably requiring that the secondary user negotiate the use of a orthogonal code with a licensed user. The operation of a SU can interfere with a PU when errors in the spectrum sensing occur. Thus, regulatory agencies, aiming to protect the PU, impose some constraints for the spectrum sensing.

2.1.1 Regulatory Constraints for Spectrum Sensing

Governmental agencies guarantee that PUs have exclusive access to their radio channels, whereas the SUs can access white spaces in the radio spectrum to opportunistically

perform communications. In this scenario, an important aspect is to guarantee that the operation of PUs is not damaged by harmful interferences (GHASEMI; SOUSA, 2008). To protect the PU from harmful interferences that can be caused by the operation of SU, sensing techniques must comply with two constraints imposed by governmental agencies: sensing sensibility and sensing periodicity.

Sensing Sensibility

The interference caused by a SU is considered harmful if it causes the Signal-to-Interference Ratio (SIR) of the PU to fall below a threshold value Γ, supplied by the governmental agencies. This threshold depends on the PU robustness towards interference and varies according to the channel used, characteristics of the transmitted signal and interference type, *e.g.,* continuous or intermittent interference (GHASEMI; SOUSA, 2008).

The maximum interference that a SU can cause (Γ) in a PU without causing harmful interference depends on four variables: (*I*) the PU transmission power (P_{PU}), (*II*) the SU transmission power (P_{SU}), (*III*) the maximum distance between the transmitter PU and a receiver PU (R), which depends on the type of network considered, and (*IV*) the interference range (D), which is the maximum distance that any PU can be from a transmitter SU so that the interference is still considered harmful. The distance R is the distance from the PU transmitter to the furthest PU receiver. It is noteworthy that R is less than or equal to the coverage radius of the PU. Figure 2.3 illustrates the distance R, D and the PU coverage radius.

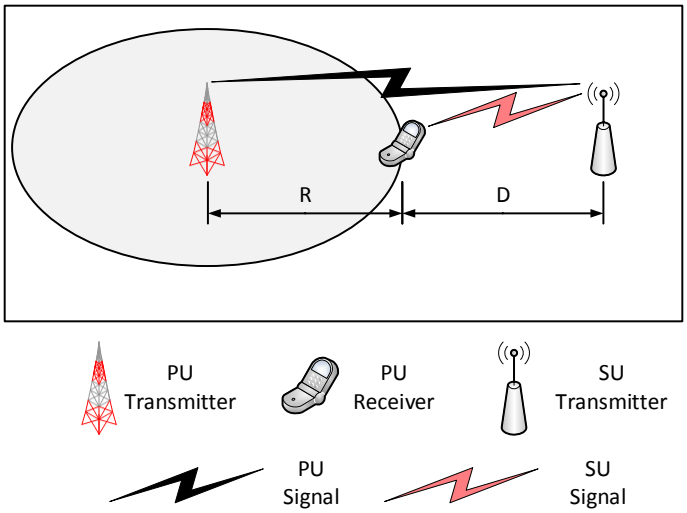


Figure 2.3: Interference range of a SU transmitter

The relation between P_{PU}, P_{SU}, D and R is presented in Equation 2.1 (GHASEMI; SOUSA, 2008). In the equation P_{BI} is the background interference power at the PU receiver and $L(\cdot)$ denotes the total path loss at a given distance. Since the path loss depends on the frequency band, geographical characteristics and antenna heights, these parameters indirectly affect the equation.

$$\Gamma = \frac{P_{PU}L(R)}{P_{SU}L(D) + P_{BI}} \tag{2.1}$$

Equation 2.1 ensures that a PU receiver is protected from harmful interferences even if it is located at the edge of the PU transmitter coverage radius. A consequence of this

equation is that regulatory agencies require that the SU detects any PU transmitter in the radius of $R+D$. To guarantee that this requirement is fulfilled, the SU sensing sensibility Γ_{min} is calculated as a function of the PU transmission power P_{PU}, the total path loss function $L(D+R)$ and the noise power P_{NP}.

$$\gamma_{min} = \frac{P_{PU} L(D+R)}{P_{NP}} \tag{2.2}$$

In practice, regulatory agencies must take into account the type of PU and SU signal to determine the values for $Gamma$, P_{PU} and R. Based on these values, the minimum sensing sensibility (Γ_{min}) is calculated and SUs must adjust the total sensing duration to comply with the defined value.

Sensing Periodicity

Sensing and transmission operations cannot be performed simultaneously in the same white space because the SU can detect its own signal. In such a case, the SU might evaluate the channel as occupied when in fact it is vacant. The most common solution to work around this problem is to interleave these two operations. Therefore, the SU should transmit in the frequency band for T_{TX} seconds and sense for T_{SS} seconds. Thus, the sensing periodicity ($T_P = T_{TX} + T_{SS}$) is the maximum interval between two consecutive sensing periods. The value for T_P must be defined by regulatory agencies, which must take into account the type of service provided by the PU, *e.g.,* the sensing periodicity is expected to be in the order of milliseconds for public safety channels and of seconds for television channels (GHASEMI; SOUSA, 2008). The relation between the sensing interval T_{SS}, the transmission interval T_{TX} and the sensing periodicity T_P is illustrated in Figure 2.4. We note that the sensing interval T_{SS} is different from the sensing duration. More precisely, several channels must be sensed during a sensing interval T_{SS}, whereas the sensing duration is the time required to sense a single channel.

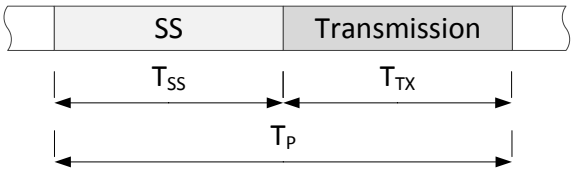
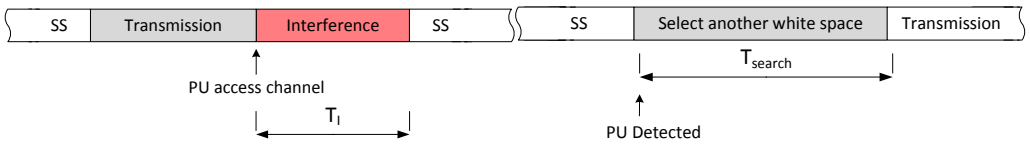


Figure 2.4: Sensing periodicity

The time interval during which the SU is harmfully interfering with the PU transmission is defined as interference interval (T_I). In the worst case, T_I is equal to the transmission period ($T_I = T_{TX}$) (GHASEMI; SOUSA, 2008). Figure 2.5a illustrates the interference interval T_I. In addition, when a PU signal is detected, the SU must search another white space to use. During the searching interval (T_{search}) the SU performs SS until a vacant channel is found, as shown in Figure 2.5b. Moreover, $T_{search} = \frac{1}{p_{vacant}} T_{SS}$, where p_{vacant} is the probability of finding a vacant channel. Ideally, the searching interval should be as fast as possible to maximize the overall throughput of the SU.

The throughput (R) of a radio that only transmits can be calculated by $R = T_{TX} r$, where r is the average data rate achieved in the radio channel (GHASEMI; SOUSA, 2008). However, a SU not only transmits but also senses the channel. In this case, we must consider the sensing interval T_{SS} in the calculation of R, as showed in Equation 2.3.



(a) Harmfull interference caused by a SU (b) SU operation when a PU is detected

Figure 2.5: Interference interval and search interval

Analysing the equation, we can note that is desirable to maintain $T_{SS} \lll T_{TX}$ in order to maximize the time available for data transmissions and, hence, the throughput.

$$R = \frac{T_{TX}}{T_{TX} + T_{SS}} r \tag{2.3}$$

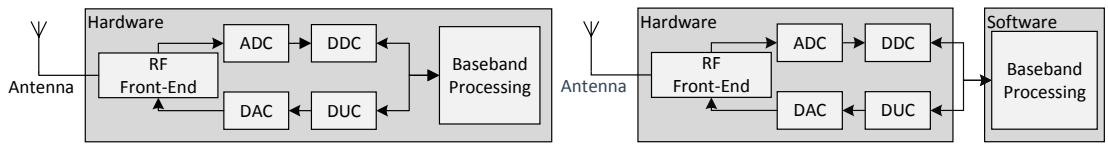
The definitions of regulatory agencies regarding sensing sensibility and periodicity, and the sensing technique play an important role in the SU operation. In addition, as regulatory constraints are applied to all SUs, the sensing technique becomes a significant component to differentiate equipments that access the radio spectrum dynamically.

To detect white spaces, the SU must be able to reconfigure its Radio Frequency (RF) parameters, such as central frequency, bandwidth, and modulation. The flexibility required for a SU to adapt the RF parameters is achieved through the use of SDR, detailed in the next section.

2.2 Software Defined Radio

In an article published in 1993, Mitola (MITOLA, 1993) envisioned a radio which could be completely reconfigured just by changing the software running on it. Being software based, Mitola naturally baptised his radio as SDR. SDR refers to technologies where the baseband processing is performed by software modules running either on Field Programmable Gate Array (FPGA), Digital Signal Processor (DSP), General Purpose Processor (GPP), or a combination thereof (MITOLA, 1993).

The use of software modules enables the programability of the baseband processing. As a consequence, the operations executed in the radio, such as coding algorithm, modulation type, and frequency band, can be easily changed, simply by loading a new module. In addition, multiple radio devices with different characteristics can be replaced by a single SDR. The architectures of conventional digital radios and SDRs are compared in Figure 2.6.



(a) Conventional digital radio architecture (b) SDR architecture

Figure 2.6: Comparison of conventional and SDR architectures

The components presented in Figure 2.6 are: (I) the antenna, which transmits and receives the information encoded in radio waves, (II) the RF front-end, which is respon-

sible for translating the radio signals to an intermediate frequency, (III) the Analog-to-Digital Converter (ADC)/Digital-to-Analog Converter (DAC), which are responsible for converting the encoded data between analog and digital domains, (IV) the Digital Up Converver (DUC)/Digital Down Converver (DDC), which are responsible for translating the intermediate digitized signal to baseband frequency, and (V) the baseband processing, which is responsible for performing additional operations over the digitized signal, such as equalization, frequency hopping, correlation, coding, and decoding algorithms.

The transition from the conventional to the SDR architecture results in a substantial increase in power consumption. The higher energy consumption is one of the key reasons why SDR has not been deployed in end-user devices. However, base stations and access points, which usually are connected to external power sources, are being replaced by SDRs. In addition, the received radio signal must be sampled at least at a rate determined by the Nyquist frequency (ROCHOL, 2012). This means that high-frequency signals require very high sampling rates and, consequently, components with high performance to execute the baseband processing in software.

SDR is currently being used to build radios that support multiple interface technologies, such as CDMA and Global System for Mobile Communications (GSM). In addition, new technology advances made possible the development of cost effective SDR platforms. The flexibility offered by SDRs enables radios to switch functions and operations on demand. This flexibility enables the research, design, development and evaluation of different sensing techniques. The following section presents the background on SS, the most important function to detect white spaces in the radio spectrum (YUCEK; ARSLAN, 2009).

2.3 Fundamentals of Spectrum Sensing

The main task of the spectrum sensing is to decide if a particular radio channel is being occupied by a PU or if it is vacant. This decision is usually expressed as a discrimination between two hypotheses. The channel is considered vacant under the hypothesis $\mathcal{H}_0$ and occupied under $\mathcal{H}_1$:

$$\mathcal{H}_0 \text{ (vacant) vs. } \mathcal{H}_1 \text{ (occupied)} \tag{2.4}$$

The received signal is essentially the ambient noise when the radio channel is vacant, whereas is it the sum of the PU signal and the ambient noise when the channel is occupied. Equation 2.5 shows the expected signal received in each of both hypotheses (AXELL et al., 2012), where y is the signal received during the SS, w is the sum of all noises and interferences, x is the PU signal and, n is a signal sample, and N the total of samples considered.

$$\begin{aligned} \mathcal{H}_0 &: y[n] = w[n], & n = 1, \ldots, N \\ \mathcal{H}_1 &: y[n] = w[n] + x[n], & n = 1, \ldots, N \end{aligned} \tag{2.5}$$

Based on Equation 2.5, we can reformulate the task of the spectrum sensing as "deciding if a given signal y, composed of N complex samples, contains only the noise w or if it is the sum of signal x and noise and interferences w". This task is accomplished by applying a test statistic Λ in the received signal y and comparing its result against a predefined decision threshold τ, as showed in Equation 2.6 (AXELL et al., 2012). Although

the SS can involve some additional operations, we can somewhat say that the SS and the test statistic Λ.

$$\Lambda(y) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau \tag{2.6}$$

Deciding if the PU signal is present in the received signal y is subject to errors due to modifications caused by the sum of noises and interferences w, limited number of samples N, and the inherent randomness of the observed data. Thus, by analyzing Equation 2.6, it is possible to identify four different outcomes regarding the SS decision:

White space detection (P_{WSD}): the SS correctly detects a white space, *i.e.* no PU is transmitting in the analyzed frequency band. This outcome is associated with the following conditional probability:

$$P_{WSD} = Pr(\mathcal{H}_0|\mathcal{H}_0) \tag{2.7}$$

False alarm (P_{FA}): the SS detects a PU signal when only noise is present. A false alarm may lead to a potentially wasted opportunity for the SU to transmit. This outcome is defined by:

$$P_{FA} = Pr(\mathcal{H}_1|\mathcal{H}_0) = 1 - P_{WSD} \tag{2.8}$$

Miss detection (P_{MD}): when the PU is transmitting, the SS decides that only noise is present. A missed detection could potentially lead to a collision with the PU, leading to wasted transmissions for both PU and SU. This outcome is associated with the following conditional probability:

$$P_{MD} = Pr(\mathcal{H}_0|\mathcal{H}_1) \tag{2.9}$$

PU detection (P_D): the SS correctly detects a transmitting PU, *i.e.* a PU is transmitting in the analyzed frequency band. This outcome is defined by:

$$P_D = Pr(\mathcal{H}_1|\mathcal{H}_1) = 1 - P_{MD} \tag{2.10}$$

The main challenge in designing a SS technique is to determine a test statistic Λ and a decision threshold τ that maximizes the protection of the PU, which is given by P_D, and the transmission of the SU, which is given by P_{WSD}. However, regulatory agencies fix a maximum P_{FA} that must be met by the SS technique. Consequently, Receiver Operating Characteristic (ROC) curves, which is a plot of the P_{FA} versus the achieved P_D, became a popular form to evaluate the SS performance. Figure 2.7 illustrates a typical ROC curve. However, ROC curves do not easily represent the sensing accuracy, calculated as shown in Equation 2.11.

$$Acc = \frac{P_D + P_{WSD}}{P_{FA} + P_{MD}} \tag{2.11}$$

Sensing techniques consider different properties of the radio signal to implement the test statistic Λ. In the following subsections we present the techniques used in this dissertation to conduct the experimental evaluation: ED, WFD, and CFD.

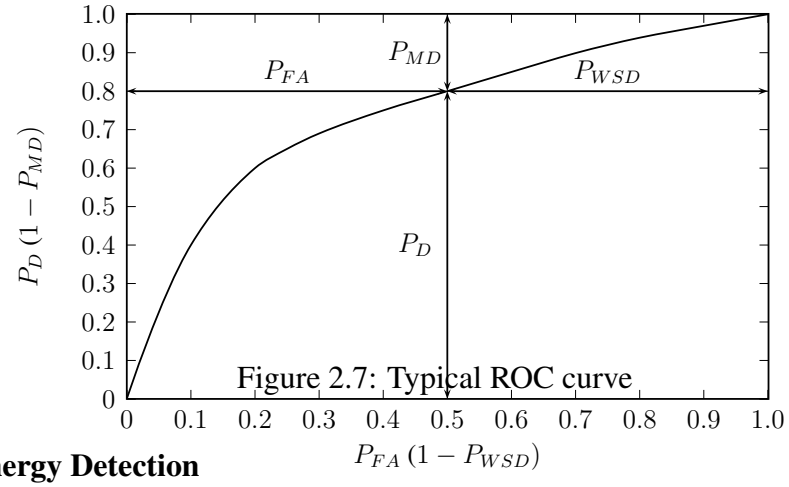


Figure 2.7: Typical ROC curve

2.3.1 Energy Detection

ED is the most efficient technique when the parameters of the PU signal are unknown (SAHAI; HOVEN; TANDRA, 2004). This efficiency, allied with the combination of low computational complexity and ease of implementation, made the ED widely adopted. The most common test statistic for the ED is showed in Equation 2.12 (AXELL et al., 2012).

$$\Lambda(y) = \frac{\sum_{i=0}^{N} y(i)^2}{N} \underset{H_0}{\overset{H_1}{\gtrless}} \tau \tag{2.12}$$

Equation 2.12 calculates the average energy of the received signal y and compares the resulting value against the decision threshold τ. A PU is transmitting in the radio channel when the average energy is greater than τ. Thus, the definition of a near optimal value for τ is essential. The probability of detection P_D of this detector is well known in the literature and is expressed in Equation 2.13.

$$P_D = P(\Lambda(y) > \tau|\mathcal{H}_1) = 1 - F_{\chi^2_{2N}}\left(\frac{F^{-1}_{\chi^2_{2N}}(1-P_{FA})}{1+\frac{P_{PU}}{P_{NP}}}\right) \tag{2.13}$$

where $F_{\chi^2_N}(\cdot)$ is the chi-square distribution. Clearly, the probability of detection P_D is a function of P_{FA} and the Signal-to-Noise Ratio (SNR) P_{PU}/P_{NP}[1]. It is worth noticing that for a fixed P_{FA}, $P_D \to 1$ as $N \to \infty$ at any SNR. This means that any pair of values (P_D, P_{FA}) can be achieved if the sensing duration could be arbitrarily long. But in practice the sensing must be done in short periods, typically in order of milliseconds.

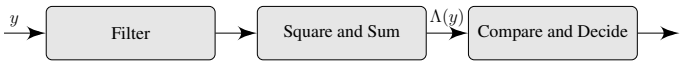


Figure 2.8: Block diagram of an ED

A block diagram of the ED implementation is shown in Figure 2.8. The block *Filter* discards all frequencies outside the range of interest. Then, the resulting signal is digitized in the ADC. Afterwards, the average power of the received signal y is calculated in the *Square and Sum* block. In the *Compare and Decide* block, the resulting energy is compared against the decision threshold τ and the final hypothesis is provided to other modules of the dynamic spectrum access, *i.e.* a module that will select the best vacant channel in terms of characteristics such as bandwidth, SNR, central frequency, among others (CABRIC; TKACHENKO; BRODERSEN, 2006).

[1]We remember that P_{PU} is the PU transmitting power and P_{NP} is the noise power.

2.3.2 Waveform Detection

Wireless communications usually involve the transmission of signal patterns, with the purpose of synchronizing transmitters and receivers. These patterns included preambles, midambles, pilot patterns, and spreading codes. If known *a priori*, a SU can utilize these patterns to detect the presence of a PU (YUCEK; ARSLAN, 2009). Equation 2.6 is rewritten for the WFD as showed in Equation 2.14, where p is a known pattern and N_p is the number of samples it contains. In Equation 2.14, the received signal y is correlated with the conjugate of p and the resulting value is compared against the decision threshold τ.

$$\Lambda(y) = Re\left[\sum_{i=0}^{N_p} y(i)p^*(i)\right] \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau \tag{2.14}$$

The channel is considered occupied by a PU when the correlation of y and p is higher than the decision threshold τ. The probability of detection P_D for the WFD is expressed in Equation 2.15.

$$P_D = P(\Lambda(y) > \tau|\mathcal{H}_1) = \sqrt{N_p \frac{P_{PU}}{P_{NP}}} \tag{2.15}$$

In the WFD, a linear reduction in the SNR demands a linear increase in the total number of samples N_p to maintain a fixed P_D, whereas the ED requires a quadratic increase. However, the performance of the WFD is degraded in the presence of frequency and timing offsets, and fading.

The block diagram in Figure 2.9 shows the implementation of a WFD (CABRIC; TKACHENKO; BRODERSEN, 2006). The *Correlate* block correlates the received signal y and a known signal pattern p, which is stored in a *Pattern Database*. In practice, the *Pattern Database* must store a pattern for each type of PU signal that must be identified and each of these patterns must be correlated with the received signal. Therefore, if the number of signal patterns and the number of samples in each pattern is high, the usage of this detector is compromised due to the high computational power required to provide the final hypothesis in an acceptable time interval. The WFD can also be performed in the frequency domain, with the use of a Fast Fourier Transform (FFT) before the correlation block. As a consequence, the number of patterns is reduced and problems with frequency and time offsets are almost eliminated.

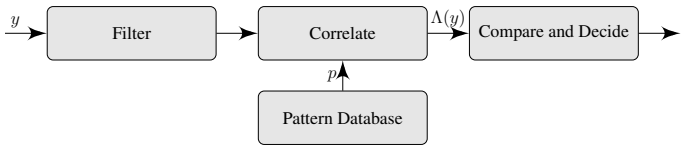


Figure 2.9: Block diagram of a WFD

2.3.3 Cyclostationary Feature Detection

A signal is said to be cyclostationary when its second order statistic parameters, *i.e.* energy and autocorrelation, are periodic in time (YUCEK; ARSLAN, 2009). Digitally

modulated signals present such periodicities due to the data rate, carrier frequency, location of guard bands, symbol rate and cyclic prefixes. The CFD makes use of these periodicities to detect the presence of PUs (YUCEK; ARSLAN, 2009). The device executing the CFD is able to detect even weak signals but at the cost of high computational power and a considerably long sensing duration when compared to other sensing techniques. The test statistic for the CFD is showed in Equation 2.16, where γ is a time lag and α is a cyclic frequency.

$$\Lambda(y) = \frac{1}{N}\sum_{i=0}^{N} y(n)y^*(n+\gamma)e^{-j2\pi\alpha n} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau \tag{2.16}$$

The meaning of this equation is to calculate the cyclic correlation of the received signal with a frequency f and delay γ. A PU signal is considered present when the resulting cyclic correlation is greater than the decision threshold τ. As the cyclic correlation must be calculated for each pair of γ and α, the total number of cyclic frequencies I and time lags P must be carefully selected in a way that accurate sensing evaluations are possible while maintaining a low sensing duration (AXELL et al., 2012). In practice, a good sensing accuracy and sensing duration is obtained using only a few cyclic frequencies and time lags.

Figure 2.10 shows the block diagram of a widely known implementation of the CFD, known as Cyclostationary Time Smoothing Fast Fourier Transform Accumulation Method (FAM) (CABRIC; TKACHENKO; BRODERSEN, 2006). The *FFT (N points)* block transforms the signal from the time to the frequency domain. The resulting N complex numbers are shifted in frequency by multiplying the real and complex parts for $e^{-2\pi kI/N}$, where k is the index of the sample in the FFT, and then correlated before going into the *FFT (M points)*. Afterwards, the cyclostationarity is calculated by squaring and summing the complex output of the second FFT in the *Square and Sum* block. Finally, in the *Compare and Decide* block, the cyclostationarity must be compared with a predefined decision threshold τ.

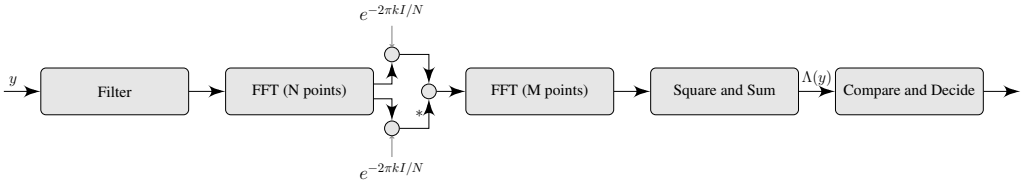


Figure 2.10: Block diagram of the CFD

2.3.4 Defining the decision threshold τ

The definition of the decision threshold τ plays a major role in guaranteeing the sensing accuracy of a given sensing technique. The most adopted method to define a value for τ is the Constant False Alarm Rate (CFAR) (AXELL et al., 2012). The widely adoption of CFAR is due to its simplicity and efficiency in maintaining a fixed probability of false alarm P_{FA}. Using CFAR, the value of τ is obtained as follows:

1. The SU device is positioned in the physical environment in which it is going operate. Also, it must be ensured that no PU is transmitting in the environment, *i.e.* only noise is present in the captured radio signal;

2. The sensing technique of the SU is activated and the values of the test statistic Λ are collected. We highlight the value collected is the real/complex value generated by the sensing technique and not the decision hypothesis. More precisely, the value gathered is the one entering the *Compare and Decide* block of the diagrams presented in Figures 2.8, 2.9, and 2.10;

3. Lets define as V the list of the values collected sorted in ascending order. The decision threshold τ assumes the value given in Equation 2.17. The meaning of the equation is selecting the element at index $|V|(1 - P_{FA})$ as the decision threshold.

$$\tau = V\left[|V|(1 - P_{FA})\right] \tag{2.17}$$

2.3.5 Comparison of the Sensing Techniques

We define four characteristics that must be considered in a sensing technique: (I) complexity, which specifies the computational power required and difficulty in implementing the technique, (II) sensibility, which is the capacity to distinguish a PU signal from noise, (III) robustness, which determines the capacity of the sensing technique to maintain a constant P_D and P_{FA} under different interferences conditions, and (IV) duration, which specifies the sensing duration of the technique. A basic comparison of the aforementioned sensing techniques considering these characteristics can be seen in Table 2.1.

Sensing Technique	Complexity	Sensibility	Robustness	Duration
ED	Low	Low	Low	Low
WFD	Medium	High	High	Medium
CFD	High	High	High	High

Table 2.1: Comparison of sensing techniques

The ED is the simplest technique, being the easiest to implement and requiring low computational power. The sensibility and robustness are reduced because the ED does not use any information from the PU signal to perform detection, thus difficulting the distinction of a PU signal from noise. Due to its simplicity, this technique presents the lower sensing duration.

The WFD is considered an intermediate technique in terms of complexity. When the number of patterns to be correlated increases, the complexity increases in the same factor. This detector can easily distinguish PU signals from noise and from other PU signals that are using different waveforms, *e.g.,* distinguish signals using Gaussian Minimum Shift Keying (GMSK) from signals using Orthogonal Frequency Division Multiplexing (OFDM). The correlation with known patterns also increases the robustness of this technique. However, the correlation calculated in the WFD increases the sensing duration.

The complexity of the CFD is high due to the mathematical functions that are not common outside the field of data communications, restricting the implementation of this technique only for experienced engineers. This technique can easily distinguish one PU signal from noise if the cyclostationary parameters are well known. Finally, the CFD presents the highest sensing duration of all sensing techniques, generally one order of magnitude slower than the WFD.

The dynamic access to the radio spectrum is strongly controlled by regulatory agencies, which define several constraints in the operational parameters of the sensing techniques. The uttermost goal of these constraints is to guarantee the safety and quality of the licensed channel reserved for the PU. For example, the sensing sensibility is a parameter that indirectly impacts in the interference caused in PU and, thus, is an important parameter that must be controlled by regulatory agencies.

Performing spectrum sensing in an individual SU has a number of limitations. First of all, energy constraints can limit the frequency of sensing periods. For example, a sensor node might have to increase the interval between sensing periods to reduce the energy consumption when the battery is at critical levels. Furthermore, the SU might be located in a region where the radio signals are deeply faded, and so, making difficult the detection of any PU transmitter. In such situations, the PU transmitter is not detected but the SU could still harmfully interfere with PU receivers. This situation is well known in the literature as the hidden terminal problem and is shown in Figure 2.11.

2.4 Chapter Summary

This chapter makes a review on dynamic spectrum access, showing how SUs can access the white spaces in five different dimensions and how they can improve the overall spectrum usage. The constraints that can be imposed by regulatory agencies were also discussed, highlighting the sensing sensibility and periodicity. Afterwards, the chapter presented the SDR technology and how it enables the design, implementation and evaluation of new radio components.

The fundamentals of SS, the main techniques were discussed. The three main sensing techniques presented in the literature were detailed, namely ED, WFD and CFD. Finally, we compared different characteristics of these techniques and made clear that the basic sensing techniques presented do not suffice the requirements of sensing accuracy and sensing duration. In the next chapter the main research efforts that aimed to improve the sensing techniques are presented.

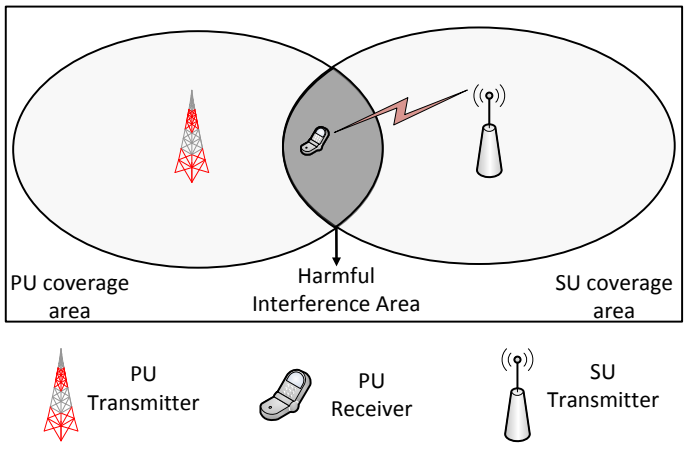


Figure 2.11: Hidden terminal problem

3 RELATED WORK

The main research to improve the sensing techniques are presented in this chapter. We divided the research into two approaches: cooperative and multi-stage sensing. Section 3.1 presents the cooperative approach, which utilizes algorithms to fuse the decision of multiple SUs into a single decision (AKYILDIZ; LO; BALAKRISHNAN, 2011). Afterward, Section 3.2 shows the multi-stage approach, which combines two or more sensing techniques in a hierarchical structure (AXELL et al., 2012).

3.1 Cooperative Spectrum Sensing

Cooperative SS arises as a prominent approach to improve the sensing accuracy of the SU by making it more robust against faded signals and the hidden terminal problem (AXELL et al., 2012). The concept of cooperative SS is to make multiple SUs send their decisions to a common receiver, which combines the decisions into one final value, as shown in Figure 3.1. The individual decisions are sent through a reporting channel, which can be a channel, allocated by governmental agencies, solely for this purpose or the same channel that is accessed dynamically. Therefore, the main challenge in this approach is to define a fusion rule, *i.e.* the algorithm used to combine the individual decisions of the SUs (CHEN; CHEN; MENG, 2014).

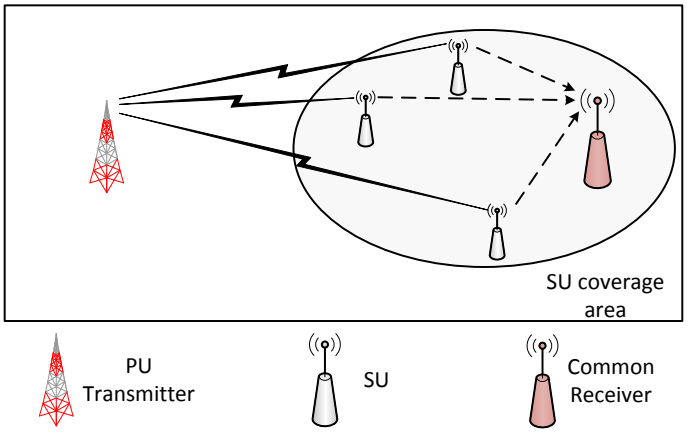


Figure 3.1: Cooperative SS architecture

Zhang, Mallik and Letaief (ZHANG; MALLIK; LETAIEF, 2009) considered a cooperative SS in which all SUs used the ED. A common receiver collects the individual decisions from k SUs and executes the fusion rule to infer the channel occupancy status. Additionally, it assumes that all SUs are concentrated in a small geographical area distant

from the PU transmitter. This leads the signal sensed in all SU to have almost identical SNR.

The objective of the authors was to derive the optimal value of n for a n-out-of-k voting rule to maximize the sensing accuracy. The value n in the voting rule defines how many of the k SU must decide that the channel is occupied so that the common receiver also considers the channel as occupied. The results presented showed that the best value for n is half the number of SUs, *i.e.* $n = \lceil \frac{k}{2} \rceil$. Additionally, it was shown that the sensing accuracy of the common receiver increases with k.

The results obtained by Zhang, Mallik, and Letaief are valid only when the distance assumption made initially is valid in a real environment. However, there are no guarantees that this assumption will hold in real radio environments. More precisely, in real radio environments, some SUs might be localized in a privileged position to detect the PU signal, while others can be localized in a region where the PU signal is deeply faded.

Peh *et al.* (PEH et al., 2010) advanced the previous research by considering a scenario where the SUs are distributed over a wide geographical area. As a consequence, the SNR of the PU signal received is different among the SUs. The objective of their research was to propose a method that attributes higher weights to the decisions of SUs that are better localized to detect the PU. When a SU sends a decision, a common receiver updates its weight by applying likelihood-ratio test (LEHMANN; ROMANO, 2006) considering the newly received decision and the previous decisions of all other SU, which are stored in a database.

The authors evaluated three variations of the proposed method. In variation *I*, the parameters that can be adjusted by the proposed method are the weights of the SUs in the common receiver, the individual decision threshold of the SU and the sensing duration of any individual SU. The authors then considered the fact that if the number of SU is large, adjusting the parameters mentioned above for all individual SU can be time-consuming. Therefore, in variation *II* the weights of all SUs in the common receiver are constrained to be the same. Consequently, the proposed method needs to compute only one optimal weight. In variation *III*, the authors considered that the SUs know the noise power of the environment and can optimize their decision threshold, while the common receiver optimizes only the weight of the SUs.

The authors simulated all three variations and compared the achieved sensing accuracy with fusion rules commonly used as a baseline in the cooperative approach, *i.e.* "and" rule and "or" rule. The "and" is a special case of the voting rule in which $n == k$, whereas for the "or" rule $n == 1$. Variation *I* and *II* presented a sensing accuracy of nearly 99%, with variation *I* being slightly above. Variation *III* had the highest sensing accuracy, at nearly 100%. This occurs because each SU user can adjust its decision threshold based on the SNR received from the PU. Finally, the "and" rule presented the worst results, while the "or" rule had a sensing accuracy between the variations *II* and *III*.

The main problem of the method proposed by Peh *et al.* is that it needs to store a significantly large number of decisions to calculate the weight of each SU reliably. Also, the authors obtained the results in a static environment, *i.e.* the SNR of the PU signal is constant in a given SU. Therefore, their proposal might be impractical in time-varying environments, *i.e.* the SNR of the PU signal varies in time and independently for the individual SUs. More precisely, the SNR variation increases the time to converge to the correct weights attributed to individual SU.

Zhang, Wu, and Lu (ZHANG; WU; LU, 2014) proposed a method to adapt the weights of the SUs in time-varying environments. To promptly react to abrupt changes in the

PU signal, the method used a temporal discount factor, which reduces the influence of previous individual decisions exponentially in time, *i.e.* a decision received in time $t-2$ receives a considerably higher discount factor than a decision received in time $t-1$.

The proposed method was compared with the "and" and "or" voting rules in a simulated environment. The results showed that using discount factors reduce the number of iterations required to find the optimal weights for the individual SUs. In addition, the proposed discount factor outperformed the "and" and "or" voting rules when individual SUs suffered from abrupt SNR changes.

The research in the cooperative approach mentioned so far make two assumptions to simplify the gathering of results: (*I*) the reporting channel is error-free, *i.e.* without noise, and (*II*) the SNR of the PU signal is known in the individual SU. However, these assumptions are not practical in a real radio environment and can lead to a wrong picture of the real sensing accuracy of the cooperative approach (SHEN et al., 2009).

Atapattu, Tellambura and Jiang (ATAPATTU; TELLAMBURA; JIANG, 2011) evaluated the sensing accuracy of the cooperative approach in a noisy reporting channel. They evaluated two different methods for reporting the SS results for the common receiver: "decision" and "data". In the first method, only the decisions are sent to the common receiver. Because the decision is represented in a single bit of data, this method requires a reporting channel with a very small channel bandwidth. In the second method, each SU acted as a relay, amplifying and sending a replica of the received signal to the common receiver. Because a replica is sent, this method requires a reporting channel with a bandwidth equal to the received signal.

The authors evaluated the "decision" and "data" methods in the noisy reporting channel analytically. The results showed that the sensing accuracy of both methods is reduced in the noisy channel when compared to an error-free channel. Additionally, the sensing accuracy in both methods decreases rapidly when the number of hops between the common receiver is increased. Also, the ROC curve of the "decision" method could not reach $P_{FA} = 1$ and $P_D = 1$ at the right upper corner and cannot reach $P_{FA} = 0$ and $P_D = 0$ at the left lower corner.

Although the cooperative approach can increase the sensing accuracy, the final gain is limited by many factors. For example, the individual decisions of the SUs are correlated when they are blocked by the same obstacles. The sensing accuracy might be severely degraded if the number of SUs blocked by obstacles is considerable (GHASEMI; SOUSA, 2005). In addition, cooperative SS incurs in cooperation overhead. The overhead refers to any extra sensing duration, delay, energy, and operations devoted to cooperative sensing compared to the individual SS. Moreover, vulnerability to security attacks is also a part of the cooperation overhead (AKYILDIZ; LO; BALAKRISHNAN, 2011). Multi-stage architectures were proposed in the literature to improve the SS of individual SU without the need for cooperation.

3.2 Multi-stage Spectrum Sensing

The objective of the multi-stage approach is to reduce the sensing duration and increase the sensing accuracy of an individual SUs by combining sensing techniques in a hierarchical structure. Thus, a sensing technique is applied in the received signal in each "stage". This approach enables the SU to benefit from the best characteristics of different sensing techniques. It is noteworthy that the multi-stage approach combines different sensing techniques locally in a SU, while the cooperative approach combines the deci-

sions of different SUs. Because of this, multi-stage architectures can be easily integrated into a cooperative solution. More precisely, the single-stage solutions can be replaced by a multi-stage solution without any modification in the common receiver.

Luo *et al.* (LUO et al., 2009) proposed a two-stage architecture in which the first stage performed a coarse sensing over a large bandwidth of spectrum and the second stage performed a fine sensing[1]. More precisely, the radio spectrum is divided into several contiguous portions of equal bandwidth called "coarse blocks", which usually comprise two or more PU channels. The first stage senses all the coarse blocks sequentially. Afterward, the second stage is applied in the blocks identified as vacant to detect precisely what channel inside this coarse block is vacant.

The authors considered that the SU executed the ED in the first and second stages. Moreover, the entire radio spectrum accessible by the SU is contiguous and comprise multiple channels (P) of equal bandwidth. The results, obtained analytically and through simulations, showed that the proposed solution using the ED outperforms the traditional single-stage ED in sensing duration when the number of vacant channel (L) is very high ($L/P \rightarrow 1$). However, the sensing duration required by the proposal increases dramatically as the number of vacant channels is reduced, and eventually becomes slightly higher than that of the single-stage ED. This occurs because the number of fine sensing steps increases when $L/P \rightarrow 1$.

Maleki, Pandharipande, and Leus (MALEKI; PANDHARIPANDE; LEUS, 2010) proposed a two-stage architecture that executes an ED in the first stage and a CFD in the second stage. In this architecture, the first stage is always active, whereas the second is activated only when the first stage evaluates the channel as vacant. When this occurs, the decision of the second stage is considered as correct, even if it contradicts the one provided by the first stage. This research is used as the baseline for other multi-stage solutions proposed in the literature and, also, in this dissertation. Because of its relevance, the architecture of Maleki, Pandharipande, and Leus is depicted in Figure 3.2.

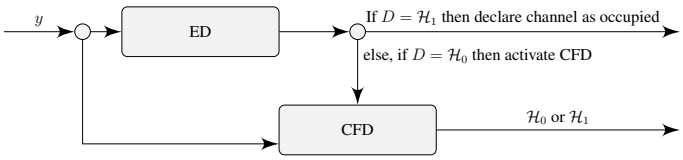


Figure 3.2: Two-stage architecture proposed by Maleki, Pandharipande, and Leus

Using an analytical evaluation, the authors showed that combining the ED and the CFD in a multi-stage architecture increases the sensing accuracy when compared to the single-stage version of the ED. In addition, the two-stage architecture reduced the sensing duration when compared to the single-stage CFD. Consequently, this architecture allied the best of the single-stage version of the ED and the CFD, which is a low sensing duration and a high accuracy.

The proposal of Maleki, Pandharipande, and Leus has a simple mechanism to control activations of the second stage. More precisely, the second stage of their proposal is activated whenever the first stage declares the channel as vacant. Thus, the sensing duration of this architecture might be severely increased when the sensed channel is often vacant. In addition, the sensing accuracy of the architecture may be degraded if the de-

[1]The nomenclature found in the literature refers to the first technique activated on the received signal as the "first stage", the next technique as the "second stage" and so on.

cision threshold of the first stage is set to a value below the expected. This may lead the first stage to frequently declare the channel as occupied and not activate the second stage.

Nair *et al.* (NAIR; VINOD; KRISHNA, 2011) proposed a mechanism to control the activation of the second stage, aiming to reduce the sensing duration of the solution proposed by Maleki, Pandharipande, and Leus. The authors performed a rigorous analysis to obtain the minimum SNR at which the first stage can operate without using the second stage. Thus, the second stage is never activated when the sensed channel presents a SNR above the minimum required. This SNR was calculated according to a P_{FA}, *i.e.* the probability of considering a channel occupied when it is vacant.

The authors considered that both first and second stages executed an ED to obtain the results. The target P_D and P_{FA} were kept at 90% and 10%, respectively. Also, the PU has occupied the sensed channel in 80% of the time, *i.e.* $P_{\mathcal{H}_0} = 80\%$. The results showed that the control mechanism could inhibit the execution of the second stage when the SNR is greater than -15.4 dB, thus reducing the sensing duration compared to the work of Maleki, Pandharipande, and Leus. Moreover, the sensing accuracy was increased in relation to the single-stage ED for SNR below -15.4 dB.

Ejaz, Hasan, and Kim (EJAZ; HASAN; KIM, 2012) extended the architecture of Maleki, Pandharipande, and Leus by adding a third stage that executed a matched filter detector (PROAKIS, 2007). This detector is similar to the WFD, but with nearly perfect accuracy, and low sensing duration. Despite these advantages, its overall complexity is the major factor that inhibits its adoption. In their proposal, the third stage is parallel to the first and second stages and it is activated only when specific radio channels are sensed. These radio channels are the ones whose the PU signal is well known, and therefore, detectable by the matched filter.

Similarly to other multi-stage solutions, the results presented by the authors showed that the proposed architecture reduces the sensing duration when compared to the slowest detector used, in this case, the CFD. In addition, the sensing accuracy is increased when compared to the single-stage ED.

The main constraint of the research in cooperative and multi-stage solutions is the use of static decision thresholds, which is a problem in dynamic radio environments. For example, the SU may encounter different noise or interference levels when switching between different public safety radio channels, requiring different values for the decision threshold. Moreover, SUs may be operating in unknown radio environments and may not have perfect knowledge of the characteristics of the other existing PUs or SUs (YUCEK; ARSLAN, 2009). Another constraint, specifically for the multi-stage approach, is the lack of an advanced mechanism to activate the second stage.

According to Bkassiny, Li, and Jayaweera (BKASSINY; LI; JAYAWEERA, 2013), the use of machine learning allows the SU to increase its performance in unknown radio environments by adapting the decision threshold. In this sense, we propose a multi-stage adaptive threshold architecture that allies the multi-stage approach with machine learning to overcome the identified constraints in the state-of-the-art research.

3.3 Chapter Summary

Throughout this chapter, the main research to improve the SS was presented. Initially, the research efforts on the cooperative approach were presented, with a focus on methods to efficiently fuse the decisions received from the SUs. Although this approach can efficiently increase the sensing accuracy, the overhead added, the need for a report-

ing channel and security vulnerabilities are drawbacks that can make the adoption of this approach unfeasible in certain environments.

Afterward, the chapter presented the research efforts on the multi-stage approach. In this approach, the focus is to combine two or more SS techniques in such a way that the improvement surpasses the single-stage version of each one. The results showed that combining two or more sensing techniques in a single SU can improve the SS significantly. Despite the improvement, multi-stage architectures are limited to static decision thresholds, which is a problem for dynamic radio environments. In the next chapter, we present our proposal: a multi-stage solution that uses machine learning techniques to adapt the decision threshold in real time.

4 ADAPTIVE THRESHOLD ARCHITECTURE

This chapter begins by describing the general overview of the proposed Multi-Stage Adaptive Threshold Architecture (ATA) in Section 4.1. In this overview, we describe the basic functionality of the *Sensing Component* and the *Machine Learning Component*. Afterward, in Section 4.2, each component is detailed, with focus on the operations performed and the order that they occur in ATA operation.

4.1 Multi-Stage Adaptive Threshold Architecture Overview

ATA has three main components, each of them with very specific functionalities, as shown in Figure 4.1. Considering a bottom-up approach, the three components are: (*I*) the *RF Front-End*, detailed in Subsection 4.1.1, (*II*) *Sensing Component*, which is presented in Subsection 4.1.2, and (*III*) the *Machine Learning Component*, presented in details in Subsection 4.1.3.

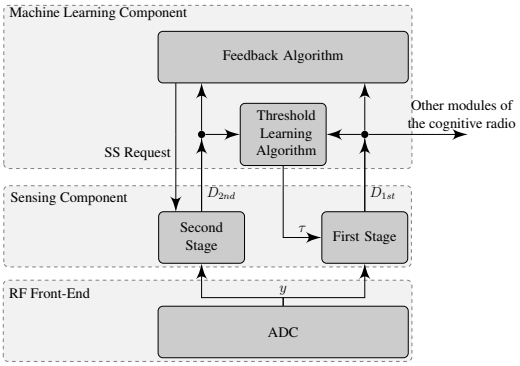


Figure 4.1: Proposed architecture

4.1.1 RF Front-End

This component is responsible for receiving and digitizing the signal of a given channel of the radio spectrum. Although this component is used in all operations involving radio signals, such as reception and transmission of useful data, ATA uses it only to acquire digitized signal samples during the sensing period. In addition, ATA assumes that the *RF Front-End* is correctly configured when the sensing period starts. Thus, parameters such as the central frequency, bandwidth, and sample rate must be configured before the sensing period. These configurations can be performed, for example, by other modules of the SU radio, such as a module responsible for managing all the configurations required when a sensing or transmission period start.

4.1.2 Sensing Component

This component performs the operations required to detect if the sensed channel is occupied or vacant. This component utilizes a multi-stage architecture with two sensing techniques to enable ATA to achieve a high sensing accuracy while maintaining a low sensing duration.

The *First Stage* is always active, *i.e.* during the sensing period this stage provides several decisions about the occupancy status. The inputs of this stage are the digitized signal y, from the *RF Front-End*, and the decision threshold τ, from the *Machine Learning Component*. The output is the decision that indicates if the channel is vacant or occupied. The decision is sent to the *Threshold Learning Algorithm*, the *Feedback Algorithm* and to SU radio modules that will use it to select which channel of the radio spectrum should be accessed.

The *Second Stage* enables the adaptation of the decision threshold used by the *First Stage*. The inputs are the digitized signal y and the *SS Request*, which is detailed in Subsection 4.1.3. The output is the decision that indicates the channel status occupancy, which is sent to the *Machine Learning Component*. With the purpose of reducing the sensing duration of ATA, this stage is not continuously activated. More precisely, the *Second Stage* is activated only after receiving a *SS Request*.

We highlight that the *Second Stage* is only activated after the *First Stage* finishes its evaluation of a given signal. Lets define as t_{1st} the time required for the *First Stage* to evaluate a given signal y once and t_{2nd} the time required for the *Second Stage*. Thus, when the *Second Stage* is activated, the time to evaluate a given signal y is $t_{1st} + t_{2nd}$. More generally, the average sensing time of ATA can be expressed as shown in Equation 4.1, where n_{1st} and n_{2nd} are the total activations of the *First* and *Second Stage*, respectively. Based on this equation, we can conclude that the average sensing duration is increased the more the *Second Stage* is activated.

$$t_{total} = \frac{(n_{1st} - n_{2nd})t_{1st} + n_{2nd}(t_{1st} + t_{2nd})}{n_{1st} + n_{2nd}} \tag{4.1}$$

The selection of which sensing techniques will be executed in each stage depends on both the sensing accuracy and the sensing duration required by the SU. Although any sensing technique could be used, to obtain the best performance from ATA, a faster technique should be used in the *First Stage*, *e.g.,* ED, and a more accurate technique in the *Second Stage*, *e.g.,* WFD or CFD. This is required because the *First Stage* is continuously activated to provide updated decisions regarding the channel occupancy status, whereas the *Second Stage* is activated only when the decision threshold must be adapted, *i.e.* increase, and decrease or maintain the decision threshold.

4.1.3 Machine Learning Component

The *Machine Learning Component* is the main novelty presented in ATA, filling the drawbacks of current SS solutions. This component is divided in two algorithms: (*I*) the *Threshold Learning Algorithm*, which adapts the decision threshold τ of the *First Stage* and (*II*) *Feedback Algorithm*, which controls the activation of the *Second Stage*.

The *Threshold Learning Algorithm* adapts the decision threshold τ of the *First Stage* based on the decision received from the *Sensing Component*. For this adaptation, the *Threshold Learning Algorithm* assumes that the decision of the *Second Stage* is more accurate than the decision of the *First Stage*. Based on this assumption, the learning

algorithm can compare the received decisions and adapt τ in order to make the decisions of both stages equal. In this context, the decision of the *Second Stage* is a "feedback" for the *Threshold Learning Algorithm*. Although the algorithm must rely on the comparison of decisions to adapt the decision threshold, different machine learning algorithms can be used in the adaptation, such as Bayesian learning (GONG et al., 2009) and Q-learning (FAGANELO et al., 2013).

The *Feedback Algorithm* reduces the sensing duration by controlling the activations of the *Second Stage*. As inputs, the algorithm receives the decisions of both stages of the *Sensing Component*. Based on these inputs, the algorithm must calculate the interval until the next activation of the *Second Stage*, and consequently, until an updated feedback is provided to the *Threshold Learning Algorithm* (more details about the interval are provided in the next section). The output of this block is the *SS Request*, which is sent only when the calculated interval has expired. If the decisions received from the *Sensing Component* converged to the same hypothesis, the *Feedback Algorithm* increases the interval between activations of the *Second Stage*. As a consequence, the sensing duration is reduced because only the *First Stage* is activated. Similarly, if the decisions did not converge to the same hypothesis, the interval is reduced, which increases the sensing duration but also quickens the threshold adaptation.

It is worth highlighting that the *Feedback Algorithm* and the *Threshold Learning Algorithm* are independent. Despite this, a better sensing accuracy and a lower sensing duration can be achieved if the *Feedback Algorithm* is designed considering the limitations of the learning algorithm. For example, given that a learning algorithm requires frequent feedbacks to adapt the decision threshold τ, then the interval between feedbacks could be limited to a certain value to help the threshold adaptation.

4.2 Detailing the Adaptive Threshold Architecture

In this section, the details of each block are presented. The functionality of each block is explained with the help of a sequence diagram, which illustrates the main interactions and operations performed. It is noteworthy that the sequence diagrams comprise the operation of ATA only during a sensing period. In addition, considerations regarding possible optimizations are made.

4.2.1 First Stage

The operations performed by the *First Stage* are illustrated in the sequence diagram in Figure 4.2. In the *digitize* operation of the *ADC*, N samples of the signal y are collected. Afterwards, the signal y is sent to the *First Stage* and *Second Stage* in the *signalReady* operation. After receiving the signal y, the *First Stage* applies the test statistic Λ, generating the decision D_{1st}. Finally, the decision D_{1st} is sent to the *Feedback Algorithm* and the *Threshold Learning Algorithm*, in the *updateDecision* operation. After this, the entire sequence is repeated again until the sensing period is over.

It is worth highlighting that the first *signalReady* operation do not block the execution of the next operation of the ADC. More precisely, the second *signalReady* is sent without waiting the *First Stage* to finish its operation. This also applies for the *updateDecision*.

An optimization that can be applied in this sequence is starting the *ADC digitize* operation right after the *signalReady*. Thus, when the *First Stage* finishes its operations, the ADC can immediately send a new signal y. This optimization can lead to a significant reduction in the sensing duration because the *digitize* and Λ operations are run in parallel.

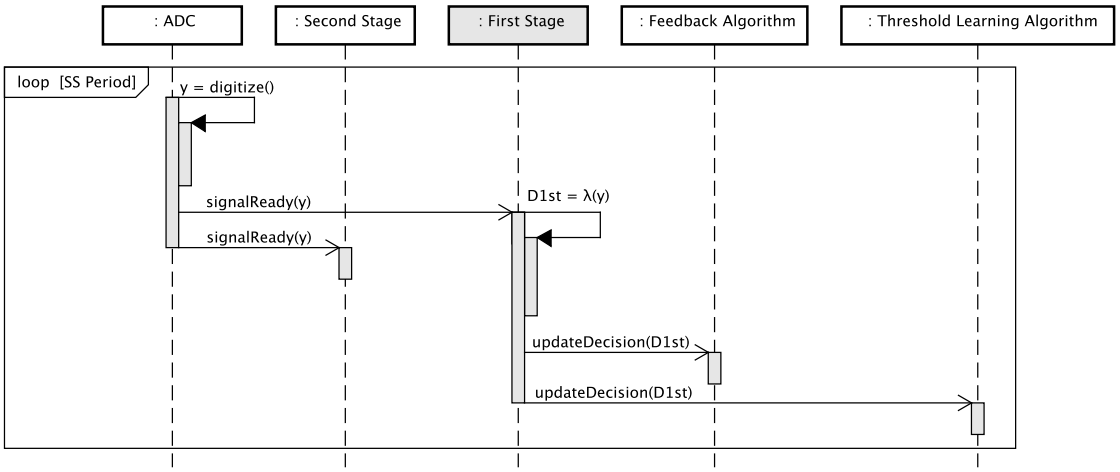


Figure 4.2: Sequence diagram for the *First Stage*

A race condition arises if the *digitize* operation is faster than the test statistic Λ. In such a case, the *signalReady* might be executed while the test statistic is being executed in the *First Stage*. This race condition can be eliminated by discarding any signal received if the test statistic Λ is in execution in the *First Stage*.

4.2.2 Second Stage

The sequence diagram of the *Second Stage* is illustrated in Figure 4.3. Since this block must receive a *SSRequest* to start its operation, we show the *SSRequest* as the first operation performed. After receiving the request, the *Second Stage* awaits until the signal y is provided by the ADC. When this occurs, the test statistic Λ is applied to obtain the decision D_{2nd}. In the sequence, D_{2nd} is sent to the *Feedback Algorithm* and the *Threshold Learning Algorithm* in the *updateDecision* operation. After this operation, the *Second Stage* awaits until the next *SSRequest* is received. In the meantime, the *signalReady* operations received are ignored.

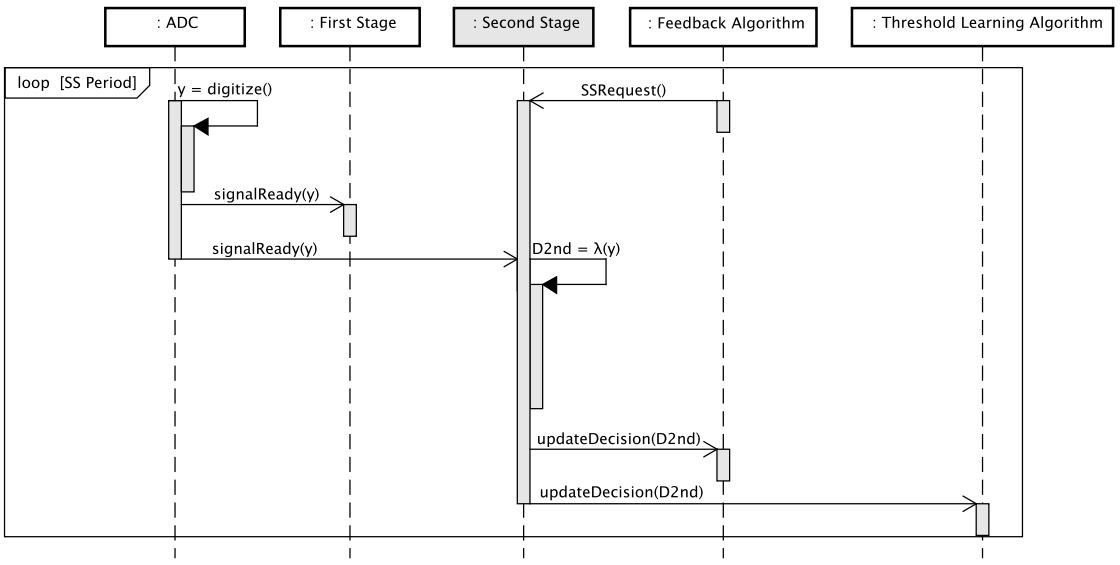


Figure 4.3: Sequence diagram for the *Second Stage*

An optimization that can reduce the sensing duration is activating the *Second Stage* in parallel with the *First Stage*. As a consequence, a control must be added to avoid race conditions when the *First Stage* is considerably faster than the *Second Stage*. In such case, the *Machine Learning Component* can receive multiple decisions D_{1st}, while the *Second Stage* is applying the test statistic in an old signal y. This control must be an operation indicating that *First Stage* cannot generate any new decision until the *Second Stage* generated a D_{2nd}.

4.2.3 Threshold Learning Algorithm

The operations performed by this block are detailed in the sequence diagram in Figure 4.4. We note that functions $updateDecision$, $new\tau$, and $setnew\tau$ are generic and its specific implementation depends on the machine learning algorithm used. The operation of this block starts only when the feedback, *i.e.* the decision D_{2nd}, is received. When this occurs, the *new*τ operation adapts the decision threshold τ based on the received D_{2nd} and the latest D_{1st}. The *new*τ operation uses the machine learning techniques to select the next value for τ. Afterward, the selected value for τ is informed to the *First Stage* in the *setnew*τ operation.

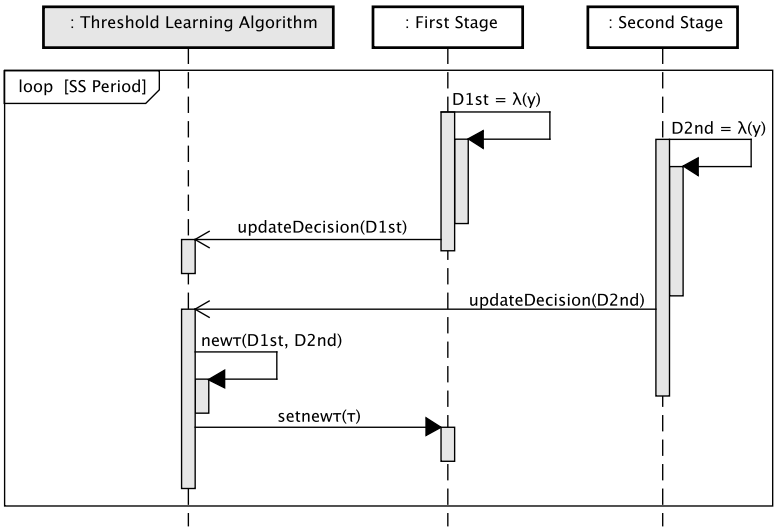


Figure 4.4: Sequence diagram for the *Threshold Learning Algorithm*

The *Threshold Learning Algorithm* must adapt the decision threshold τ in such a way that the decisions D_{1st} and D_{2nd} become equal. As mentioned, the learning algorithm assumes that the decisions D_{2nd} reflect the real status of the channel (D_R) and tries to make D_{1st} converge to the same set of hypotheses. We can analyze how the threshold τ must be adapted based on the combinations of D_{1st} and D_R. More precisely, based on these combinations, the threshold τ must be decreased ($\downarrow$), increased ($\uparrow$) or maintained (–) in order to make D_{1st} equal to D_R. Table 4.1 shows combinations of D_{1st} and D_R and the expected adaptation of τ. A detailed explanation of the expected adjustment for τ is:

- $D_{1st} = D_R$: the learning algorithm does not need to adapt the decision threshold τ because the detector is providing accurate decisions.
- $D_{1st} = \mathcal{H}_0$ and $D_R = \mathcal{H}_1$: the decision threshold τ must be decreased. For example, given a situation where the test statistic over the received signal y resulted in $\Lambda(y) = \tau - 1$. By comparing both $\Lambda(y)$ and τ, the decision is $D_{1st} =$

$\left(\Lambda(y) \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \tau\right) = \mathcal{H}_0$. Thus, the decision threshold τ must be decreased in order to make $D_{1st} = D_R = \mathcal{H}_1$.

- $D_{1st} = \mathcal{H}_1$ and $D_R = \mathcal{H}_0$: the decision threshold τ must be increased. Considering a similar example from the previous item, but with the test statistic over the received signal y resulting in $\Lambda(y) = \tau + 1$, the decision is $D_{1st} = \left(\Lambda(y) \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \tau\right) = \mathcal{H}_1$. Thus, the decision threshold τ must be increased in order to make $D_{1st} = D_R = \mathcal{H}_1$.

		D_R	
		$\mathcal{H}_0$	$\mathcal{H}_1$
D_{1st}	$\mathcal{H}_0$	–	$\downarrow$
	$\mathcal{H}_1$	$\uparrow$	–

Table 4.1: Expected adaptation of τ considering D_{1st} and D_R

The combinations in Table 4.1 show the expected adaptation of the decision threshold τ based on D_R. In practice, this adjustment is based on D_{2nd}, provided by the *Second Stage*, which may be incorrect. Therefore, the combinations can be extended considering the hypotheses for D_{1st}, D_{2nd} and D_R, which are shown in Tables 4.2a and 4.2b. The first table presents the combinations for $D_R = \mathcal{H}_0$ and the second one for $D_R = \mathcal{H}_1$ The incorrect hypothesis for D_{2nd} is highlighted in both tables, *i.e.* $\mathcal{H}_1$ in Table 4.2a and $\mathcal{H}_0$ in Table 4.2b. In such cases, the machine learning algorithm will not be able to find the correct value for the decision threshold τ.

It is noteworthy that the convergence to a correct threshold occurs only when the *Second Stage* has an accuracy above 50%. More precisely, for any learning algorithm "learn" the correct threshold, at least more than half of its inputs must indicate the direction (increase or decrease) correctly. In addition, the decision threshold will converge faster for higher accuracies and to a wrong value if the accuracy is below 50%. Thus, it is of utmost importance to ensure that the *Second Stage* have the highest possible sensing accuracy.

		D_{2nd}	
		$\mathcal{H}_0$	$\mathcal{H}_1$
D_{1st}	$\mathcal{H}_0$	–	$\downarrow$
	$\mathcal{H}_1$	$\uparrow$	–

(a) $D_R = \mathcal{H}_0$

		D_{2nd}	
		$\mathcal{H}_0$	$\mathcal{H}_1$
D_{1st}	$\mathcal{H}_0$	–	$\downarrow$
	$\mathcal{H}_1$	$\uparrow$	–

(b) $D_R = \mathcal{H}_1$

Table 4.2: Expected adaptation of τ considering D_{1st}, D_{2nd} and D_R

4.2.4 Feedback Algorithm

The sequence diagram of the *Feedback Algorithm* is illustrated in Figure 4.5. The *Feedback Algorithm* sends the *SSRequest* as soon as its operation is started. The algorithm waits until the decision D_{2nd} is received from the *Second Stage*. In the sequence, the new *interval* is calculated, based on latest D_{1st} and D_{2nd}. After this, the *Feedback Algorithm* enters a loop, which is left only when *interval* decisions are received from the *First Stage*. After this, the entire sequence is repeated again.

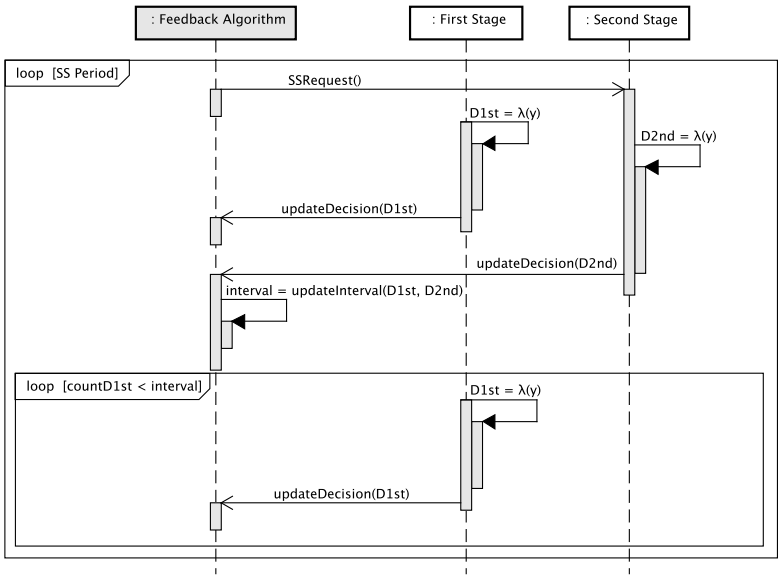


Figure 4.5: Sequence diagram for the *Feedback Algorithm*

The *interval* is measured in decisions received from the *First Stage* and not in duration. Although it might seem strange at first glance, using the former brings some advantages over the latter: (*I*) it is easier to design the algorithm because the sensing duration of the *First* and *Second Stages* are abstracted, (*II*) the same algorithm can be used, without modifications, with any combination of SS techniques, and (*III*) it is easier to meet the requirements of different *Threshold Learning Algorithm* regarding the frequency of feedbacks.

ATA eliminates the drawback of current SS solutions by adding a machine learning algorithm to dynamically adapt the decision threshold τ. With this adaptation, ATA can maintain a high sensing accuracy in different radio environments. In addition, the drawbacks related to controlling the activations of the second stage were mitigated by the introduction of the *Feedback Algorithm*. Therefore, ATA combines the advantages of a multi-stage architecture with the capability of adaptation to different radio environments.

4.3 Chapter Summary

This chapter presented the Multi-Stage Adaptive Threshold Architecture (ATA) for spectrum sensing. The chapter began with an overview of the architecture, focusing on describing the functionality of its three components, namely *RF Front-End*, *Sensing Component* and, *Machine Learning Component*. The first component is responsible for receiving and digitizing the signal of a given channel of the radio spectrum. The second component employs a two-stage architecture to perform the detection of PUs in the digitized signal and to help in the adaptation of the decision threshold. The third component uses a *Threshold Learning Algorithm* to adapt the decision threshold τ used in the *Sensing Component* and a *Feedback Algorithm* to reduce the sensing duration.

In the sequence, a detailed description of the four blocks of the architecture was presented. (*I*) the *First Stage*, used to provide fast decisions regarding the PU presence in the sensed channel, (*II*) the *Second Stage*, which is used as feedback to adapt the decision threshold, (*III*) the *Feedback Algorithm*, which reduces the average sensing duration of ATA by controlling the activations of the *Second Stage* and, (*IV*) the *Threshold Learning*

Algorithm, which dynamically adapts the decision threshold τ of the *First Stage* based on the feedback received.

ATA improves current SS solutions by introducing the use of machine learning to increase the sensing accuracy and a mechanism to maintain a low sensing duration. In addition, the dynamic adaptation of the decision threshold enables ATA to maintain its performance in different radio environments. In the next chapter, the prototype of ATA is detailed.

5 PROTOTYPE

This chapter presents the prototype developed to evaluate ATA. The SDR platform used in the prototype is detailed in Section 5.1 The sensing techniques utilized in the evaluation are detailed with the help of block diagrams in Section 5.2. Finally, the *Machine Learning Component* is detailed in Section 5.3. In addition, algorithms developed for this dissertation are detailed in pseudocode.

5.1 Software Defined Radio Platform

The SDR platform used to implement the prototype of ATA is divided into two parts: hardware and software. The first one, which comprises the *RF Front-end*, is presented in Subsection 5.1.1. The second one, in which the *Sensing* and *Machine Learning Components* were developed, is presented in Subsection 5.1.2

5.1.1 Hardware part: Universal Software Radio Peripheral

The RF front-end of the developed prototype is the USRP. The USRP is an equipment developed by Ettus Research (Ettus Research, 2008) to transform any General Purpose Machine (GPM), *e.g.,* a personal computer, in a complete SDR platform. It is one of the most popular equipment of its class, widely adopted in the academia for research purposes (Aguayo Gonzalez; DIETRICH; REED, 2009), (ULVERSOY, 2010), (KIM; XIN; RANGARAJAN, 2010), (GU et al., 2010).

The USRP is divided into daughterboards and one motherboard. The number of daughterboards depends on the USRP model, going from at least two to more than ten. Each daughterboard is connected to a unique ADC and DAC, which in turn, are connected to the motherboard. The motherboard is responsible to communicate with the GPM for transferring IQ samples from/to the software modules running in it. The daughterboards are used to hold the RF receiver and the RF transmitter interfaces.

Each daughterboard has an independent RF path and an exclusive antenna. This allows a single USRP device to operate simultaneously in different portions of the radio spectrum, increasing the range of applications that can be used with the USRP. In practice, a single equipment can be used for evaluations in simple scenarios. For example, a single device can be used to transmit and receive a radio signal simultaneously. With this simple setup, several algorithms involved in both transmission and reception can be evaluated.

Ettus commercialize up to seven different models of USRP and daughterboards. In this dissertation, we focus on the equipment used in this dissertation, which is the USRP model N210 and the WBX daughterboard. This model has two slots for connecting up to

two daughterboards, namely TX/RX and RX2. In the first, a daughterboard with reception and/or transmission capabilities can be connected, whereas the second accepts only daughterboards with reception capabilities. Figure 5.1 illustrates the architecture of the USRP board and the integration with a GPM.

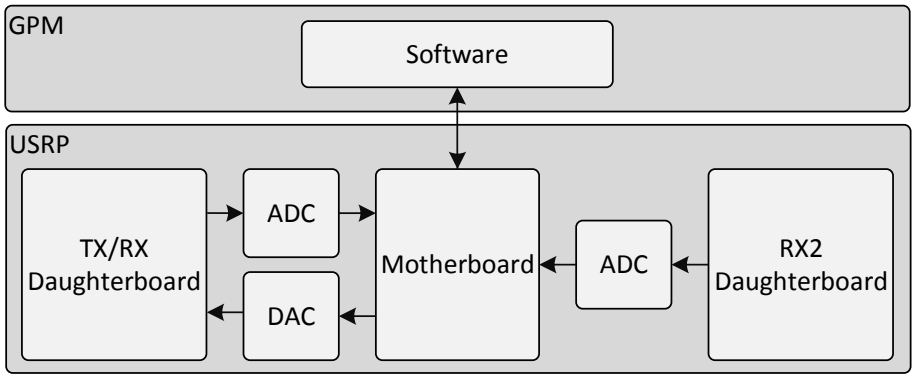


Figure 5.1: USRP N210 and integration with a GPM

The N210 is equipped with an Altera Cyclone EP1C12 FPGA (ALTERA, 2003). The ADC used is capable to digitize 100 millions of samples per second (Msamples/s) with a resolution of 14 bits. With this, the board can receive a radio signal with a bandwidth of 50 MHz. The DAC can generate 400 Msamples/s with 16 bits of resolution, which enables the transmission of signals with 200 MHz of bandwidth. The equipment has a frequency accuracy of 2.5 ppm, *i.e.* an error of ± 2.5 Hz for each 1 MHz of difference from 0 Hz. All communications with the GPM, such as configurations and transference of digital samples, are performed through a Gigabit Ethernet interface. Given these characteristics, the USRP hardware is ideally suited for applications requiring high RF performance and great bandwidth. Such applications include dynamic spectrum access and cognitive radios.

The WBX daughterboard is a wide bandwidth transmitter and receiver. This daughterboard has a bandwidth capacity of 40 MHz and can be tuned in any frequency between 50 MHz and 2.2 GHz. It is capable to transmit signals with up to 100 mW of power and receive signals with a noise figure, which indicates how much noise the equipment itself adds to the received signal, of 5 dB. The main characteristics of a radio device combining the N210 with a WBX daughterboard are summarized in Table 5.1.

Parameter	Value	Unit
ADC sample rate	100	Msamples/s
ADC resolution	14	bits
DAC sample rate	400	Msamples/s
DAC resolution	16	bits
Noise figure	5	dB
Frequency accuracy	2.5	ppm
Frequency range	$40 - 2200$	MHz
Maximum bandwidth	40	MHz
Transmission power	100	mW
Ethernet Interface	1	Gbps

Table 5.1: USRP N210 with WBX daughterboard overview

5.1.2 Software part: GNU Radio

Since the beginning of its development in 1998, GNU Radio became the most popular software toolkit for advanced applications in SDR (GNU Radio, 2004). GNU Radio stands among other SDR toolkits due to its distribution under the GNU Public License (GPL) model. This means that GNU Radio source code can be downloaded and used by anyone. Also, GPL grants the right to modify the original source code. To use the full potential of this toolkit, the application developer must have some degree of competence with object-oriented programming, Linux operating systems, and wireless communications systems.

The signal processing in GNU Radio is performed by "blocks". A block is responsible to perform simple and specific operations. GNU Radio has several blocks for the most common signal processing tasks, such as mathematical operations, modulators, demodulators, and error correction algorithms. A block in the GNU Radio is usually developed in C++ and must be of one of the following types: "source", "processing", or "sink".

Blocks of the type "source" are used to produce digital data. The data can be produced either through mathematical functions, such as a random number generator, or through a RF front-end, such as a USRP with a reception daughterboard. Blocks of the type "processing" modify the data received and provide the resulting data in an output port. Modulators, multiplexers, filters, and error correctors are examples of processing blocks. Blocks of the type "sink" are similar to "processing" blocks, but the modified data is not provided in an output port. USRPs with a transmitter daughterboard, files and speakers are examples of sink blocks. The blocks that come in an installation of GNU Radio are all developed in C++. In addition, third-party developers can implement new blocks either in C++ for high-performance, or Python, for fast development but with a slightly higher execution time.

A sequence of blocks connected together are referred to as a signal processing path. This connection can also be performed in C++ or Python. Is worth highlighting that connecting blocks in Python presents a minimal performance loss when compared to C++. This occurs because GNU Radio uses an intermediate layer to bind Python and C++ code, called SWIG. More precisely, the Python language is used only to connect the blocks, while all the processing is made in blocks developed in C++. In addition, an application can have multiple independent signal processing paths. Figure 5.2 illustrates the GNU Radio toolkit architecture.

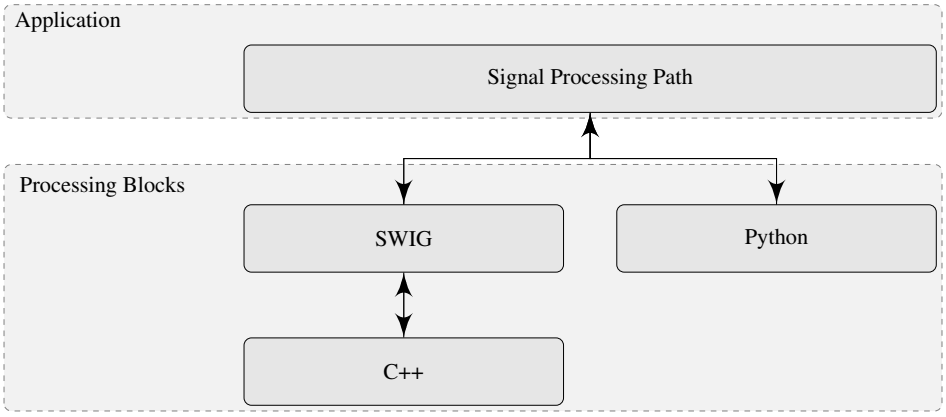


Figure 5.2: GNU Radio architecture

A basic installation of GNU Radio usually has the source and sink blocks that rep-

resent USRP equipments. These blocks provide functions that allow the configuration of several parameters of the equipment, such as the central frequency, sample rate, bandwidth, and gain. It is important to note that any GNU Radio application is always executed in the GPM, even when using a USRP. More precisely, the block representing a source USRP is responsible for digitizing the received radio signal and sending the samples to be processed in the application executing in the GPM. Similarly, the sink USRP receives the digitized samples from the GPM and converts them to radio signals, which will be transmitted in the antenna.

This section presented the SDR platform used to prototype ATA in details, including the *RF Front-End*, the GNU Radio toolkit, and their integration. In the following section, we present how the *Sensing Component* was prototyped.

5.2 Sensing Component

The prototype of ATA was evaluated using combinations of ED, WFD and CFD in the *First* and *Second Stages*. In this section, we detail how these three sensing techniques were implemented in the GNU Radio toolkit.

5.2.1 Energy Detection

The blocks used in the ED's signal processing path to detect a transmitting PU are illustrated in Figure 5.3. The *USRP Source* block generates a continuous stream of samples from the signal received in the antenna. The *Serial to Parallel* stores the last N samples received from the previous block in a buffer. When the buffer is filled, a single vector containing the N samples is sent to the FFT block. It is worth highlighting that this vector represents the signal y used to perform the PU detection. Thus, henceforward the vector of samples will be referred to as signal y. The FFT block transforms the signal y from time to frequency domain.

Figure 5.3: Signal processing path for the ED

The *Magnitude and Square* block receives the signal in the frequency domain and performs two operations in each of the N samples. The first operation is to calculate the magnitude of each sample. The second is to replace each sample by the square of its magnitude. After these two operations, each sample of the signal y will contain the energy of each frequency received. Afterward, the N samples are summed up in a value that quantifies the energy of the received signal. The value generated in the *Sum* block corresponds to the tests statistic $\Lambda(y)$ of the ED. Finally, the block *Compare and Decide* generates the decision hypothesis by comparing the energy of the signal y against a decision threshold τ.

The first four blocks of the ED's signal processing path are available in any GNU Radio installation. However, the blocks *Sum* and *Compare and Decide* had to be implemented to enable the prototyping of ATA. The implemented blocks are detailed in order to allow further research to be able to reproduce the results obtained in this dissertation.

Sum Block

Algorithm 1 details the operations performed in the block *Sum*. The algorithms presented throughout this chapter will present the main functionality of a block in the namesake function, which in this case is the *sum* function. In this function, the C columns and R rows of the *input* matrix are summed into a single value (line 2). Afterward, the final value *sum* is sent to the next block in the processing path. This implementation is used in the ED block, as well as the CFD, which is explained in Subsection 5.2.3.

Algorithm 1 Algorithm for the *Sum* block

Require: *input* as a matrix with C columns and R rows
1: **function** SUM(input)
2: *sum* $\leftarrow$ The sum of all elements of *input*
3: Send *sum* to the next block

Compare and Decide Block

This block performs the final operation of any SS technique, which is comparing the result of the test statistic Λ with a predefined decision threshold and selecting the appropriate hypothesis. This simple operation is detailed in Algorithm 2. Due to its simplicity, this block is also used in the signal processing path of the CFD.

Algorithm 2 Algorithm for the *Compare and Decide* block

Ensure: *input* as a single real value
Ensure: τ as a single real value
1: **function** COMPARE_AND_DECIDE(input)
2: **if** input < τ **then**
3: Send $\mathcal{H}_0$ to the next block
4: **else**
5: Send $\mathcal{H}_1$ to the next block

5.2.2 Waveform Detection

Figure 5.4 illustrates the signal processing path of the WFD. The only difference from the ED is the replacement of the last two blocks by the *Correlate and Decide*, which correlates the received signal y with a set of patterns p (readers can refer back to Chapter 2 for more information). Based on the correlation of the received signal y and each pattern, the block also decides if the sensed channel is occupied or vacant. The *Correlate and Decide*, in addition to the two blocks described previously, also was implemented and integrated with the GNU Radio toolkit.

Figure 5.4: Signal processing path for the WFD

Correlate and Decide Block

The implementation of the *Correlate and Decide* block is detailed in Algorithm 3. Internally, the algorithm stores a set of patterns (P), which contains a pattern (p) for each type of PU signal that must be detected. For example, if two different PUs must be

detected, one transmitting an OFDM signal and other transmitting a television signal, then these two patterns must be added to the set P. These patterns represent the waveform of the PU that must be identified and must be collected before the operation of ATA is started. In addition, each pattern can have a different decision threshold used in the comparison. These thresholds are stored in the vector C.

When the signal y is received, the block executes a sequence of operations for each pattern p stored (lines $2 - 7$). Firstly, the correlation of the signal y and a pattern p is calculated (line 3). Then, the algorithm verifies if the calculated correlation (c) is greater than the threshold for the pattern p (line 4). The PU identified by the pattern p is considered as occupying the sensed channel if the correlation c is greater than the predefined threshold (line 5); otherwise the channel is considered as vacant (line 7). Finally, the decision indicating an occupied channel is sent if at least one correlation was greater than the threshold (line 9); otherwise, the decision indicating a vacant channel is sent (line 11).

Algorithm 3 Algorithm for the *Correlate and Decide* block

Ensure: y is a vector with N samples
Require: P as a vector. Each element is a signal pattern p
Require: C as a vector. Each element is a threshold for a signal pattern p
Require: D as a vector. Each element is a hypothesis for a signal pattern p

```
1:  function CORRELATE_AND_DECIDE(y)
2:      for all p in P do
3:          c ← correlation of y and p
4:          if c is greater than the threshold for p in C then
5:              Add the hypothesis H1 in D for p
6:          else
7:              Add the hypothesis H0 in D for p
8:      if C has at least one pattern with hypothesis H1 then
9:          Send H1 to the next block
10:     else
11:         Send H0 to the next block
```

5.2.3 Cyclostationary Feature Detection

Figure 5.5 illustrates the signal processing path implemented for the CFD. The *Demodulates* and the *FAM* blocks are the novelties of this implementation. The first is responsible for computing the complex demodulate of the received signal y. The second computes the cyclostationary correlation of the demodulated signal. In this dissertation, a brief overview of these two blocks is made. Major details can be found in the article published by Roberts, Brown and Loomis (ROBERTS; BROWN; LOOMIS, 1991).

Figure 5.5: Signal processing path for the CFD

Demodulate Block

The operations performed by the *Demodulate* block are detailed in Algorithm 4. The inputs of this block are the latest M signals received from the previous block. Therefore, the value of M defines the maximum duration in which the cyclostationarity is searched.

The *Demodulate* block uses a matrix of M rows and N columns to store the demodulate of each sample received. The algorithm calculates the demodulate independently for each sample s of all y_i, where $i = 1, \ldots, M$ (line $2 - 4$). The demodulation of a single sample is calculated as shown in the function $D(s, i)$ (line 6). In the end, the block sends the *demodulated* matrix to the next block in the signal processing path (line 5).

Algorithm 4 Algorithm for the *Demodulate* block

Require: M as the internal FFT size
Require: *demodulates* as a matrix with M rows and N columns of real number
Require: $y_1, y_2, \ldots, y_M$ are the received signal with N samples each
1: **function** DEMODULATES($y_i, \ldots, y_M$)
2: **for all** i in $1, \ldots, M$ **do**
3: **for all** sample s of y_i **do**
4: *demodulates*$[i][$ index of $s] \leftarrow D(s,$ index of $s)$
5: Send *demodulates* to the next block
6: **function** D(s, i)
7: $r \leftarrow Re(s) * \cos(2\pi i) + Im(s) * \sin(2\pi i)$
8: $i \leftarrow [Im(s) * \cos(2\pi i) - Re(s) * \sin(2\pi i)]$
9: **return** $r + ij$

FAM Block

The *FAM* block is detailed in Algorithm 5. This block uses a matrix with M rows and $2N$ columns to store the cyclic correlations. Each row of the matrix stores the cyclic correlations from the frequency $-f_s/2$ to $f_s/2$, where f_s is the sampling rate. More precisely, the ith row will store the coefficients of the frequency $(-f_s + i)/2$. The input of this block is the *demodulates* matrix generated in the previous block, which is used to calculate the cyclostationary correlation (lines $2 - 4$). After obtaining the cyclostationary correlation, the matrix *cyclo* is sent to the next block (line 5).

The function *calc_cyclo* details the calculations performed to obtain the correlation. First, a vector to store M samples is initialized. Then, the samples i and j are correlated for each of the M received signals and the value obtained is stored in the fft_in vector (lines $8 - 9$). This calculation extracts the relation between the two samples as time passes. Afterwards, the FFT is performed in the *fft_in* vector (line 10). This calculation results in high values when the correlations between the two samples are periodic in time. Finally, the absolute value of the calculated coefficients is copied to the *cyclo* matrix (lines $11 - 13$).

In this section the implementation of the ED, WFD and CFD was detailed. These three sensing techniques were used in the prototype of ATA. In the next section, the implementation of the *Machine Learning Component* is detailed.

Algorithm 5 Algorithm for the *FAM* block

Require: *cyclo* is a matrix of M rows and $2N$ columns of real number
Require: *demodulates* is a matrix with M rows and N columns of real number

```
1: function FAM(demodulates)
2:     for all i in N do
3:         for all j in N do
4:             CALC_CYCLO(i, j)
5:     Send cyclo to the next block
6: function CALC_CYCLO(i, j)
7:     fft_in is an empty vector of M elements
8:     for all m in 1, ..., M do
9:         fft_in ← demodulates[m][i] * demodulates*[m][j]
10:    fft_out ← FFT(fft_in)
11:    col ← i + j − 1
12:    for all m in 1, ..., M do
13:        cyclo[i][col] ← abs(fft_out[m])
```

5.3 Machine Learning Component

In this section, the implementation of the *Feedback Algorithm* and the *Threshold Learning Algorithm* is detailed. These algorithms were developed and integrated with the GNU Radio toolkit.

5.3.1 Feedback Algorithm

The details of the *Feedback Algorithm* are shown in Algorithm 6. The configuration parameters are the maximum possible interval between two consecutive feedbacks (I_{max}) and the base of an exponential function (b). The inputs are the decisions from the *First Stage* (D_{1st}) and from the *Second Stage* (D_{2nd}).

The first operation performed by the *Feedback Algorithm* is requesting an updated decision to the *Second Stage* (line 3). In the sequence, the *Feedback Algorithm* waits until the requested decision (line 4) and the decision D_{1st} (line 5) are received. Afterward, the interval until the next activation of the *Second Stage* is calculated based on these decisions (line 6). Finally, the algorithm wait until *interval* decisions are received from the *First Stage* (line 7).

The operations performed to calculate the *interval* value are shown in the *updateInterval* function (lines $8-13$). The value of x is increased if both detectors provided the same decision (lines $9-1-$), or decreased if not (line 12). The *interval* assumes the value given by the exponential function b^x (line 13). The exponential function is used in the prototype because, in this way, the *interval* is rapidly increased when the *First* and *Second Stages* are converging to the same decision. Similarly, if the decisions start to differ at some point, the value of *interval* is rapidly decreased. Such divergence can occur, for example, when the PU and/or SU change its location.

5.3.2 Threshold Learning Algorithm

The *Threshold Learning Algorithm* adapts the decision threshold τ based on the decisions D_{1st} and D_{2nd}. Based on the combinations of these two decisions, the adaptation can be separated in the following functions: (*I*) *increase*τ, when $D_{1st} = \mathcal{H}_1$ and D_{2nd} = $\mathcal{H}_0$,

Algorithm 6 Exponential Feedback Algorithm

Ensure: D_{1st} as the sensing result of the *First Stage*
Ensure: D_{2nd} as the sensing result of the *Second Stage*
Ensure: I_{max} as the maximum interval between feedbacks
Ensure: b as a value greater than 1
1: **function** FEEDBACK_ALGORITHM
2: **while** SS is required **do**
3: Call the SSREQUEST of the *Second Stage*
4: Wait until an updated D_{2nd} is received
5: Wait until an updated D_{1st} is received
6: *interval* $\leftarrow$ UPDATEINTERVAL(D_{1st}, D_{2nd})
7: Wait until *interval* decisions are received from the *First Stage*
8: **function** UPDATEINTERVAL(D_{1st}, D_{2nd})
9: **if** D_{1st} is equal to D_{2nd} **then**
10: $x \leftarrow x + 1$
11: **else**
12: $x \leftarrow x - 1$
13: **return** b^x

(*II*) *decrease*τ, when $D_{1st} = \mathcal{H}_0$ and $D_{2nd} = \mathcal{H}_1$, and (*III*) *maintain*τ, when $D_{1st} = D_{2nd}$. The operations performed in each function depend on the machine learning algorithm used. For the sake of brevity, we do not show the function that compares the latest decision values of D_{1st} and D_{2nd} and calls the corresponding function. However, this function exists and it is executed whenever a decision D_{2nd} is received.

In this dissertation, we based the implementation of the aforementioned functions in the Bayesian algorithm proposed by Gong *et al.* (GONG et al., 2009), which is detailed in Algorithm 7. We highlight that the original algorithm of Gong *et al.* was proposed to be used in a cooperative environment. Thus, we changed the original algorithm in order to consider a two-stage architecture.

The parameters required by this algorithm are the decision threshold (τ), the minimum (τ_{min}) and maximum (τ_{max}) value that can be assigned to τ, the vector of the Bayesian risk for each valid threshold between τ_{min} and τ_{max} (R), the probabilities of having the channel vacant ($P_{\mathcal{H}_0}$) or occupied ($P_{\mathcal{H}_1}$), and two vectors to store the false alarm (P_f) and the miss detection (P_m) probabilities achieved by each threshold used.

The function *maintain*τ only updates the probabilities used by the current threshold τ (lines $2 - 3$). The function *increase*τ updates these same probabilities (line $5 - 6$), and in addition, updates the Bayesian risk of all thresholds in the range $[\tau, \tau_{max}]$ (line 7). Then, the decision threshold τ is increased by selecting the threshold with the smallest Bayesian risk in the interval $[\tau, \tau_{max}]$ (line 8). The function *decrease*τ is similar to the previous function, but updating the risk of the thresholds in the range $[\tau_{min}, \tau]$ (line 12) and selecting a threshold smaller than the current threshold τ (line 13).

The *a priori* probabilities $P_{\mathcal{H}_0}$ and $P_{\mathcal{H}_1}$ are calculated using the counting rule (Simple counting rule for optimal data fusion, 2003), shown in Equation 5.1. It is worth remembering that the algorithm proposed by Gong *et al.* is based on the exchange of decisions between cooperative users to calculate these probabilities. However, in the developed prototype these probabilities are based on the decisions received from the *Second Stage*.

Algorithm 7 Bayesian Threshold Learning Algorithm

Require: τ as current decision threshold
Require: τ_{min} as the minimum value for τ
Require: τ_{max} as the maximum value for τ
Require: R as a vector of the Bayesian risk of each valid threshold between *min* and *max*
Require: P_f and P_m as vectors for the false alarm and miss detection probabilities of each valid threshold between *min* and *max*
Require: $P_{\mathcal{H}_0}$ and $P_{\mathcal{H}_1}$ as *a priori* probabilities

function MAINTAINτ( )
    Update $P_{\mathcal{H}_0}$ and $P_{\mathcal{H}_1}$ using the counting rule
    Update $P_f(\tau)$ and $P_m(\tau)$
function INCREASEτ( )
    Update $P_{\mathcal{H}_0}$ and $P_{\mathcal{H}_1}$ using the counting rule
    Update $P_f(\tau)$ and $P_m(\tau)$
    Update the Bayesian risk in R of all threshold in the interval $[\tau, \tau_{max}]$
    $\tau \leftarrow$ The threshold in R and in the interval $[\tau, \tau_{max}]$ with the smallest Bayesian risk
function DECREASEτ( )
    Update $P_{\mathcal{H}_0}$ and $P_{\mathcal{H}_1}$ using D by counting rule
    Update $P_f(\tau)$ and $P_m(\tau)$
    Update the Bayesian risk in R of all threshold in the interval $[\tau_{min}, \tau]$
    $\tau \leftarrow$ The threshold in R and in the interval $[\tau_{min}, \tau]$ with the smallest Bayesian risk

$$\begin{aligned} P_{\mathcal{H}_0} &= \frac{\text{Total of } D_{2nd} == \mathcal{H}_0}{\text{Total of } D_{2nd}} \\ P_{\mathcal{H}_1} &= \frac{\text{Total of } D_{2nd} == \mathcal{H}_1}{\text{Total of } D_{2nd}} \end{aligned} \tag{5.1}$$

The Bayesian risk of a valid threshold is calculated as shown in Equation 5.2, where r is the risk, τ is the threshold value, $P_f(\cdot)$ is the probability of false alarm for τ, $P_m(\cdot)$ is the probability of miss detection and k is a weight parameter. The probabilities $P_f(\cdot)$ and $P_m(\cdot)$ are calculated independently for each valid threshold based on the decisions of the *First* and *Second Stages*.

$$r(\lambda) = P_{\mathcal{H}_0} P_f(\lambda) + k P_{\mathcal{H}_1} P_m(\lambda) \tag{5.2}$$

The parameter k is used to adjust the risk value of the miss detection. In a conservative behavior, in which harmful interference with the PU is prohibited, it is desirable to reduce the decision threshold τ. In this case, the risk value of a miss detection should be greater than the risk value of a false alarm, *i.e.* it is more desirable to have several false alarms than a single miss detection. Similarly, the decision threshold τ should be increased for a more aggressive behavior. This behavior is achieved by assigning a risk value $k < 1$ for the miss detection.

5.4 Chapter Summary

This chapter presented the prototype developed to evaluate ATA. It began by presenting the SDR platform in two parts: hardware and software. The hardware part made a brief overview of the USRP front-ends and then detailed the combination of motherboard and daughterboard used in this dissertation. The second part detailed the GNU Radio toolkit and its integration with the USRP. Afterwards, the implementations required by the *Sensing* and *Machine Learning Components* were explained.

For the *Sensing Component*, the ED, WFD and CFD were detailed. Block diagrams were used to provide the details about the operations performed by each sensing technique to generate the decision hypothesis. In addition, the difference in the complexity of each sensing technique was made clear with the analysis of the block diagrams. The use of a multi-stage architecture, which combines two or more techniques in a SU, becomes an interesting approach to reduce the sensing time while maintaining a high accuracy.

For the *Machine Learning Component*, the implementation of its two algorithms was detailed. The first one was the *Feedback Algorithm*, which uses an exponential function to adapt the interval between activations of the *Second Stage*. Then, details about the Bayesian threshold learning algorithm and how this algorithm adapts the decision threshold was given.

The integration of the *Sensing Component* and the *Machine Learning Component* enables the developed prototype to achieve a high sensing accuracy and maintain an overall low sensing time. In the next chapter, the evaluation scenario and the results obtained are presented.

6 EXPERIMENTAL EVALUATION

This chapter presents the experimental scenario and the results obtained in the evaluation of ATA. The evaluation scenario and the parameters are described in Section 6.1. Afterward, the results regarding the sensing accuracy and sensing duration are presented and discussed in Section 6.2. The proposed architecture is compared to other multi-stage architectures in Section 6.3.

6.1 Evaluation Scenario and Parameters

ATA was evaluated in an experimental environment that emulates a public safety radio channel (STEVENSON et al., 2009). In this type of channels, the PUs are police, ambulance, and firefighter cars. Usually, a public safety channel in a given geographical area is vacant because there is no presence of such cars. A SU, such as a Internet Service Provider (ISP), can access a vacant public safety channel to perform its communication. However, in special situations, such as football games or accidents, these channels become occupied. In such a scenario, the SU must sense the radio channel to detect if PU is not transmitting, *i.e.* detect the white space. The PU and the SU are detailed in Sections 6.1.1 and 6.1.2, respectively.

6.1.1 The PU: Police, ambulance, and firefighter cars

An application was developed in the GNU Radio toolkit to emulate the behavior of the PU. This application used a continuous-time Markov chain with ON and OFF states to control the generation of radio signals. This model has been widely adopted in the literature because it approximates the spectrum usage pattern of public safety radio channels (SALEEM; REHMANI, 2014). The application generates the radio signal in a preconfigured central frequency only when the ON state is active. The duration of the ON and OFF periods followed a Poisson distribution of mean and variance (σ) of 3 seconds (FAGANELO et al., 2013).

In our experimental scenario, we considered different probabilities for the PU to occupy the licensed channel (P_{ON}), *i.e.* the probability to select the ON state. The Markov chain with the ON and OFF states is illustrated in Figure 6.1. We obtained the results considering three different values for P_{ON}: (*I*) 10%, indicating a PU that rarely occupies its channel, *i.e.* low occupancy, (*II*) 50%, indicating a PU that usually occupies its channel, *i.e.* intermediate occupancy, and (*III*) 90%, indicating a PU that often occupies its licensed channel, *i.e.* high occupancy With these probabilities, we can analyze the impact of the PU activity in the sensing accuracy and sensing duration of ATA (MACALUSO et al., 2013).

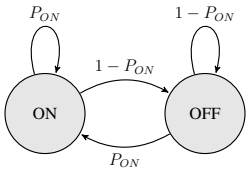


Figure 6.1: Markov chain for the public safety PU

The signal generated by the PUs in the ON state was configured to be at the central frequency of 200 MHz with 2 MHz bandwidth and using a OFDM signal. In addition, we evaluated the prototype considering a signal generated in PU with different values of Energy per bit to Noise Spectral Density Radio (EBN0), *i.e.* the normalized SNR per bit transmitted. These values were: -20, -15, -10, -5, 0 and 5 dB. A EBN0 equal to 0 dB indicates that the noise power and the signal power are equal; values below 0 dB indicate that the noise power is greater, and values above 0 dB indicate that the signal power is greater. The use of different EBN0 allowed us to analyze the prototype performance in detecting radio signals generated with different intensities (TANDRA; SAHAI, 2007). The parameters used in the experimental environment for the PU are summarized in Table 6.1.

Parameter	Value	Description
TV Channel Central Frequency	200 MHz	Central frequency when transmitting
TV Channel Bandwidth	2 MHz	Channel bandwidth when transmitting
TV Signal Modulation	OFDM	Modulation used when when transmitting
EBN0	$-20, -15, -10, -5, 0$ and 5 dB	Normalized energy per bit at the transmission
P_{ON}	10%, 50%, and 90%	Probability to select P_{ON}
σ	3 seconds	Mean time in the ON and OFF states

Table 6.1: Parameters used to configure the public safety PU

6.1.2 The SU: ISP

The prototype of ATA was deployed to sense the public safety channel and detect whether or not a police, ambulance or firefighter car is transmitting. The results were obtained using two different combinations of sensing techniques. The first combination used the ED and the WFD in the *First* and *Second Stages*, respectively. In the second combination, the WFD was replaced by the CFD. For the sake of simplicity, these two combinations are referred to as ATA-ED/WFD and ATA-ED/CFD. These combinations allowed the analysis of the impact of different sensing techniques in the performance of ATA.

The configuration of the sensing techniques used in ATA was as follows: the number of samples used to apply the test statistic Λ of the sensing techniques were fixed in 1024. The initial decision threshold (th_{1st}) of the *Threshold Learning Algorithm* was set to -75 dB, with a threshold adjustment step ($\Delta\tau_{1st}$) of 0.1 dB. The decision threshold for the *Second Stage* (th_{2nd}) was set to match a given P_{FA}. The base b for the *Feedback Algorithm* was 2 and the maximum interval between activations of the *Second Stage* (I_{max}) was 64. The parameters used in the configuration of ATA are summarized in Table 6.2.

The PU was configured to generate its signal uninterruptedly to obtain the pattern (p) used in the WFD. An application was developed to receive the television signal and to calculate the average power received in each frequency of the signal bandwidth. Figure 6.2 illustrates the resulting pattern p in the frequency domain, wherein the x axis represents

Parameter	Value	Description
N	1024 samples	Samples used by the SS techniques
τ_{1st}	-75 dB	Initial decision threshold for the First Stage
$\Delta\tau_{1st}$	0.1 dB	Step to adjust τ
τ_{min}	-120 dB	Minimum valid value for τ
τ_{max}	0 dB	Maximum valid value for τ
τ_{2nd}	Based on the P_{FA}	Decision threshold for the Second Stage
b	2	Base for the Exponential Feedback Algorithm
I_{max}	64	Maximum interval between feedbacks

Table 6.2: Parameters used to configurate ATA

the N points of the FFT and the y axis shows the normalized power in each point. The signal generated by the television broadcaster is the straight line in the center of the x axis.

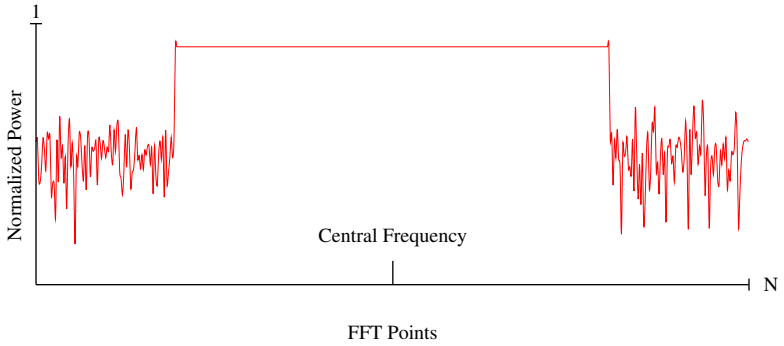


Figure 6.2: Pattern used detect the television broadcaster

Aiming to evaluate the performance of ATA, we compared it with the single-stage version of the ED, WFD, and CFD. In addition, the comparison includes the hierarchical architecture proposed by Maleki, Pandharipande, and Leus (MALEKI; PANDHARIPANDE; LEUS, 2010), referred to as Two-Stage Hierarchical Architecture (TSHA). The decision threshold of these techniques was defined in the same way as for the *Second Stage* of ATA, *i.e.* to satisfy a given P_{FA}. Finally, each experiment was executed for 20 seconds and repeated until a 95% confidence level was achieved.

6.2 Sensing Accuracy and Duration Results

In this section, we present the results regarding the sensing accuracy and sensing duration obtained in the experimental radio environment. The decision thresholds were configured to satisfy the P_{FA} of 0%, 50% and 100%. For the sake of simplicity, we separated the results in three analyses, based on the P_{FA}. In addition, we show the sensing accuracy and sensing duration for the three occupancy probabilities defined, which are $P_{ON} = 10\%, 50\%$ and 90%.

6.2.1 Results for P_{FA} equal to 0%

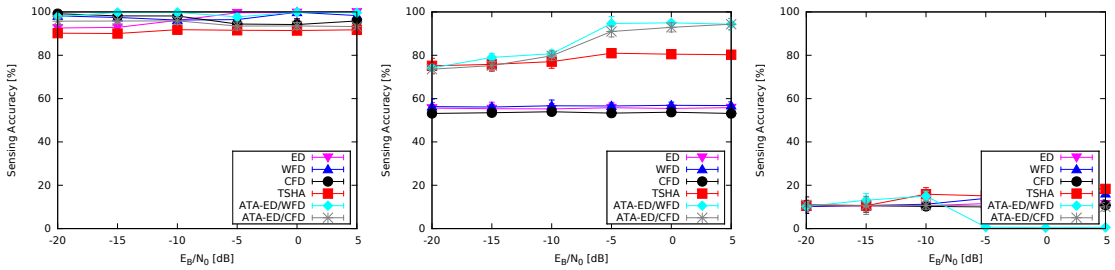
The results of the first analysis are presented in Figure 6.3, in which we considered a P_{FA} is equal to 0%. It is worth highlighting that with this P_{FA} the sensing techniques

used the highest value observed when applying the test statistic in the noise. For example, the ED used the highest energy power observed in the noise, whereas the WFD used the highest correlation and the CFD the highest cyclic correlation. Thus, with this P_{FA} the sensing techniques tend to declare the channel as vacant.

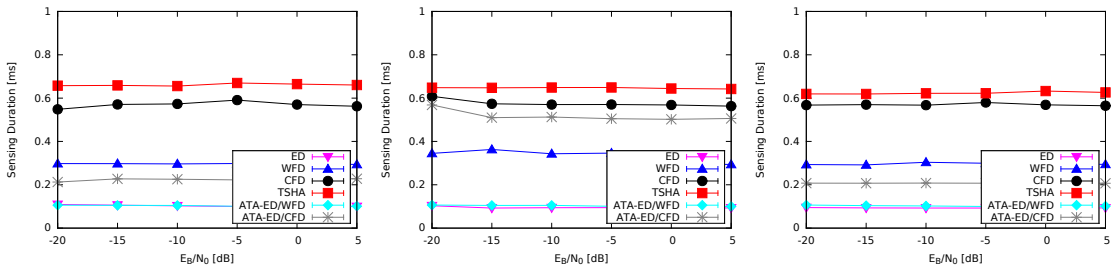
The sensing accuracy obtained when P_{FA} is equal to 0% is shown in Figures 6.3(a), 6.3(b) and 6.3(c). As can be noted, the sensing accuracy of the single-stage techniques decreased as the channel became more occupied by the PU. More precisely, when P_{ON} is 10% the sensing accuracy is above 90%, and it decreases to 50% when P_{ON} is 50% and to 10% when P_{ON} is 90%. This occurs because the sensing techniques tend to declare the channel as vacant due the P_{FA} used. We can conclude that the accuracy of a single-stage architecture for a P_{FA} of 0% is equal to $100 - P_{ON}$, *i.e.* probability of the PU not occupy the sensed channel. The decrease in accuracy of the ED and the CFD also impacted in the TSHA.

The two variations of ATA presented the highest sensing accuracy for P_{ON} equal 10% and 50%. This occurs because the learning algorithm was able to adapt the decision threshold. However, the two variations were not able to achieve a high sensing accuracy when P_{ON} was equal to 90%. This occurs because the P_{FA} used made the learning algorithm adapt the decision threshold in the wrong direction.

The sensing duration of the techniques is shown in Figures 6.3(d), 6.3(e), 6.3(f). The sensing duration of the single-stage architectures is constant for all values of P_{ON}. This occurs because the sensing duration of this architecture does not depend on the decision threshold used or the channel status occupancy. Specifically, the ED, WFD, and CFD had a sensing duration of approximately 0.1 ms, 0.30 ms and 0.56 ms, respectively. The P_{FA} equal to 0% makes the TSHA activate the second stage frequently. Because of this, the sensing duration of TSHA was of approximately 0.65 ms, which is higher than the single-stage version of the two sensing techniques used in the architecture (ED and CFD).



(a) $P_{ON} = 10\%$ (b) $P_{ON} = 50\%$ (c) $P_{ON} = 90\%$



(d) $P_{ON} = 10\%$ (e) $P_{ON} = 50\%$ (f) $P_{ON} = 90\%$

Figure 6.3: Results for $P_{FA} = 0\%$

The sensing duration of ATA-ED/CFD was 0.5 ms for P_{ON} equal to 50%, indicat-

ing that the learning algorithm required frequent feedbacks while adapting the decision threshold. For P_{ON} equal to 10% and 90%, the sensing duration of ATA-ED/CFD was of approximately 0.22 ms, indicating that less feedbacks were required to adapt the decision threshold. This occurred because the decision threshold converged to the value that made the decisions of the *First* and *Second Stages* of ATA equal. However, for P_{ON} equal to 90%, the threshold converged to a wrong value. Moreover, ATA-ED/WFD presented a sensing duration similar to the single-stage ED, *i.e.* 0.11 ms, indicating that few feedbacks were required.

With the obtained results we can conclude that ATA-ED/WFD is the best solution when P_{FA} is equal to 0%. ATA-ED/WFD presented a sensing accuracy higher than other solutions, while the sensing duration was near to the single stage ED. In addition, ATA-ED/CFD presented a high accuracy, but the sensing duration was higher than ATA-ED/WFD.

6.2.2 Results for P_{FA} equal to 50%

The second analysis considered a P_{FA} equal to 50%. The results of this analysis are shown in Figure 6.4. A P_{FA} of 50% uses the median value observed when applying the test statistic in the noise[1]. Thus, with this P_{FA} the sensing techniques declare the channel as vacant in 50% of the decisions made when only noise is present.

The sensing accuracy obtained is shown in Figures 6.4(a), 6.4(b) and 6.4(c). The single-stage architectures maintained the sensing accuracy in approximately 55% in all values P_{ON}. We highlight that with this P_{FA} the sensing accuracy should increase when the channel is often occupied. However, this was not observed in the collected results because the values used for EBN0 were not high enough to easily separate the noise from the PU signal.

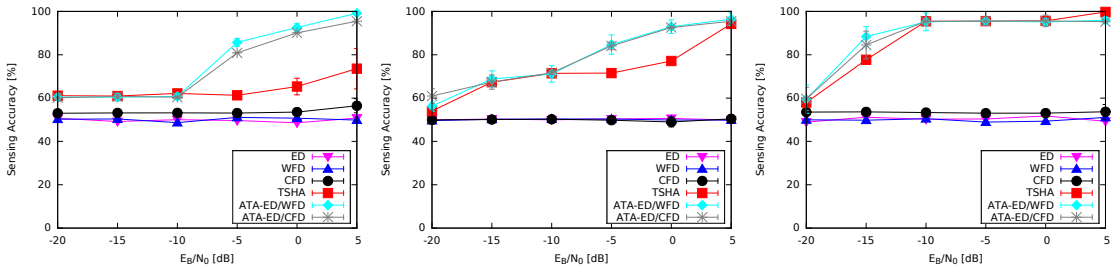
The increase in accuracy achieved by the TSHA is very clear in this analysis. We can note that the TSHA achieves better accuracies for channels often occupied. This occurred because a P_{FA} equal to 50% is suited for channels often occupied.

The two combinations of ATA achieved a sensing accuracy higher than TSHA for all values of P_{ON}. For P_{ON} equal to 10% and 50%, the two combinations and TSHA had a similar accuracy for EBN0 below -10, whereas for other EBN0, ATA achieved a higher sensing accuracy. In addition, the two combinations of ATA and TSHA had a similar sensing accuracy when P_{ON} is equal to 90%. From these results, we can conclude that the sensing accuracy of ATA and TSHA are equivalent for low EBN0, while for higher EBN0 ATA is better.

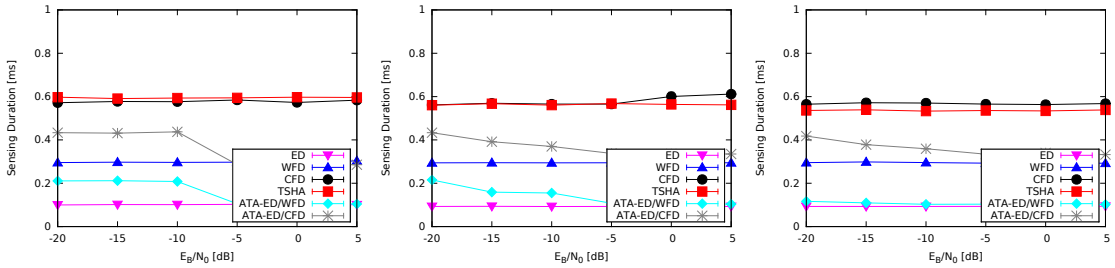
The sensing duration achieved for the P_{FA} equal to 50% is shown in Figures 6.4(d), 6.4(e), 6.4(f). The sensing duration of the single-stage architectures was equal for all values of P_{ON} and, in addition, equal to the previous P_{FA}. The TSHA reduced the number of activations of the second stage and, consequently, reduced the sensing duration, when compared to the P_{FA} equal to 0%. Also, the sensing duration of TSHA reduced as P_{ON} increased, being equal to 0.59 ms, 0.56 ms and 0.53 ms for P_{ON} equal to 10%, 50% and 90%, respectively. This occurred because the number of activations of the second stage is reduced when the occupancy of the sensed channel increases.

The sensing duration of both combinations of ATA was higher than the sensing duration obtained when P_{FA} was equal to 0%. This occurred because the learning algorithm required more feedbacks to adapt the decision threshold and not because of the change

[1]Another definition is that a P_{FA} of 50% is equivalent to select the $50th$ percentile of the observed values.



(a) $P_{ON} = 10\%$ (b) $P_{ON} = 50\%$ (c) $P_{ON} = 90\%$



(d) $P_{ON} = 10\%$ (e) $P_{ON} = 50\%$ (f) $P_{ON} = 90\%$

Figure 6.4: Results for $P_{FA} = 50\%$

in P_{FA}. Moreover, given a fixed P_{ON}, we can note that the sensing duration decreased as the EBN0 increased. This occurred because ATA requires frequent feedbacks to adapt the decision threshold for lower values of EBN0, *i.e.* the PU is weaker than noise, which hardens the threshold adaptation.

The results showed that ATA-ED/WFD was the best combination when P_{FA} is equal to 50%. This variation presented a sensing accuracy similar to or higher than the other architectures and a sensing duration that was always below the single-stage version of the WFD.

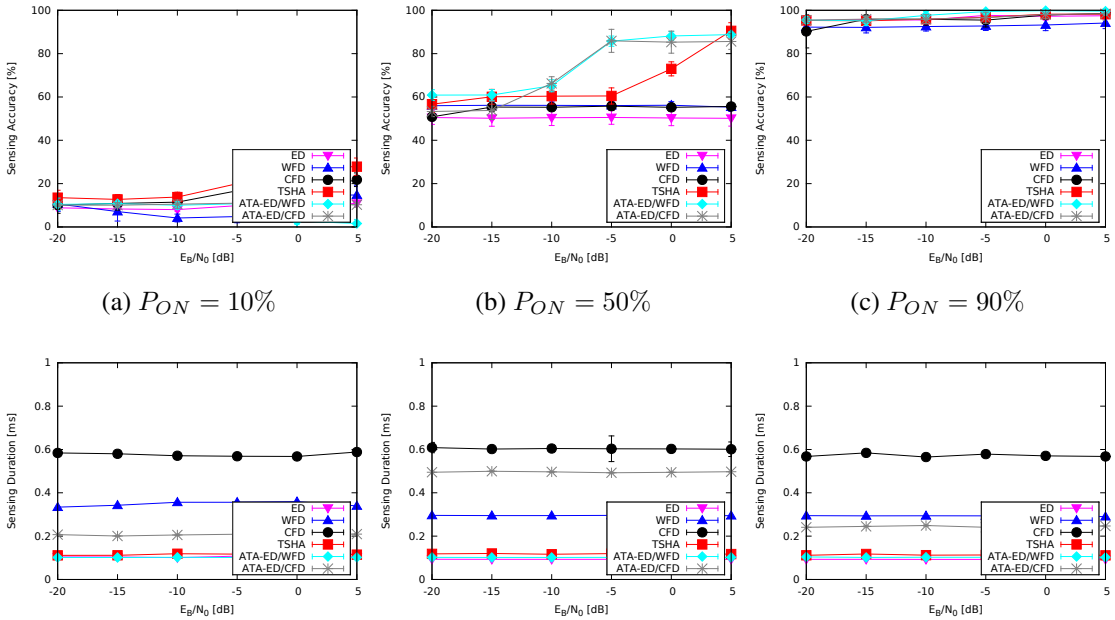
6.2.3 Results for P_{FA} equal to 100%

The third and final analysis considered a P_{FA} equal to 100%. Figure 6.5 shows the obtained results. This P_{FA} has the effect of making the sensing technique select as decision threshold the lowest value observed when applying the test statistic in the noise. Thus, the sensed channel is evaluated as occupied in practically all decisions.

The sensing accuracy obtained for P_{FA} equal to 100% is shown in Figures 6.5(a), 6.5(b) and 6.5(c). In this analysis, the sensing accuracy of the single-stage architectures and the TSHA increased with P_{ON}. More precisely, when P_{ON} is 10% the sensing accuracy was approximately of 10%, and it increased to 50% when P_{ON} was 50% and to 90% when P_{ON} was 90%. This occurred because the P_{FA} declares the channel as occupied in practically all decisions. So, the sensing accuracy increases as the PU occupies the sensed channel more often. In effect, a P_{FA} of 100% causes the sensing accuracy to be directly related with the PU occupancy probability.

The combinations of ATA presented the highest sensing accuracy for P_{ON} equal 50% and 90%. However, the two combinations were not able to achieve a high sensing accuracy when P_{ON} was equal to 10%. In this case, the P_{FA} used made the learning algorithm adapt the decision threshold in the wrong direction, which is the same problem mentioned when P_{FA} and P_{ON} are equal to 0% and 90%, respectively.

The sensing duration obtained is shown in Figures 6.3(d), 6.3(e), 6.3(f). As expected, the sensing duration of the single-stage architectures remains unchanged. The sensing duration of the TSHA was of approximately 0.12 ms for all values of P_{ON}, which is near the single-stage ED. The TSHA achieved such a low sensing duration because the first stage declared the channel as occupied in almost all decisions, rarely activating the CFD in the second stage. It is noteworthy that this P_{FA} lead the TSHA to achieve the lowest sensing duration possible by combining the ED in the first stage and the CFD in the second.



(a) $P_{ON} = 10\%$ (b) $P_{ON} = 50\%$ (c) $P_{ON} = 90\%$

(d) $P_{ON} = 10\%$ (e) $P_{ON} = 50\%$ (f) $P_{ON} = 90\%$

Figure 6.5: Results for $P_{FA} = 100\%$

The sensing duration achieved by the ATA-ED/CFD combination was higher than the sensing duration of TSHA. However, the ATA-ED/WFD combination achieved a sensing duration better than TSHA for all P_{ON}. This occurred because executing the *Second Stage* with the WFD is faster than executing it with the CFD.

With the obtained results we can conclude that ATA-ED/WFD is the best solution for all values of P_{FA}. In almost all analyses ATA-ED/WFD presented a sensing accuracy higher than the other solutions, while the sensing duration was near to the single stage ED. The two exceptions are when the P_{FA} configured lead the learning algorithm to adapt the threshold in the wrong direction, leading to low sensing accuracies.

6.3 Comparison of Multi-Stage Architectures

Finally, we perform a comparison of the current multi-stage architectures for SS. We summarized the main characteristics of the architectures in the following items: (*I*) sensing techniques used in each stage, (II) if the architecture considers threshold adaptation mechanisms, (III) if a mechanism to optimize the activations of the second stage is present, and (IV) the evaluation methodology used to obtain the results. The comparison is summarized in Table 6.3.

The lack of mechanisms to adapt the decision threshold in real time is one of the constraints identified in current multi-stage architectures for SS. To overcome the lack of such a mechanism, we integrate machine learning algorithms and SS. The threshold adaptation enabled the proposed architecture to achieve a high sensing accuracy in radio environments with different channel occupancy probabilities.

Another constraint identified in current architectures is the lack of an efficient mechanism to control the activations of the second stage. For example, the proposal of Maleki, Pandharipande, and Leus activates the second stage whenever the first stage evaluates the channel as vacant. Such mechanism is not efficient in channels often vacant. In this dissertation, we proposed a mechanism to control the activations of the second stage based on the accuracy of the first stage. Thus, the activations of the second stage are reduced when the first stage is accurate. The results showed that this mechanism enabled the proposed architecture to achieve a lower sensing duration when compared to the proposal of Maleki, Pandharipande, and Leus.

Finally, we evaluated the proposed architecture in an experimental radio environment using the USRP radio front-end and the GNU Radio toolkit. Through the experimental evaluation and the results obtained, we proved that the proposed architecture is applicable in current radio platforms.

Proposal	Techniques	Threshold Adaptation	Optimize 2nd Stage	Evaluation Methodology
Luo *et al.*	ED	No	No	Analytical and Simulation
Maleki, Pandharipande, and Leus	ED and CFD	No	No	Analytical
Nair *et al.*	ED and CFD	No	No	Simulation
Ejaz, Hasam and Kim	ED, CFD and Match Filter	No	No	Simulation
Proposal	Any	Yes	Yes	Experimental

Table 6.3: Comparison of multi-stage architectures

7 CONCLUSION AND FUTURE WORK

In this dissertation, we presented the inefficiency of the current spectrum allocation policies. Governmental agencies, knowing that the licensed users scarcely use allocated radio channels, are willing to propose new solutions to enhance the usage of the radio spectrum. In such a scenario, the dynamic spectrum access arose as the most promising solution to enhance the spectrum usage.

The dynamic spectrum access enables unlicensed users to transmit in underutilized radio channels. One constraint imposed by regulatory agencies is that these transmissions cannot interfere in the operation of a licensed user. Therefore, a radio performing dynamic access needs to perform spectrum sensing to analyze whether or not a licensed user is using a given radio channel.

The device performing spectrum sensing may encounter different noise and interference levels when sensing the radio channels. Also, the secondary user can be mobile, changing its location over time. Thus, the operating parameters of the spectrum sensing need to be adapted to different situations. Despite the research efforts to improve the performance of spectrum sensing techniques, few research efforts considered the adaptation of spectrum sensing parameters.

In this dissertation, we proposed the Multi-Stage Adaptive Threshold Architecture (ATA). The proposed architecture employs machine learning algorithms to adapt the decision threshold of sensing techniques. In addition, ATA uses a mechanism to reduce the sensing duration in environments where the sensing accuracy is high.

7.1 Main Contributions and Results Obtained

Along with the proposed architecture, this dissertation contributes to the development of a prototype of ATA. The prototype shows that the architecture can be deployed in future radios performing spectrum sensing. Moreover, a marginal contribution is the implementation of the energy detection, waveform detection, and cyclostationary feature detection sensing techniques in the GNU Radio toolkit.

Another significant contribution of this dissertation is the proposal and implementation of a more robust mechanism to control the activations of the second stage in multi-stage architectures for spectrum sensing. Although this mechanism was integrated into ATA to reduce the sensing duration when the sensing accuracy is high, other multi-stage architectures could adopt it.

ATA prototype was evaluated in an experimental radio environment resembling the IEEE 802.22. Different occupancy probabilities, noise power and probabilities of false alarm were considered. The results showed that the proposed architecture achieves a high sensing accuracy and a low sensing duration in different radio environments.

7.2 Final Remarks and Future Work

There are several opportunities for future research. Once the operation of ATA was shown as an improvement of current state-of-the-art sensing architectures, we intend to extend the analyses to more dynamic radio environments, such as the radio channels used by the 4G cellular network. To enable it, the licensed user must be modified to emulate the operation of 4G mobile devices and 4G base stations. In addition, ATA can be compared with cooperative sensing solutions.

Another improvement to ATA is to consider different machine learning algorithms to adjust the decision threshold. Such algorithms must consider some restrictions that difficult their operation, such as real-time learning and only one feedback to verify their correctness. Rafael Rodrigo Nicolay conducted an initial analysis in his course conclusion work entitled "Sensoriamento Espectral por Detecção de Energia Utilizando Aprendizado por Reforço". Nicolay proposed a State-Action-Reward-State-Action (SARSA) learning algorithm and compared it with the Bayesian learning algorithm used in this dissertation. The results showed that SARSA achieves a similar sensing accuracy bayesian but with a much lower sensing duration.

Finally, ATA can be improved to consider a cooperative feedback. More precisely, an additional input in ATA could be the cooperative decision provided by a common receiver. This input could be used as feedback by the learning algorithm to adapt the decision threshold. In addition, an algorithm to give different weights for the feedback provided by the Second Stage and the cooperative decision can be added.

REFERENCES

Aguayo Gonzalez, C.; DIETRICH, C.; REED, J. Understanding the software communications architecture. **IEEE Communications Magazine**, [S.l.], v.47, n.9, p.50–57, 2009.

AKYILDIZ, I. F.; LO, B. F.; BALAKRISHNAN, R. Cooperative spectrum sensing in cognitive radio networks: a survey. **Physical Communication**, [S.l.], v.4, n.1, p.40–62, 2011.

ALTERA. **Cyclone FPGA Family Data Sheet**. Available at http://www.altera.com/literature/ds/ds_cyc.pdf. Acessed in April/2014.

ATAPATTU, S.; TELLAMBURA, C.; JIANG, H. Energy Detection Based Cooperative Spectrum Sensing in Cognitive Radio Networks. **IEEE Transactions on Wireless Communications**, [S.l.], v.10, n.4, p.1232–1241, Apr. 2011.

AXELL, E.; LEUS, G.; LARSSON, E.; POOR, H. Spectrum Sensing for Cognitive Radio: state-of-the-art and recent advances. **IEEE Signal Processing Magazine**, [S.l.], v.29, n.3, p.101–116, 2012.

BKASSINY, M.; LI, Y.; JAYAWEERA, S. K. A Survey on Machine-Learning Techniques in Cognitive Radios. **IEEE Communications Surveys & Tutorials**, [S.l.], v.15, n.3, p.1136–1159, 2013.

CABRIC, D.; TKACHENKO, A.; BRODERSEN, R. Spectrum Sensing Measurements of Pilot, Energy, and Collaborative Detection. **IEEE Military Communications Conference**, [S.l.], p.1–7, 2006.

CHEN, X.; CHEN, H.-h.; MENG, W. Cooperative Communications for Cognitive Radio Networks From Theory to Applications. **IEEE Communications Surveys & Tutorials**, [S.l.], p.1–13, 2014.

EJAZ, W.; HASAN, N. ul; KIM, H. S. iDetection: intelligent primary user detection for cognitive radio networks. In: International Conference on Next Generation Mobile Applications, Services and Technologies, 2012. **Anais...** [S.l.: s.n.], 2012. p.153–157.

Ettus Research. Acessed in April/2014., www.ettus.com.

FAGANELO, L. R.; KUNST, R.; BOTH, C. B.; GRANVILLE, L. Z.; ROCHOL, J. Improving Reinforcement Learning Algorithms for Dynamic Spectrum Allocation in Cognitive Sensor Networks. **IEEE Wireless Communications and Networking Conference**, [S.l.], p.35–40, 2013.

FCC. **ET Docket No. 02-135**. Accessed in February, 2014., http://www.hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-228542A1.pdf.

GHASEMI, A.; SOUSA, E. Collaborative spectrum sensing for opportunistic access in fading environments. In: IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. **Anais...** [S.l.: s.n.], 2005. p.131–136.

GHASEMI, A.; SOUSA, E. Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs. **IEEE Communications Magazine**, [S.l.], v.46, n.4, p.32–39, 2008.

GNU Radio. Acessed in April/2014., www.gnuradio.org.

GONG, S.; LIU, W.; YUAN, W.; CHENG, W.; WANG, S. Threshold-Learning in Local Spectrum Sensing of Cognitive Radio. **IEEE Vehicular Technology Conference**, [S.l.], p.1–6, 2009.

GU, B.; JUNG, J.; KIM, K.; HEO, J.; PARK, N.; JEON, G.; CHO, Y. SWICOM: an sdr-based wireless communication gateway for vehicles. **IEEE Transactions on Vehicular Technology**, [S.l.], v.59, n.4, p.1593–1605, 2010.

KIM, K.; XIN, Y.; RANGARAJAN, S. Energy Detection Based Spectrum Sensing for Cognitive Radio: an experimental study. **IEEE Global Telecommunications Conference**, [S.l.], p.1–5, 2010.

KOLODZY, P. et al. Next generation communications: kickoff meeting. In: Proceedings: DARPA, 2001. **Anais...** [S.l.: s.n.], 2001. v.10.

LEHMANN, E. L.; ROMANO, J. P. **Testing statistical hypotheses**. [S.l.]: Springer, 2006. v.Third EditiThird Edition.

LUO, L.; NEIHART, N.; ROY, S.; ALLSTOT, D. A two-stage sensing technique for dynamic spectrum access. **IEEE Transactions on Wireless Communications**, [S.l.], v.8, n.6, p.3028–3037, June 2009.

MACALUSO, I.; FINN, D.; OZGUL, B.; DASILVA, L. A. Complexity of Spectrum Activity and Benefits of Reinforcement Learning for Dynamic Channel Selection. **IEEE Journal on Selected Areas in Communications**, [S.l.], v.31, n.11, p.2237–2248, 2013.

MALEKI, S.; PANDHARIPANDE, A.; LEUS, G. Two-stage spectrum sensing for cognitive radios. **IEEE International Conference on Acoustics Speech and Signal Processing**, [S.l.], p.2946–2949, 2010.

MITOLA, J. Software radios: survey, critical evaluation and future directions. **IEEE Aerospace and Electronic Systems Magazine**, [S.l.], v.8, n.4, p.25–36, 1993.

NAIR, P. R.; VINOD, A. P.; KRISHNA, A. K. A fast two stage detector for spectrum sensing in cognitive radios. **IEEE Vehicular Technology Conference**, [S.l.], p.1–5, 2011.

PEH, E. C. Y.; LIANG, Y.-c.; GUAN, Y. L.; ZENG, Y. Cooperative Spectrum Sensing in Cognitive Radio Networks with Weighted Decision Fusion Schemes. **IEEE Transactions on Wireless Communications**, [S.l.], v.9, n.12, p.3838–3847, 2010.

PEH, E.; LIANG, Y.-C.; GUAN, Y. L.; ZENG, Y. Optimization of Cooperative Sensing in Cognitive Radio Networks: a sensing-throughput tradeoff view. **IEEE Transactions on Vehicular Technology**, [S.l.], v.58, n.9, p.5294–5299, Nov 2009.

PROAKIS, J. G. **Digital communications**. [S.l.]: McGraw-Hill, New York, 2007. n.5.

ROBERTS, R.; BROWN, W.; LOOMIS, H. Computationally efficient algorithms for cyclic spectral analysis. **IEEE Signal Processing Magazine**, [S.l.], v.8, n.2, p.38–49, 1991.

ROCHOL, J. **Comunicação de Dados**. [S.l.]: Bookman Companhia Editora LTDA, 2012. n.1. (Livros Didáticos do Instituto de Informática da UFRGS, v.22).

SAHAI, A.; HOVEN, N.; TANDRA, R. Some fundamental limits on cognitive radio. **Allerton Conference on Communication, Control, and Computing**, [S.l.], p.1662–1671, 2004.

SALEEM, Y.; REHMANI, M. Primary radio user activity models for cognitive radio networks: a survey. **Journal of Network and Computer Applications**, [S.l.], v.43, 2014.

SHEN, J.; JIANG, T.; LIU, S.; ZHANG, Z. Maximum channel throughput via cooperative spectrum sensing in cognitive radio networks. **IEEE Transactions on Wireless Communications**, [S.l.], v.8, n.10, p.5166–5175, 2009.

Simple counting rule for optimal data fusion. **IEEE Conference on Control Applications**, [S.l.], p.1186–1191, 2003.

STEVENSON, C.; CHOUINARD, G.; LEI, Z.; HU, W.; SHELLHAMMER, S.; CALDWELL, W. IEEE 802.22: the first cognitive radio wireless regional area network standard. **IEEE Communications Magazine**, [S.l.], v.47, n.1, p.130–138, 2009.

TANDRA, R.; SAHAI, A. SNR Walls for Feature Detectors. **IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks**, [S.l.], p.559–570, Apr. 2007.

ULVERSOY, T. Software Defined Radio: challenges and opportunities. **IEEE Communications Surveys & Tutorials**, [S.l.], v.12, n.4, p.531–550, 2010.

YUCEK, T.; ARSLAN, H. A survey of spectrum sensing algorithms for cognitive radio applications. **IEEE Communications Surveys Tutorials**, [S.l.], v.11, n.1, p.116–130, 2009.

ZHANG, H.; WU, H. c.; LU, L. Analysis and Algorithm for Robust Adaptive Cooperative Spectrum-Sensing. **IEEE Transactions on Wireless Communications**, [S.l.], v.13, n.2, p.618–629, 2014.

ZHANG, W.; MALLIK, R.; LETAIEF, K. Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. **IEEE Transactions on Wireless Communications**, [S.l.], v.8, n.12, p.5761–5766, 2009.

APPENDIX A APPROVED PAPER – WCNC 2015

This paper presents the Multi-State Adaptive Threshold Architecture and the results obtained.

- **Title:**
  Multi-Stage Architecture for Spectrum Sensing Using Machine Learning
- **Conference:**
  IEEE Wireless Communications and Networking Conference (WCNC 2015)
- **DOI:**
  10.1109/WCNC.2015.7127484
- **Article URL:**
  https://ieeexplore.ieee.org/document/7127484
- **Conference URL:**
  http://wcnc2015.ieee-wcnc.org
- **Date:**
  March 9-12, 2015
- **Presented at:**
  New Orleans, USA

Multi-Stage Architecture for Spectrum Sensing Using Machine Learning

Maicon Kist, Leonardo Roveda Faganello, Lucas Bondan, Marcelo Antonio Marotta,
Cristiano Bonato Both, Lisandro Zambenedetti Granville, Juergen Rochol
Federal University of Rio Grande do Sul (UFRGS), Brazil
{maicon.kist, lrfaganello, lbondan, mamarotta, cbboth, granville, juergen}@inf.ufrgs.br

***Abstract*—Cognitive Radio devices perform dynamic access to the radio frequency spectrum opportunistically. Since current governmental policies do not allow interference with license holders, a cognitive radio device must perform a spectrum sensing to find vacant channels for use. Spectrum sensing techniques are based on the detection of signal characteristics, such as energy and waveform. These techniques present a trade-off regarding sensing accuracy and sensing duration. In this sense, Multi-Stage solutions emerged to combine different sensing techniques and take benefit from the best characteristics of each one. The major drawbacks in current Multi-Stage solutions are the use of fixed decision thresholds to decide if a channel is vacant and the lack of an efficient mechanism to activate the second stage. Fixed detection thresholds makes unfeasible the adaptation to different radio environments, while frequent activations of the second stage highly increases the sensing duration. In this paper we propose a Multi-Stage Adaptive Threshold Architecture for spectrum sensing, using Machine Learning algorithms to dynamically adjust the decision threshold and a feedback algorithm to control the activations of the second stage. Results showed that the proposed architecture increased the accuracy up to 36%, providing results up to 14 times faster, when compared to the literature.**

I. Introduction

Channels of the radio spectrum are usually statically allocated to primary users for a long period of time, spread along a large geographical area. The radio spectrum itself is a finite resource, and with the increasing of the spectrum usage, vacant channels are quickly becoming unavailable. Therefore, a strategy where secondary users could temporarily access allocated but idle channels should be adopted, thus overcoming the scarcity of available channels. In such a strategy, for example, a non-licensed Internet Service Provider (ISP), operating in a remote rural area, could use channels originally allocated to licensed TV broadcasters whose signal do not reach such a remote area [1].

Since secondary users must not interfere in the proper transmissions of licensed primary users, a secondary user's device must analyze the radio spectrum to evaluate which channels are vacant for use [2]. This analysis is called Spectrum Sensing (SS) and is the main function of Cognitive Radio (CR) devices [3]. In the last decade, the main techniques employed in SS were based on analyzing signal characteristics such as energy and waveform [2]. Each technique presents different performance in terms of sensing accuracy and sensing duration. For example, the Energy Detection (ED) technique is the fastest one, but it is also the least accurate, whereas the match filtering is a more accurate technique, despite also being one of the slowest [2]. To take advantage of the best characteristic of each technique, multi-stage sensing solutions were proposed, where SS is split in multiple stages, each executing different SS techniques. Usually, multi-stage solutions comprise two stages, using ED in the first stage and a Ciclostationary Feature Detection (CFD) in the second stage [4] [5].

Current state-of-the-art on SS presents two major drawbacks. The first one is related to the use of static decision thresholds to determine whether a channel is vacant. These fixed thresholds may be inappropriately defined for signal detection, mainly in unknown radio environments, where there are uncertainty about the noise power. Machine learning algorithms have been highlighted as a solution to properly adjust these thresholds, independently of the radio environment [6]. However, given the dynamicity of radio environments, machine learning techniques cannot be previously trained to automatically adapt thresholds properly to every radio environment. Another drawback in multi-stage solutions is related to the lack of an efficient mechanism to dynamically activate the second stage, which may lead to an increase in the sensing duration, degrading the performance of the CR device. To the best of our knowledge, no multi-stage solution considering the use of machine learning with a control mechanism for second stage activations, has been proposed so far.

In this paper we introduced a multi-stage Adaptive Threshold Architecture (ATA) for SS that applies machine learning to dynamically adjust the decision thresholds. In addition, a feedback algorithm to control the activation of the second stage was designed. A prototype has been developed to validate the proposed architecture in an experimental environment, using the GNU Radio framework[1] and the Universal Software Radio Peripheral 2 (USRP2) front-end[2]. The novelties of our proposed architecture are: (*i*) the use of a new machine learning based on reinforcement learning algorithm to automatically adjust the decision threshold, and (*ii*) an algorithm to dynamic control the activation of the second stage. Our proposal was compared with a two-stage (ED/CFD) solution found in the literature [4] and with the most used sensing techniques. Furthermore, we evaluated our ATA using two configurations: ED/CFD and ED/WFD. Results shown that the proposed architecture increased up to 36% the accuracy of SS

[1]http://www.gnuradio.org; [2]http://www.ettus.com

and reduced up to 14 times the sensing duration.

The remainder of this paper is organized as follows. Related work on multi-stage SS is presented in Section II. Our proposed architecture is introduced in Section III. Performance evaluations are discussed in Section IV. Finally, conclusions and future work are presented in Section V.

II. Related Work

SS has gained attention from the scientific community in the last years, due to its importance in the development of CR devices. Current state-of-the-art in SS aims to develop techniques capable of evaluate the radio channel occupancy status with the highest accuracy in the shortest duration [1]. However, balancing the existing trade-off between sensing accuracy and sensing duration is challenging. Attempting to achieve this balance, solutions such as multi-stage sensing and cooperative sensing were proposed [2] [4].

Multi-stage solutions improve the SS by applying different techniques in each stage to obtain fast yet accurate evaluations regarding the channel occupancy status. Maleki, Pandharipande, and Leus [4] proposed a two-stage architecture that executes an ED in the first stage and a CFD in the second stage. When the ED evaluates the channel as vacant, the CFD is activated to confirm its result. Using an analytical evaluation, the authors showed that combining these two techniques in a hierarchical scheme increases the accuracy when compared to a single-stage solution of both techniques. In addition, this two-stage architecture reduced the sensing duraiton when compared to the single-stage CFD. Nevertheless, in this solution the second stage is frequently activated in environments where the channels are often vacant, increasing the sensing duration.

Nair, Vinod, and Krishna [5] proposed an algorithm to control the activation of the second stage, aiming to reduce the sensing duration of the solution proposed by Maleki, Pandharipande, and Leus [4]. The authors calculated the minimum Signal-to-Noise Ratio (SNR) value necessary to consider the radio channel vacant without activating the second stage. This SNR was calculated according to a Constant False Alarm Rate (CFAR), *i.e.,* a probability of considering a channel occupied when it is vacant. Results, obtained by simulation, showed that with a minimum SNR of -15.4 dB the second stage can be inhibited without accuracy loss.

In scenarios where the cyclostationary features of licensed users are unknown, the CFD technique may be unfeasible. To overcome this obstacle, Li, Wang, and Kuang [7] extended the research of Maleki, Pandharipande, and Leus [4] by applying the maximum eigenvalue detection technique in the second stage. The performance of this extended research was evaluated analytically and by simulations. Similarly to other multi-stage solutions, the results showed that the combination of two different techniques reduces the sensing duration and raises the accuracy when compared to single-stage solutions. If the decision threshold of the first stage is unadjusted, no corrections in this sense are made, *i.e.,* in unknown environments, it may provide inaccurate results.

Different from multi-stage solutions, the goal of cooperative sensing is to improve the sensing accuracy by addressing situations where hidden users and transmission impairments could lead to a wrong picture of the spectrum occupancy. A cooperative solution may be achieved through the composition of local SS results of two or more CRs in a fusion center, *i.e.,* a centralized node responsible for providing the final result regarding the radio channel occupancy. A combination of multi-stage and cooperative sensing was proposed by Fahim, Ismail, and Tawfik [8]. In such a proposal, all CR devices execute an ED in the first stage. In the second stage, the CRs send their sensing results to a fusion center, which combine all results by majority voting. If the voting result is occupied, then all CRs assume that the radio channel is occupied. On the other hand, if the voting result is vacant, each CR considers the result of its local ED as the true result. Simulations showed that the performance of this multi-stage solution surpass the conventional cooperative solutions. Similarly, Liu *et al.* [9] proposed a Two-Stage solution that executes an ED in both stages, each one with different decision thresholds. In addition, a fusion center was introduced to gather SS results from all CRs and determine which radio channels are vacant. The main contribution of this proposal is that multi-stage sensing can be easily extended to a cooperative solution. In this sense, we argue that our architecture is also extensible to cooperative sensing.

All investigated SS solutions presented results regarding the increase of accuracy and reduction of the sensing duration. Nevertheless, we identified as one recurring constraint the use of static decision thresholds, which is a problem for dynamic radio environments. For example, different noise power and interference levels may be encountered by a mobile device that changes its location to an unknown radio environment. In such scenario, a static threshold is unfeasible [6]. Another constraint is the lack of a mechanism to activate the second stage. According to Bkassiny *et al.* [6], the use of machine learning allows the CR to increase its performance in different radio environments, by adapting its SS parameters, *e.g.,* decision threshold. In this sense, we proposed an adaptive threshold architecture to overcome the identified constraints in the discussed works. Our proposal is a multi-stage solution that uses machine learning techniques to adjust the decision threshold of the first stage, and it is better described in the next section.

III. Adaptive Threshold Architecture

In this section the proposed Multi-Stage Adaptive Threshold Architecture (ATA) for SS is presented. In Subsection III-A, we detail the architecture modeling as well as the operation of the comprised components. Finally, the feedback algorithm is explained in Subsection III-B.

A. Multi-Stage Architecture

Considering a bottom-up approach, the proposed architecture is composed of a radio front-end, the *Sensing Component*, and the *Machine Learning Component*, as can be seen in

Fig. 1. The radio front-end, *e.g.,* USRP2, provides information about the radio frequency spectrum to the *Sensing Component*.

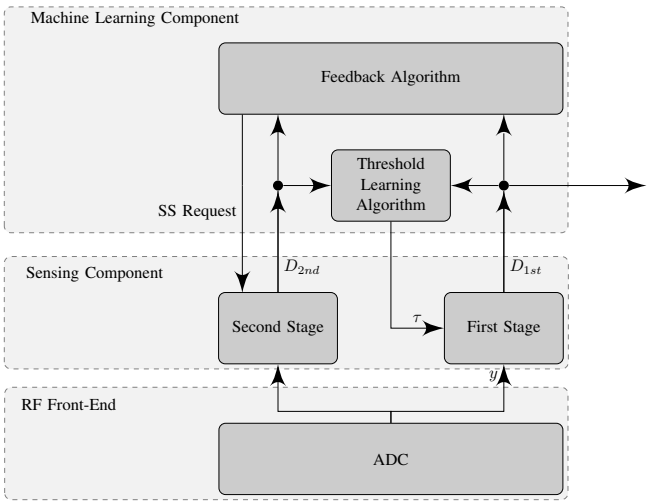


Fig. 1. Proposed Architecture

The *Sensing Component* is composed of the *First* and the *Second Stage*. The *First Stage* executes continuously, providing fast results regarding the channel occupancy for a *Channel Assignment Function*. The goal of the *Second Stage* is to be used as a feedback to verify and improve the accuracy of the *First Stage*. In this sense, a fast SS technique should be used in the *First Stage*, *e.g.,* ED, and a more accurate in the *Second Stage*, *e.g.,* CFD or WFD.

The *Machine Learning Component* is the main novelty presented in this architecture and it is composed of two algorithms: a *Threshold Learning Algorithm* and a *Feedback Algorithm*. The first is responsible for adapting the decision threshold of the *First Stage* based on the results received from the *Feedback Algorithm*. The decision threshold may be adjusted using different machine learning algorithms, *e.g.,* bayesian learning [10] or Q-learning [11]. On the other hand, the *Feedback Algorithm* controls the activation of the *Second Stage* and analyses the results of both *First* and *Second Stages*. If both stages provide divergent results, the *Second Stage* is assumed as correct, and the decision threshold of the *First Stage* is adjusted by the *Threshold Learning Algorithm*. ATA solution aims to improve the sensing accuracy and reduce sensing duration, since the number of executions of the *Second Stage* decreases as the accuracy of the *First Stage* increases. Since the *Feedback Algorithm* is one of the contributions of our proposal, we explain it in details in the following subsection.

B. Feedback Algorithm

This algorithm is responsible for (*i*) controlling the activation of the *Second Stage* of the *Sensing Component*, (*ii*) comparing the results of the *First* and *Second Stages*, and (*iii*) providing feedbacks about the accuracy of the *First Stage* to the *Threshold Learning Algorithm*. The designed algorithm can be seen in Algorithm 1.

We define D_{1st} and D_{2nd} as the results provided by the *First Stage* and *Second Stage*, respectively. In addition, the interval between feedbacks is expressed as b^x activations of the *First Stage*, where b is a positive integer value configured at the algorithm initialization, and x is controlled by the *Feedback Algorithm*. The maximum value that b^x can assume is given by I_{max}, also configured in the initialization.

Algorithm 1 Feedback Algorithm

Ensure: I_{max} as the maximum interval between feedbacks
Ensure: b as a value greater than 1
1: **function** FEEDBACK_ALGORITHM
2: **while** SS is required **do**
3: Call the SSREQUEST of the *Second Stage*
4: Wait until an updated D_{2nd} is received
5: Wait until an updated D_{1st} is received
6: $interval \leftarrow$ UPDATEINTERVAL(D_{1st}, D_{2nd})
7: Wait until $interval$ decisions are received from the *First Stage*
8: **end function**
9: **function** UPDATEINTERVAL(D_{1st}, D_{2nd})
10: **if** D_{1st} is equal to D_{2nd} **then**
11: $x \leftarrow x + 1$
12: **else**
13: $x \leftarrow x - 1$
14: **return** b^x
15: **end function**

The first operation performed by the *Feedback Algorithm* is requesting an updated decision to the *Second Stage* (line 3). In the sequence, it waits until the requested decision (line 4) and the decision D_{1st} are received (line 5). Afterwards, the interval until the next activation of the *Second Stage* is adjusted based on these decisions (line 6). Finally, the algorithms wait until $interval$ decisions are received from the *First Stage* (line 7).

The operations performed to adjust the $interval$ value are shown in the *updateinterval* function (lines $8-13$). The value of x is increase if both detectors converged in the same hypothesis (line $11-12$) or decreased if not (line $13-14$). The $interval$ assumes the value given by an exponential function, *i.e.,* b^x (line 13). The exponential function is used in the prototype because the *interval* is rapidly increased when the *First* and *Second Stages* are converging to the same decision. Similarly, the $interval$ is rapidly decreased if the threshold τ becomes invalid and must be adapted to another valued.

The first drawback presented in the researched multi-stage solutions was eliminated in this architecture, by adding a machine learning algorithm to dynamically adapt the decision threshold. In addition, the second drawback, related to the control of second stage activations was mitigated by the design of the Feedback Control Algorithm. Therefore, we combined the advantages of a multi-stage architecture with the capability of adaptation to unknown radio environments. In the next section the proposed architecture is evaluated and results are

discussed.

IV. Performance Evaluation

The evaluation of the proposed architecture as well as the obtained results are presented in this section. In Subsection IV-A, the experimental scenario and parameters are detailed. Finally, results are presented in Subsection IV-B.

A. Experimental Scenario and Parameters

The scenario comprises an Internet Service Provider (ISP) in a rural area that does not hold a licensed channel, as can be seen in Fig. 2. This ISP is a secondary user and operates within the same coverage area of a TV Broadcaster, which is a primary user. Since in our scenario the TV Broadcaster does not use its licensed channel all the time, the ISP performs opportunistic access to this channel, providing Internet access to all its clients.

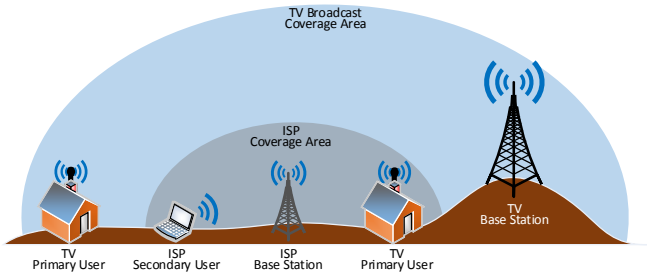


Fig. 2. Experimental Scenario

We deployed the ISP and TV Base Stations in the USRP2 radio front-end using the GNU Radio framework. The TV Base Station occupies the channel according to a Poisson distribution [11], with mean and variance (λ) of 2 seconds. The ISP Base Station performs SS to detect the primary user activity. In our experimental scenario, we considered three variations of the TV Base Station occupancy probability, defined as $P_{occ} = 10\%$, and 90%. Since ATA is based on machine learning, these occupancy probabilities were chosen in order to stress ATA and verify if the learning process does not interfere negatively in the spectrum sensing, as highlighted by Macaluso *et al.* [12].

Additional parameters considered in the experimental environment can be observed in Table I. The TV Broadcaster signal central frequency was set to 200 MHz with 2 MHz bandwidth, using a OFDM modulation. The number of signal samples (N_{samp}) considered in all sensing techniques was fixed as 1024. We used as the Threshold Learning Algorithm the Bayesian Detector proposed by Gong *et al.* [10] with an initial decision threshold (th_{1st}) of -75 dB and a threshold adjustment step (Δth_{1st}) of 0.1 dB. The decision threshold for the Second Stage (th_{2nd}) was defined as a correlation coefficient of 0.1 in all analysis. The base b for the Feedback Algorithm is 2 and the maximum interval between activations (I_{max}) is 32. Finally, each experiment was executed for 20 seconds and repeated until a 95% confidence level was achieved.

TABLE I
Parameters in the Evaluation

Parameter	Value
TV Channel Central Frequency	200 MHz
TV Channel Bandwidth	2 MHz
TV Signal Modulation	OFDM
N_{samp}	1024 samples
th_{1st}	-75 dB
Δth_{1st}	0.1 dB
th_{2nd}	0.1
b	2
I_{max}	64 activations
λ	2 s
Execution Time	20 s
Confidence Level	95%

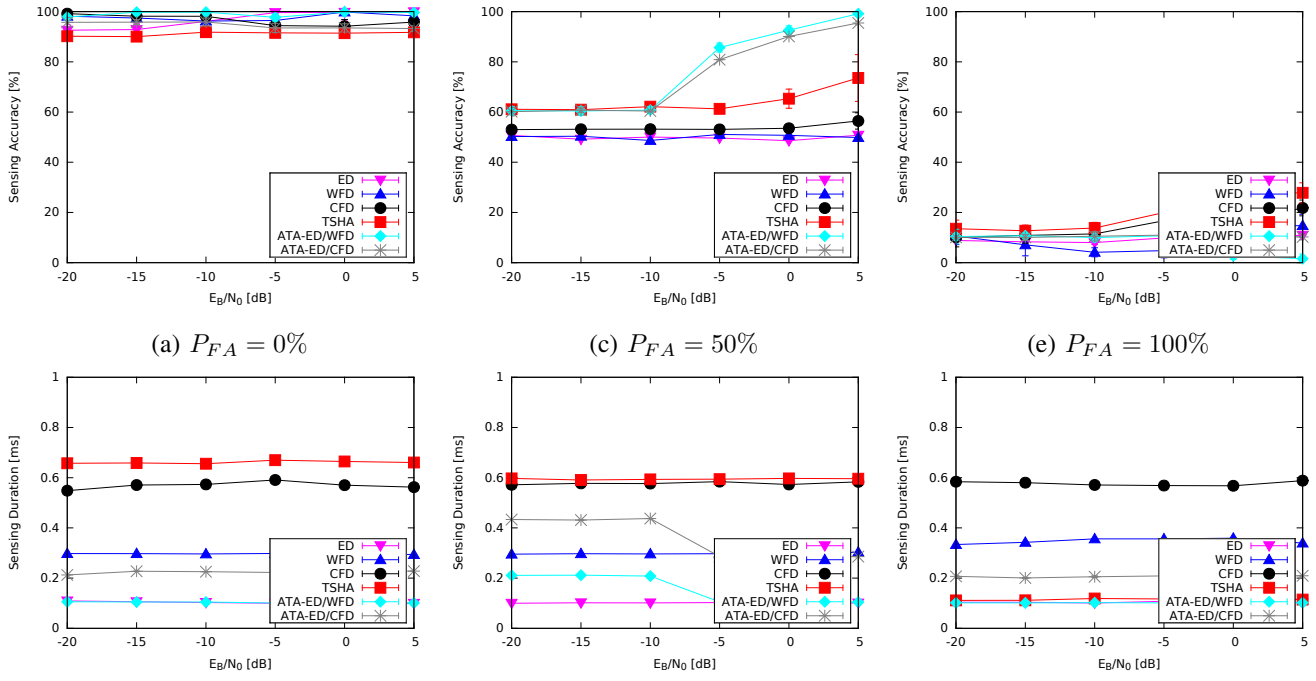
B. Results

Aiming to evaluate the performance of ATA, we compared it to the hierarchical architecture of Maleki, Pandharipande, and Leus [4], referred to as Two-Stage Hierarchical Architecture (TSHA). The decision thresholds were configured to satisfy the P_{FA} of 0%, 50% and 100%. For the sake of simplicity, we separated the results in two analysis, based on the probability of occupancy P_{ON}.

ATA results were obtained using two different combinations of SS techniques: ED/CFD and ED/WFD. These combinations allowed the analysis of the impact of different SS techniques in ATA performance. In addition, the obtained results considered two metrics: the accuracy and the sensing duration. These metrics were measured considering different Energy per Bit to Noise Spectral Power Density Ratio (E_b/N_0), *i.e.,* the normalized SNR per bit transmitted. An E_b/N_0 less than 0 indicates that the noise power is greater than the signal power.

The results of the first analysis are presented in Fig. 3. The sensing accuracy and sensing duration for $P_{FA} = 0\%$ are shown in Figs. 3(a) and 3(b). The accuracy of all techniques were constant for all E_b/N_0 values. This occurred because the $P_{FA} = 0\%$ makes the sensing techniques declare the channel as vacant, *i.e.,* the sensing accuracy is high because the channel is often vacant in this analysis. In addition, this P_{FA} made the first stage of TSHA declare the channel as vacant frequently, consequently activating the second stage and increasing the sensing duration. Because of this, the sensing duration of TSHA was above the single-stage version of the CFD. Both ATA combinations presented a higher sensing accuracy and a lower sensing duration than TSHA.

In Figs. 3(c) and 3(d) are showed the results when $P_{FA} = 50\%$. The TSHA presented a lower sensing duration for $P_{FA} = 50\%$, since less activations of CFD were required. However, both combinations of ATA were faster than TSHA. It is worth noticing that the sensing duration of ATA decreased for E_b/N_0 greater than -10 dB. This occurred because the learning algorithm was able to adapt the decision threshold to a correct value, as can be noted in the increase in the sensing accuracy for E_b/N_0 greater than -10 dB.



(a) $P_{FA} = 0\%$ (c) $P_{FA} = 50\%$ (e) $P_{FA} = 100\%$

(b) $P_{FA} = 0\%$ (d) $P_{FA} = 50\%$ (f) $P_{FA} = 100\%$

Fig. 3. Results obtained for $P_{occ} = 10\%$

Results for $P_{FA} = 100\%$ are presented in Figs. 3(e) and 3(f). Using this P_{FA} the sensing techniques always declare the channel as vacant. Because of this, TSHA did not activated the second stage, which in turn reduced it sensing duration to 0.12 ms. In addition, this P_{FA} reduced the accuracy of all sensing techniques because the sensed channel is rarely occupied. Moreover, this P_{FA} made the learning algorithm used in ATA converge the decision threshold in the wrong direction, *i.e.,* the learning algorithm adapted the decision threshold to a value in which the *First Stage* always declared the channel as occupied, matching the feedbacks provided by the *Second Stage*.

The second analysis performed to evaluate the architectures considering a PU that occupies the channel in 90% of the time, *i.e.,* $P_{occ} = 90\%$. The results obtained in this environment are showed in Fig. 4. The same P_{FA} used in the previous analysis were considered.

Results for $P_{FA} = 0\%$ are presented in Figs. 3(a) and 3(b). Differently from the previous analysis, in which this P_{FA} benefited the sensing techniques, in this analysis the sensing accuracy was severely degraded due the P_{FA} used, *i.e.,* the channel was often occupied and the P_{FA} made the sensing techniques declare the channel as vacant. In addition, the sensing duration of both combinations of ATA were lower than TSHA and the single-stage WFD and CFD.

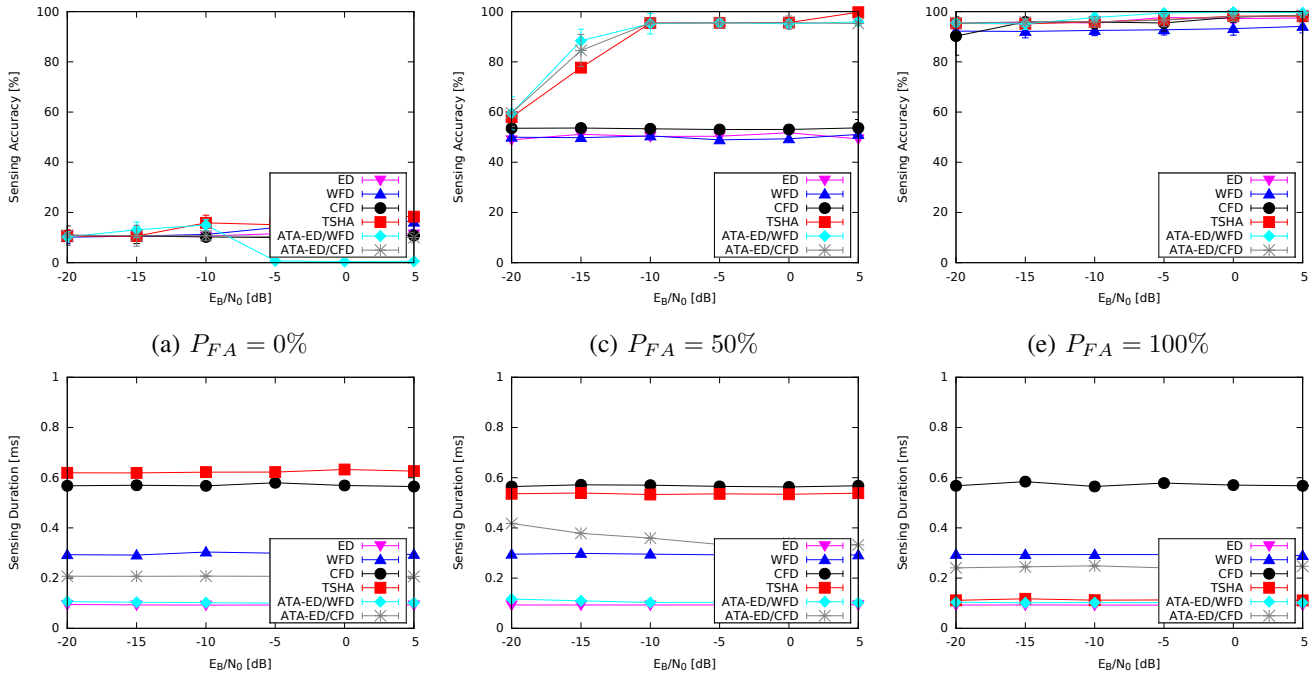
In Figs. 3(c) and 3(d) are showed the results when $P_{FA} = 50\%$. Is worth noticing that the results obtained for the sensing accuracy and sensing duration are very similar to when $P_{occ} = 10\%$. This occurred because the decision threshold used by the sensing techniques is the same in both analysis and because the P_{occ} are complementary, *i.e.,* in the first analysis the channel was occupied in 10% of the time and in the second analysis is was vacant in 10% of the time.

The accuracy and sensing duration for $P_{FA} = 100\%$ are shown in Figs. 3(e) and 3(f). In this case the sensing accuracy of all techniques was high. This occurred because the P_{FA} used made the sensing techniques declare the channel as occupied in almost all decisions. Thus, in radio environments where the PU is constantly occupying the channel, this P_{FA} is well suited. Although the sensing accuracy was similar for all sensing techniques, the ATA-ED/WFD variation achieved a lower sensing duration when compared to TSHA.

V. Conclusion

In this paper we proposed a multi-stage Adaptive Threshold Architecture for spectrum sensing. The proposed architecture comprises a Sensing Component and a Machine Learning Component. The first one is responsible for providing updated and accurate evaluations regarding the channel occupancy status. The second one applies machine learning techniques to improve the sensing accuracy and the sensing duration. In addition, the Machine Learning Component enables an adaptation to different radio environments.

Results to evaluate the proposed architecture were obtained from experiments performed in a real radio environment using the GNU Radio framework and the USRP2 radio front-end. Outcomes showed that the proposed solutions outperforms the static threshold architectures in radio environments with different channel occupancy probabilities. Directions for



(a) $P_{FA} = 0\%$ (c) $P_{FA} = 50\%$ (e) $P_{FA} = 100\%$

(b) $P_{FA} = 0\%$ (d) $P_{FA} = 50\%$ (f) $P_{FA} = 100\%$

Fig. 4. Results obtained for $P_{occ} = 90\%$

future investigations include other threshold learning algorithms, such as Q-learning and State-Action-Reward-State-Action (SARSA), different feedback algorithms and the effect of frequency selective fading effects over the sensing performance. Another possibility is to evaluate the proposed architecture in cooperative environments.

ACKNOWLEDGEMENTS

The authors thank Digitel S.A., Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Nível Superior (CAPES) for the financial support.

REFERENCES

[1] A. Ghasemi and E. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 32–39, 2008.

[2] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[3] J. Mitola III and G. Maguire Jr, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

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

[5] P. R. Nair, A. P. Vinod, and A. K. Krishna, "A fast two stage detector for spectrum sensing in cognitive radios," *IEEE Vehicular Technology Conference*, pp. 1–5, 2011.

[6] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A Survey on Machine-Learning Techniques in Cognitive Radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.

[7] Z. Li, H. Wang, and J. Kuang, "A two-step spectrum sensing scheme for cognitive radio networks," *IEEE International Conference on Information Science and Technology*, pp. 694–698, 2011.

[8] M. Fahim, M. H. Ismail, and H. Tawfik, "Multi-Stage Energy Detection Spectrum Sensing using Group Intelligence," *IEEE Symposium on Computers and Communications*, pp. 1–6, 2013.

[9] S.-Q. Liu, B.-J. Hu, and X.-Y. Wang, "Hierarchical Cooperative Spectrum Sensing Based on Double Thresholds Energy Detection," *IEEE Communications Letters*, vol. 16, no. 7, pp. 1096–1099, 2012.

[10] S. Gong, W. Liu, W. Yuan, W. Cheng, and S. Wang, "Threshold-Learning in Local Spectrum Sensing of Cognitive Radio," *IEEE Vehicular Technology Conference*, pp. 1–6, 2009.

[11] L. R. Faganelo, R. Kunst, C. B. Both, L. Z. Granville, and J. Rochol, "Improving Reinforcement Learning Algorithms for Dynamic Spectrum Allocation in Cognitive Sensor Networks," *IEEE Wireless Communications and Networking Conference*, pp. 35–40, 2013.

[12] I. Macaluso, D. Finn, B. Ozgul, and L. A. DaSilva, "Complexity of Spectrum Activity and Benefits of Reinforcement Learning for Dynamic Channel Selection," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2237–2248, 2013.